Pengantar Data Science:
Teori, Teknik, dan Aplikasinya di Era Digital

Nova Agustina, Ni Nyoman Utami Januhari, Padrul Jana, I Ketut Dedy Suryawan, Meilani Nonsi Tentua, Ester Lumba, Nuk Ghurroh Setyoningrum, R. Hafid Hardyanto, Firdiyan Syah, Nizirwan Anwar, I Made Adi Purwantara, Muhammad Fairuzabadi, Prahenusa Wahyu Ciptadi

Yash Media

Daftar Isi

Daftar Gambar

Daftar Tabel

Kata Pengantar

Puji syukur kehadirat Tuhan Yang Maha Esa atas terselesaikannya buku ini yang berjudul "**Pengantar Data Science: Teori, Teknik, dan Aplikasi di Era Digital**". Buku ini hadir sebagai upaya untuk memberikan pemahaman yang komprehensif mengenai konsep, metodologi, serta penerapan Data Science dalam berbagai bidang.

Data Science mengombinasikan berbagai disiplin ilmu seperti statistika, matematika, ilmu komputer, dan domain pengetahuan spesifik untuk mengekstraksi wawasan berharga dari data. Dalam buku ini, pembaca akan diperkenalkan pada dasar-dasar Data Science, mulai dari sejarah dan perkembangan, metodologi yang digunakan, hingga penerapan berbagai teknik analitik dalam berbagai bidang industri.

Buku ini terdiri dari beberapa bab yang secara sistematis membahas berbagai aspek penting dalam Data Science, seperti pengumpulan dan pengolahan data, eksplorasi serta visualisasi data, penggunaan algoritma machine learning, hingga peran Big Data dalam dunia modern.

Kami berharap bahwa buku ini dapat menjadi referensi yang bermanfaat bagi mahasiswa, akademisi, praktisi, serta siapa saja yang ingin memahami lebih dalam tentang Data Science.

Kami menyadari bahwa dalam penyusunan buku ini masih terdapat banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan guna perbaikan dan penyempurnaan edisi mendatang.

Tim Penulis

Bab 1
Pendahuluan Data Science

1.1 Latar Belakang

Perkembangan teknologi digital telah membawa perubahan besar dalam cara manusia hidup, bekerja, dan berinteraksi. Salah satu dampak dari kemajuan teknologi adalah ledakan jumlah data yang dihasilkan setiap detik. Saat ini data menjadi elemen penting dan sangat berharga yang mempengaruhi pengambilan keputusan di berbagai sektor, mulai dari bisnis, pendidikan, kesehatan, hingga pemerintahan. Fenomena ini dikenal sebagai era Big Data, di mana volume, kecepatan, dan variasi data yang tersedia tumbuh secara eksponensial.

Kebutuhan untuk memanfaatkan data secara efektif telah melahirkan bidang ilmu baru yang dikenal sebagai *data science*. Ilmu ini mencakup berbagai disiplin, seperti statistik, pemrograman, dan analisis data, untuk menggali wawasan yang dapat membantu organisasi memahami tren, mengidentifikasi peluang, serta memecahkan masalah dengan lebih efisien. Data science telah menjadi pilar utama dalam pengambilan keputusan berbasis data (*data-driven decision making*) karena kemampuannya yang dapat mengolah data besar dan kompleks.

Relevansi data science semakin terlihat pada kompetisi global yang terus meningkat. Perusahaan bersaing untuk mengumpulkan dan menganalisis data pelanggan untuk menciptakan strategi pemasaran yang lebih efektif. Selain itu, pada sektor kesehatan memanfaatkan data science untuk meningkatkan diagnosis dan perawatan pasien. Bahkan di dunia pendidikan, analisis data digunakan untuk memantau perkembangan siswa dan mengoptimalkan proses pembelajaran. Namun, data science juga menghadirkan tantangan yang perlu dihadapi. Pengelolaan data yang begitu besar membutuhkan infrastruktur teknologi yang canggih serta sumber daya manusia yang kompeten. Selain itu, isu terkait privasi dan keamanan data menjadi perhatian utama di era digital ini.

1.2 Perkembangan dan Sejarah Data Science

Data science telah menjadi salah satu bidang paling penting dalam era digital, tetapi akar keilmuannya dapat ditelusuri jauh ke masa lalu. Konsep dasar dari data science, yaitu analisis data (Codd, 1970), berawal sejak abad ke-17 ketika ilmuwan seperti John Graunt mulai menggunakan data untuk mempelajari pola populasi dan statistik demografi. Kemudian, pada abad ke-19, kontribusi tokoh seperti Florence Nightingale dalam visualisasi data melalui diagram membantu menyampaikan informasi yang kompleks dengan cara yang mudah dipahami. Istilah "Data science" sendiri baru diperkenalkan pada akhir abad ke-20. Pada tahun 1962, John Tukey, seorang ahli statistik, memperkenalkan konsep data analysis yang memadukan matematika, statistik, dan komputasi (Tukey, 1977) Dalam dekade berikutnya, berkembangnya teknologi komputer mendukung pengolahan data yang lebih besar, yang menjadi landasan bagi munculnya bidang baru. Pada tahun 1970-an hingga 1980-an, pengembangan basis data seperti SQL mempermudah pengelolaan data dalam jumlah besar.

Kemajuan pesat terjadi pada 1990-an dengan munculnya machine learning dan algoritma berbasis data yang memungkinkan komputer untuk mempelajari pola dari data. Dalam periode yang sama, internet mulai menghasilkan data dalam jumlah besar yang dikenal sebagai big data, membuka peluang baru dalam analisis data. Istilah "Data science" mulai digunakan secara resmi dalam literatur akademik pada akhir 1990-an dan awal 2000-an (Hinton & Salakhutdinov, 2006; Mammen et al., 2001) untuk menggambarkan disiplin ilmu yang menggabungkan statistik, komputasi, dan domain bisnis. Dalam satu dekade terakhir, data science telah berkembang pesat berkat inovasi dalam teknologi penyimpanan data, kekuatan komputasi, dan algoritma kecerdasan buatan. Peran data scientist menjadi sangat penting dalam berbagai industri, dari e-commerce hingga kesehatan, untuk mengolah data menjadi wawasan yang mendukung pengambilan keputusan. Dengan berkembangnya alat seperti Python, R, Hadoop, dan TensorFlow, data science telah menjadi lebih terjangkau dan dapat diakses oleh lebih banyak orang.

Saat ini, data science terus berkembang sebagai disiplin multidisiplin yang mengintegrasikan ilmu komputer, matematika, statistik, dan domain spesifik untuk menjawab tantangan besar di dunia modern.

1.3 Definisi, Konsep, dan Hubungan Data science dengan Bidang Lainnya

Sebagai sebuah disiplin ilmu yang berkembang pesat, data science telah menjadi pondasi utama dalam mengolah data untuk mendukung berbagai keputusan strategis. Dalam penerapannya, data science sering dianggap serupa dengan bidang lain, seperti *business analytics* dan *data analytics*, namun, data science, *business analytics* dan *data analytics* saling terhubung dalam hal penggunaan data, tetapi memiliki fokus dan tujuan yang berbeda. Ketiga bidang ini bekerja secara sinergis untuk membantu organisasi tidak hanya memahami data historis, tetapi juga memprediksi tren di masa depan dan merekomendasikan langkah strategis yang dapat diambil berdasarkan wawasan data yang diperoleh.

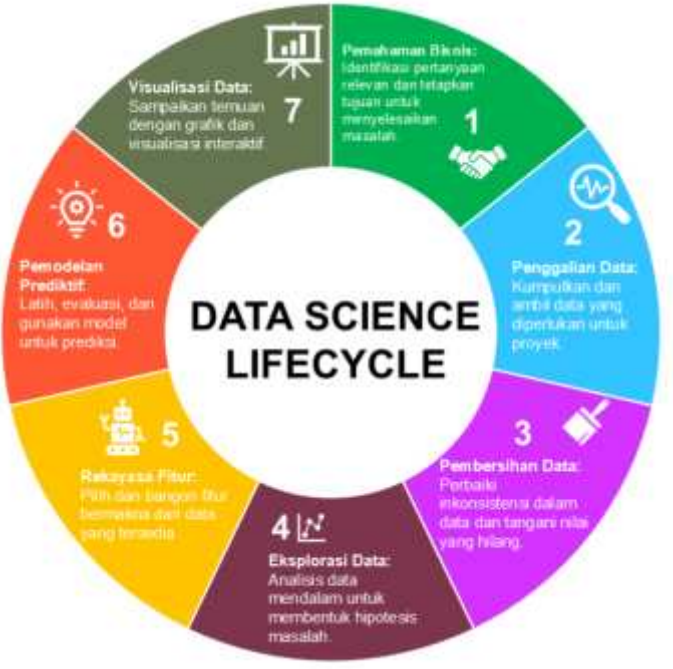
1.3.1 Definisi Data Science

Data science adalah sebuah disiplin ilmu yang berfokus pada pengumpulan, pengolahan, analisis, dan interpretasi data untuk menghasilkan wawasan yang dapat mendukung pengambilan keputusan. Ilmu data science memadukan berbagai keahlian, termasuk statistik, pemrograman, pembelajaran mesin (*machine learning*), dan visualisasi data. Tujuan utama data science adalah mengubah data mentah menjadi informasi yang bermakna dan actionable. Menurut beberapa sumber (Abdul Kadhar & Anand, 2021; Kroese et al., 2019), data science merupakan proses iteratif yang melibatkan identifikasi masalah, eksplorasi data, analisis menggunakan algoritma, hingga presentasi hasil analisis dalam bentuk yang dapat dimengerti oleh pengambil keputusan. Sehingga, data science bukan hanya tentang teknik analisis, tetapi juga bagaimana menyampaikan hasil secara efektif untuk menciptakan nilai (Fairuzabadi, Sinambela, et al., 2024).

Data science mencakup elemen statistik, matematika, pemrograman, dan domain pengetahuan khusus dengan melibatkan teknik (data mining biasanya menggunakan *tools* R, tensowflow, *spark*, *sckit-learn*) untuk membuat model yang digunakan untuk melakukan prediksi atau melakukan optimalisasi (EMC Education Services, 2015; Kroese et al., 2019). Fokus data science adalah mengelola data terstruktur dan tidak terstruktur. Data science melakukan analisis dari data yang sudah ada untuk memprediksi data dimasa yang akan datang.

1.3.2 Konsep Science Lifecycle

Untuk menerapkan data science, terdapat serangkaian tahapan yang saling berkesinambungan dan dirancang untuk menghasilkan solusi berbasis data yang efektif dan bernilai. Setiap tahapan dalam proses ini memiliki peran penting yang tidak hanya membantu mengelola data secara sistematis tetapi juga memastikan bahwa hasil akhirnya dapat memberikan wawasan yang relevan untuk pengambilan keputusan. Konsep ini adalah *Data Science Lifecycle*, yaitu kerangka kerja yang mencakup langkah-langkah mulai dari memahami masalah bisnis, mengumpulkan dan membersihkan data, hingga menganalisis dan menyajikan hasilnya dalam bentuk yang mudah dipahami oleh berbagai pemangku kepentingan. Sebagai gambaran, Gambar 1.1 mengilustrasikan tahapan-tahapan dalam *Data Science Lifecycle*.



Gambar 1.1: Data Science Lifecycle

Pada Gambar 1.1, dengan mengikuti setiap tahap secara sistematis, proses analisis berjalan secara terstruktur. Berikut penjelasan lengkap mengenai tahapan-tahapan tersebut:

1. Pemahaman Bisnis

   Tahap pertama dalam siklus hidup data science *adalah* memahami konteks bisnis. Dalam tahap ini, *data scientist* harus mampu mengidentifikasi tujuan utama proyek yang ingin dicapai. Proses ini melibatkan diskusi mendalam dengan pemangku kepentingan untuk memahami masalah yang ingin diselesaikan dan pertanyaan spesifik yang perlu dijawab melalui data. Dengan pemahaman yang jelas tentang tujuan bisnis, proses data science dapat diarahkan untuk memberikan solusi yang relevan dan efektif. Contoh kasus seperti memahami pola perilaku pelanggan atau meningkatkan efisiensi operasional sering dimulai dari tahap ini.

2. Penggalian Data

   Setelah tujuan bisnis ditetapkan, langkah selanjutnya adalah mengumpulkan data yang relevan untuk proyek. Data dapat diperoleh dari berbagai sumber, baik internal seperti *database* perusahaan, maupun eksternal seperti data publik atau hasil *web scraping*. Proses pengumpulan ini juga melibatkan pengelolaan dan pengorganisasian data agar dapat digunakan pada tahap berikutnya. Penting untuk memastikan bahwa data yang dikumpulkan cukup representatif dan sesuai untuk menjawab masalah yang telah diidentifikasi.

3. Pembersihan Data

   Data yang telah dikumpulkan sering kali tidak sempurna. Oleh karena itu, tahap pembersihan data sangat penting untuk memastikan kualitas data yang optimal. Pada tahap ini, *data scientist* akan memperbaiki inkonsistensi, mengisi nilai yang hilang, dan menangani anomali seperti *outlier*. Selain itu, data juga perlu dinormalisasi atau distandarisasi agar lebih mudah diolah pada tahap Pemodelan agar tidak memproses kualitas data yang buruk dapat memengaruhi hasil akhir analisis.

4. Eksplorasi Data

   Tahap eksplorasi data bertujuan untuk memahami pola dan hubungan dalam data melalui analisis statistik dan visualisasi. *Data scientist* menggunakan berbagai teknik eksplorasi, seperti membuat *histogram* untuk melihat distribusi data, *scatter plot* untuk memeriksa hubungan antar variabel, atau *heatmap* untuk menganalisis korelasi. Dari eksplorasi ini,

hipotesis awal dapat dibentuk, yang kemudian akan diuji pada tahap pemodelan berikutnya.

5. Rekayasa Fitur

   Rekayasa fitur adalah proses memilih, menciptakan, dan mengubah data mentah menjadi bentuk yang lebih bermanfaat untuk analisis atau pemodelan. Tahap ini melibatkan teknik seperti feature selection untuk memilih variabel yang paling relevan, dan *feature transformation* seperti *scaling* atau *one-hot encoding*. Kadang-kadang, data scientist juga menggunakan teknik pengurangan dimensi seperti untuk menyederhanakan data tanpa kehilangan informasi penting. Fitur yang baik dapat meningkatkan akurasi model secara signifikan.

6. Pemodelan Prediktif

   Tahap ini merupakan inti dari proses data science, di mana algoritma pembelajaran mesin digunakan untuk membangun model prediktif. Data yang telah diproses sebelumnya dibagi menjadi data pelatihan, validasi, dan pengujian untuk memastikan performa model yang optimal. Pemilihan algoritma dilakukan berdasarkan jenis masalah, apakah itu regresi, klasifikasi, atau *clustering*. Model yang telah dibuat kemudian dievaluasi menggunakan metrik seperti akurasi, presisi, dan recall, dan dioptimalkan melalui proses *hyperparameter tuning*.

7. Visualisasi Data

   Tahap terakhir adalah menyampaikan hasil analisis kepada pemangku kepentingan. Visualisasi data yang efektif dapat membantu menjelaskan wawasan yang telah diperoleh secara jelas dan menarik. Data scientist menggunakan alat seperti Tableau, Power BI, atau Python untuk membuat grafik interaktif yang mempermudah komunikasi pengambilan keputusan.

1.3.3 Hubungan *Data science* dengan Business Analitic dan Data Analitic

Business analytic adalah sebuah proses analisis yang berfokus pada manajemen kinerja bisnis yang biasanya digunakan pada skala perusahaan untuk membantu perusahaan membuat keputusan (Albright & Winston, 2017). Manajemen kinerja bisnis yang dimaksud untuk meningkatkan pendapatan perusahaan, misalnya melakukan analisis data menggunakan parameter permintaan, biaya satuan produksi, dan harga satuan untuk mendapatkan keputusan penentuan

strategi pemasaran, penetapan harga produk, manajemen rantai pasokan, identifikasi peluang pasar baru, dan pengelolaan risiko.

Data analytic adalah sebuah proses analisis untuk mengidentifikasi pola atau tren dengan menggunakan beberapa metode (pemrosesan, pembersihan, transformasi, dan visualisasi) dan dapat digunakan pada berbagai konteks (bukan hanya untuk bisnis). Konteks yang dimaksud misalnya kesehatan, bisnis, dan saham yang memiliki sensitifitas yang terukur (Elliott, 2020) yang membantu dalam pengambilan keputusan. Metode *data analytic* biasanya menggunakan adalah klasifikasi, klasterisasi, forecasting (Maheshwari, 2015). *Data analytic* biasanya melibatkan tools atau aplikasi sebagai metode melakukan analisis dan membuat keputusan. Tools data analytic yang umum digunakan adalah StatTools, Excel, NeuralTools, BigPicture, Evolver, Precision Tree, TopRank, dan @RISK(Albright & Winston, 2017). Fokus *data analytic* adalah data terstruktur. *Data analytic* melakukan analisis dari data yang sudah ada. Perbedaan antara *data science* dengan *business analytic* dan*, data analytic* dapat dilihat pada Tabel 1.1.

Tabel 1.1 Perbedaan Data Science, Business Analytic dan Data Analytic

Aspek	Business Analytic	Data Analytic	Data Science
Fokus Utama	Meningkatkan kinerja bisnis dan mendukung pengambilan keputusan bisnis melalui analisis data.	Mengidentifikasi pola dan tren dari data untuk pengambilan keputusan pada berbagai konteks, tidak terbatas pada bisnis.	Menciptakan model prediktif dan algoritma canggih, seringkali menggunakan machine learning dan deep learning, untuk mengekstrak wawasan dari data yang besar dan kompleks.
Metodologi	Statistik deskriptif, analisis trend, dan reporting.	Statistik deskriptif dan inferensial, eksplorasi data, dan penggunaan teknik statistik untuk menganalisis dan	Data mining, machine learning, deep learning, natural language processing.

Aspek	Business Analytic	Data Analytic	Data Science
		menginterpretasikan data.	
Penggunaan Alat	Excel, Tableau, Power BI, SAP.	SQL, Python, R, Excel, alat visualisasi data (seperti Tableau).	Python, R, TensorFlow, PyTorch, Jupyter Notebook, dan alat untuk *machine learning* serta *big data* (seperti Apache Spark).
Konteks Penerapan	Terbatas pada analisis bisnis, seperti pemasaran, keuangan, dan operasional.	Dapat diterapkan di berbagai bidang.	Dapat diterapkan di berbagai bidang.
Tingkat Kompleksitas	Rendah hingga sedang.	Sedang.	Tinggi, melibatkan data besar, algoritma kompleks, dan teknik *machine learning*.
Output	Dashboard, laporan kinerja, dan *Key Performance Indicator* bisnis.	Laporan analitis, visualisasi data, rekomendasi berbasis data.	Model prediktif, algoritma, aplikasi cerdas berbasis data, dan inovasi berbasis data.

Kesimpulan dari pemahaman tersebut adalah *business analytic* adalah subset dari *data analytic* dan fokusnya spesifik untuk keputusan bisnis. *Data analytic* adalah subset dari *data science* karena mencakup teknik atau metode yang yang digunakan saat menerapkan *data analytic*. Hubungan antara *Business Analytic*, *Data Analytic*, dan *Data Science* dapat dilihat pada Persamaan 1.

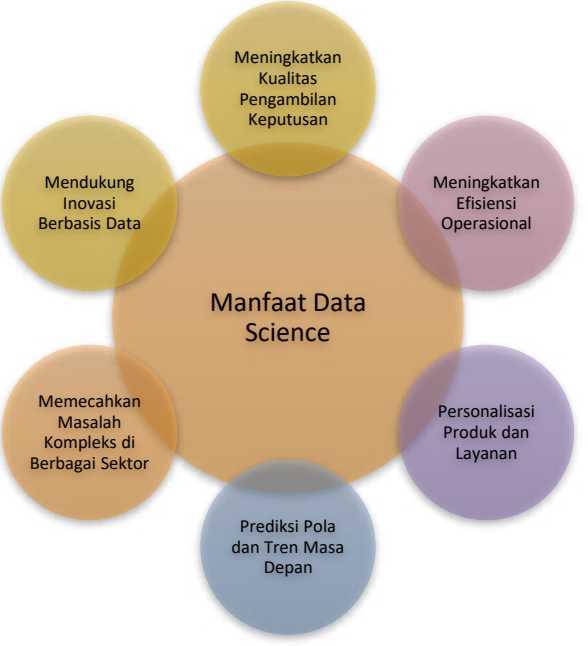
business analytic ⊆ *data analytic* ⊆ *data science* (1)

1.4 Manfaat dan Tantangan Data Science

Data Science telah menjadi salah satu bidang yang paling berpengaruh di era digital ini. Keberadaan data science memungkinkan organisasi, pemerintah, dan individu untuk memanfaatkan data secara optimal dalam pengambilan keputusan. Namun, meskipun memiliki banyak manfaat, penerapan data science juga tidak terlepas dari berbagai tantangan yang perlu diatasi.

1.4.1 Manfaat Data Science

Data science telah menjadi pilar utama dalam mendukung pengambilan keputusan di berbagai sektor, baik dalam skala individu, organisasi, maupun pemerintah. Manfaat utamanya terletak pada kemampuannya untuk mengolah, menganalisis, dan menginterpretasi data dalam jumlah besar untuk menghasilkan wawasan yang berharga. Berikut adalah penjabaran mendalam mengenai manfaat data science dalam berbagai bidang:



Gambar 1.2: Manfaat Data Science

1. Meningkatkan Kualitas Pengambilan Keputusan

   Salah satu kekuatan utama data science adalah kemampuannya untuk mendukung keputusan berbasis fakta. Dengan menganalisis data historis dan data waktu nyata, data science memungkinkan organisasi untuk memahami pola, tren, dan anomali yang mungkin tidak terlihat secara langsung. Misalnya, perusahaan dapat menggunakan analisis data untuk menentukan produk yang paling diminati, mengoptimalkan strategi pemasaran, atau mengidentifikasi segmen pelanggan yang memerlukan perhatian khusus. Keputusan berbasis data ini tidak hanya meningkatkan akurasi tetapi juga mengurangi risiko yang berkaitan dengan pengambilan keputusan intuitif (Provost & Fawcett, 2013).

2. Meningkatkan Efisiensi Operasional

   Data science memainkan peran penting dalam mengotomatisasi proses yang sebelumnya manual dan memakan waktu. Dengan menggunakan algoritma pembelajaran mesin (machine learning) dan otomatisasi berbasis data, organisasi dapat mengurangi biaya operasional dan meningkatkan produktivitas. Contohnya, perusahaan logistik menggunakan data science untuk mengoptimalkan rute pengiriman, yang tidak hanya menghemat waktu tetapi juga mengurangi konsumsi bahan bakar. Di sektor manufaktur, data science digunakan untuk memprediksi kerusakan mesin melalui analisis prediktif, sehingga mengurangi waktu henti operasional (downtime) (Choi et al., 2018) (Choi et al., 2018).

3. Personalisasi Produk dan Layanan

   Dalam dunia bisnis, personalisasi telah menjadi faktor kunci untuk menarik dan mempertahankan pelanggan. Data science memungkinkan analisis mendalam terhadap preferensi, perilaku, dan kebutuhan pelanggan. Dengan memanfaatkan data ini, perusahaan dapat menawarkan produk dan layanan yang sesuai dengan keinginan pelanggan. Misalnya, platform streaming seperti Netflix menggunakan algoritma berbasis data untuk merekomendasikan film dan serial berdasarkan kebiasaan menonton pengguna, sementara e-commerce seperti Amazon memberikan rekomendasi produk yang relevan (Rajaraman & Ullman, 2011).

4. Prediksi Pola dan Tren Masa Depan

   Salah satu kekuatan data science adalah kemampuannya untuk melakukan analisis prediktif. Dengan menggunakan model statistik dan algoritma machine learning, data science dapat memproyeksikan tren masa depan berdasarkan data historis. Dalam dunia bisnis, hal ini membantu perusahaan

merespons perubahan pasar dengan cepat. Di sektor keuangan, data science digunakan untuk memprediksi risiko investasi atau pola pasar saham. Di bidang kesehatan, data science digunakan untuk memprediksi penyebaran penyakit, membantu dalam pengambilan keputusan kesehatan masyarakat, dan merancang strategi mitigasi yang efektif (James et al., 2013).

5. Memecahkan Masalah Kompleks di Berbagai Sektor

   Data science telah menunjukkan kemampuan luar biasa dalam memecahkan masalah kompleks yang sebelumnya sulit diatasi. Misalnya, dalam bidang energi, data science digunakan untuk mengoptimalkan distribusi energi berdasarkan permintaan real-time dan prediksi cuaca. Di sektor pertanian, data science membantu petani meningkatkan hasil panen dengan menganalisis data cuaca, kualitas tanah, dan kebutuhan tanaman. Dalam dunia pendidikan, data science digunakan untuk mempersonalisasi pengalaman belajar siswa dan mengidentifikasi faktor-faktor yang memengaruhi kesuksesan akademik (Zikopoulos & Eaton, 2012).

6. Mendukung Inovasi Berbasis Data

   Data science tidak hanya membantu dalam pengambilan keputusan tetapi juga mendorong inovasi. Dengan menggali wawasan dari data yang tersedia, organisasi dapat menemukan peluang baru yang sebelumnya tidak terlihat. Misalnya, perusahaan teknologi menggunakan data science untuk mengembangkan produk berbasis kecerdasan buatan seperti asisten virtual, chatbot, atau sistem rekomendasi (Domingos, 2015).

Data science telah membuktikan dirinya sebagai alat yang sangat kuat untuk mendorong pertumbuhan, efisiensi, dan inovasi di berbagai sektor. Dengan memanfaatkan data yang ada secara efektif, organisasi tidak hanya dapat meningkatkan kinerja mereka tetapi juga menciptakan nilai yang signifikan bagi pelanggan, masyarakat, dan pemangku kepentingan lainnya. Namun, untuk memaksimalkan manfaat ini, penting bagi organisasi untuk memiliki infrastruktur data yang kuat, tenaga kerja yang terampil, dan komitmen terhadap pengelolaan data yang etis.

1.4.2 Tantangan Data Science

Meskipun data science memiliki manfaat yang luar biasa, penerapannya juga menghadapi sejumlah tantangan yang memengaruhi efektivitas dan keberhasilannya di berbagai sektor. Berikut adalah penjabaran lebih mendalam mengenai tantangan utama dalam penerapan data science dan bagaimana pendekatan strategis dapat membantu mengatasinya:

1. Kualitas dan Integritas Data

   Kualitas data yang buruk menjadi salah satu hambatan terbesar dalam data science. Data yang tidak lengkap, tidak akurat, atau tidak terstruktur sering kali mengurangi keandalan hasil analisis. Sebagai contoh, data dengan nilai yang hilang (missing values) atau data yang bias dapat menyebabkan model prediktif memberikan hasil yang tidak valid (Provost & Fawcett, 2013). Pengelolaan data yang baik melalui proses pembersihan data (data cleaning) dan validasi menjadi langkah penting untuk mengatasi masalah ini.

2. Privasi dan Keamanan Data

   Dalam banyak kasus, data yang digunakan dalam data science mengandung informasi sensitif, seperti data pribadi pengguna atau data finansial perusahaan. Pelanggaran privasi dan serangan siber dapat memiliki konsekuensi serius, termasuk kehilangan kepercayaan pelanggan dan denda yang besar. Regulasi seperti GDPR (General Data Protection Regulation) di Eropa dan PDP (Perlindungan Data Pribadi) di Indonesia memberikan kerangka hukum untuk memastikan privasi dan keamanan data tetap terjaga (Voigt & dem Bussche, 2017). Organisasi harus mengadopsi langkah-langkah teknis seperti enkripsi, autentikasi multifaktor, dan audit keamanan secara berkala untuk melindungi data.

3. Kekurangan Tenaga Ahli Data Science

   Permintaan yang terus meningkat terhadap data scientist yang terampil menciptakan kesenjangan yang signifikan antara kebutuhan industri dan pasokan tenaga kerja. Data scientist tidak hanya harus memiliki kemampuan teknis seperti pemrograman dan analisis statistik, tetapi juga pemahaman bisnis untuk menginterpretasi data dengan konteks yang tepat. Upaya untuk meningkatkan keterampilan tenaga kerja melalui program pelatihan, kursus online, dan pendidikan formal di bidang data science menjadi solusi yang penting untuk mengatasi kekurangan ini (Zikopoulos & Eaton, 2012).

4. Biaya dan Infrastruktur Teknologi

   Data science memerlukan infrastruktur teknologi yang canggih, seperti penyimpanan data berskala besar, komputasi awan, dan perangkat keras yang kuat untuk mendukung analisis data yang kompleks. Biaya untuk membangun infrastruktur ini bisa menjadi kendala bagi organisasi kecil atau yang memiliki keterbatasan sumber daya (Choi et al., 2018). Namun, adopsi teknologi cloud computing yang fleksibel dan scalable dapat

membantu mengurangi beban biaya awal dan memungkinkan organisasi untuk hanya membayar sumber daya yang digunakan.

5. Kompleksitas Interpretasi Hasil Analisis

   Hasil analisis data sering kali sulit dipahami oleh pihak non-teknis dalam organisasi, sehingga mengurangi dampak dari wawasan yang diperoleh. Data scientist harus mampu menyajikan hasil analisis dalam bentuk yang lebih sederhana dan mudah dimengerti, misalnya melalui visualisasi data atau storytelling berbasis data (James et al., 2013). Kemampuan komunikasi yang baik menjadi keterampilan yang tak kalah penting bagi data scientist untuk menjembatani kesenjangan antara hasil analisis teknis dan implementasi strategis.

Meskipun menghadapi berbagai tantangan, data science memiliki potensi besar untuk menciptakan nilai tambah dan inovasi yang berkelanjutan. Berikut adalah beberapa pendekatan yang dapat diambil untuk mengatasi tantangan-tantangan ini:

1. Investasi dalam Infrastruktur dan Teknologi

   Memanfaatkan teknologi cloud dan open-source tools dapat membantu organisasi kecil mengadopsi data science tanpa biaya besar.

2. Pengelolaan Data yang Lebih Baik

   Mengimplementasikan kebijakan manajemen data yang ketat, termasuk validasi dan pembersihan data secara berkala, untuk memastikan kualitas data yang tinggi.

3. Pelatihan dan Pengembangan Tenaga Kerja

   Meningkatkan keterampilan tenaga kerja melalui pelatihan internal, kolaborasi dengan institusi pendidikan, atau sertifikasi profesional di bidang data science.

4. Penerapan Regulasi yang Ketat

   Mengikuti standar privasi dan keamanan data untuk melindungi data sensitif, sekaligus membangun kepercayaan pelanggan.

5. Peningkatan Keterampilan Komunikasi

   Melatih data scientist untuk menyampaikan hasil analisis secara sederhana dan relevan bagi para pemangku kepentingan non-teknis.

Bab 2
Peran dan Ruang Lingkup Data Science

2.1 Peran Data Science dalam Dunia Modern

Dalam era digital yang berkembang pesat, data telah menjadi salah satu aset paling berharga di dunia. Setiap aktivitas manusia, mulai dari interaksi di media sosial hingga transaksi keuangan, menghasilkan jejak data yang dapat dikumpulkan, dianalisis, dan dimanfaatkan untuk berbagai tujuan. Data tidak lagi hanya menjadi produk sampingan dari aktivitas bisnis, melainkan telah berubah menjadi bahan bakar utama yang mendorong inovasi dan pengambilan keputusan strategis di berbagai sektor (Provost & Fawcett, 2013). Transformasi ini membawa kita ke era di mana kemampuan untuk mengelola dan memanfaatkan data menjadi keunggulan kompetitif yang signifikan (Brynjolfsson & McAfee, 2014).

Gambar 2.1: Ilustrasi Peran dan Ruang Lingkup Data Science

Revolusi digital yang dimulai dengan internet dan meluas melalui teknologi mobile, media sosial, dan perangkat IoT (Internet of Things) telah menciptakan lonjakan eksponensial dalam volume data yang tersedia (Gandomi & Haider, 2015). Menurut laporan IDC (2021), jumlah data global diperkirakan akan mencapai 175 zettabytes pada tahun 2025, menandakan kebutuhan mendesak untuk teknologi dan pendekatan yang mampu mengelola, menganalisis, dan mengekstrak nilai dari data tersebut.

Data science muncul sebagai disiplin yang memungkinkan organisasi untuk memanfaatkan kekayaan data ini guna menghasilkan wawasan yang mendalam dan membuat keputusan berbasis bukti (Dhar, 2013). Dengan menggabungkan metode statistik, pembelajaran mesin, dan teknik pemrograman, data science memainkan peran sentral dalam mengubah data mentah menjadi informasi yang berarti (Wickham & Grolemund, 2017).

Peran data science dalam dunia modern meliputi berbagai aspek yang secara langsung memengaruhi kehidupan sehari-hari dan proses bisnis:

1. Pengambilan Keputusan Berbasis Data

   Organisasi menggunakan data untuk memperkuat proses pengambilan keputusan, mengurangi ketidakpastian, dan mengidentifikasi pola yang tidak dapat dideteksi melalui metode tradisional (McKinsey Global Institute, 2016).

2. Otomatisasi dan Optimasi Proses Bisnis

   Data science memungkinkan otomatisasi proses yang kompleks melalui algoritma pembelajaran mesin dan kecerdasan buatan (Russell & Norvig, 2020). Contohnya adalah implementasi sistem rekomendasi di platform e-commerce seperti Amazon dan Netflix, yang meningkatkan pengalaman pengguna dan pendapatan perusahaan (Aggarwal, 2016).

3. Inovasi dan Produk Baru

   Perusahaan teknologi besar seperti Google dan Facebook mengandalkan data science untuk mengembangkan produk dan layanan baru yang revolusioner. Misalnya, pengenalan suara dan pengenalan wajah yang didukung oleh teknologi kecerdasan buatan memperkuat interaksi pengguna dengan perangkat mereka (LeCun et al., 2015).

4. Analisis Prediktif dan Preskriptif

   Melalui teknik analisis prediktif, perusahaan dapat memproyeksikan tren masa depan dan merancang strategi yang lebih efektif. Selain itu, analisis preskriptif membantu memberikan rekomendasi tindakan yang optimal berdasarkan data yang tersedia (Shmueli & Koppius, 2011).

Meskipun data science menawarkan berbagai manfaat, terdapat tantangan yang harus diatasi, termasuk masalah privasi data, etika, dan keamanan informasi (Zuboff, 2019). Namun, dengan regulasi yang tepat dan pengembangan teknologi yang bertanggung jawab, potensi data science untuk meningkatkan kualitas hidup dan efisiensi bisnis tetap sangat besar (Dignum, 2019).

Dalam dunia modern yang didorong oleh data, peran data science tidak dapat disangkal. Sebagai alat yang mampu mengungkap pola tersembunyi, memperkirakan hasil, dan mengoptimalkan proses, data science menjadi komponen integral dari strategi bisnis dan inovasi teknologi. Dengan kemajuan yang terus berlanjut dalam kecerdasan buatan dan pembelajaran mesin, data science akan terus membentuk masa depan digital di berbagai sektor kehidupan.

2.2 Ruang Lingkup dan Aplikasi Data Science

Data science merupakan pendekatan multidisiplin yang menggabungkan metode ilmiah, algoritma, dan sistem komputasi untuk mengekstrak pengetahuan dan wawasan dari data yang terstruktur maupun tidak terstruktur. Dengan kemampuan ini, data science telah menjadi pendorong utama inovasi dan efisiensi dalam berbagai industri. Dalam bagian ini, akan dibahas secara mendalam tentang disiplin ilmu yang membentuk data science, proses kerjanya, serta berbagai aplikasi di berbagai sektor.

2.2.1 Disiplin Ilmu

Data science juga menggabungkan elemen penting dari kecerdasan buatan (AI) dan pembelajaran mendalam (deep learning) yang memungkinkan model untuk meningkatkan akurasi prediksi dengan memanfaatkan jaringan saraf tiruan. Dengan algoritma yang lebih kompleks, model deep learning dapat mengenali pola yang lebih halus dalam data, sehingga mendukung aplikasi seperti pengenalan wajah, analisis sentimen, dan pengolahan bahasa alami (Goodfellow, Bengio, & Courville, 2016).

Gambar 2.2: Disiplin Ilmu Data Science

Data science merupakan bidang multidisiplin yang menggabungkan berbagai cabang ilmu pengetahuan dan teknologi untuk mengolah data dan mendapatkan wawasan yang bernilai. Beberapa disiplin utama yang membentuk data science antara lain:

1. Statistik dan Probabilitas
   Statistik menyediakan dasar untuk pengumpulan, pengolahan, dan interpretasi data. Metode statistik digunakan untuk menguji hipotesis, mengukur hubungan antar variabel, dan membuat prediksi (Montgomery & Runger, 2018).
2. Matematika Terapan
   Matematika, khususnya aljabar linear dan kalkulus, menjadi fondasi algoritma pembelajaran mesin dan teknik optimasi (Strang, 2016).
3. Pemrograman Komputer
   Bahasa pemrograman seperti Python, R, dan SQL memungkinkan manipulasi data, pembuatan model, dan visualisasi hasil (McKinney, 2017).
4. Pembelajaran Mesin (*Machine Learning*)
   Machine learning mencakup algoritma yang memungkinkan komputer untuk belajar dari data dan membuat prediksi tanpa diprogram secara eksplisit (Fairuzabadi, Adytia, et al., 2024; Goodfellow et al., 2016).

5. Basis Data dan Manajemen Data
   Mengelola data dalam volume besar membutuhkan keahlian di bidang basis data, termasuk sistem manajemen basis data relasional (RDBMS) dan teknologi big data seperti Hadoop dan Spark (Zikopoulos & Eaton, 2012).
6. Visualisasi Data
   Teknik visualisasi digunakan untuk menyajikan data dalam bentuk grafik, diagram, dan dashboard yang mudah dipahami oleh pemangku kepentingan (Few, 2009).

2.2.2 Proses Kerja Data Science

Proses kerja dalam data science mengikuti alur yang sistematis untuk memastikan hasil yang akurat dan relevan. Proses tersebut meliputi:

Gambar 2.3: Proses Kerja Data Science

1. Pengumpulan Data
   Data dikumpulkan dari berbagai sumber seperti database internal, sensor IoT, dan media sosial. Tahap ini melibatkan teknologi seperti web scraping dan API.
2. Pembersihan Data (*Data Cleaning*)
   Data mentah sering kali tidak terstruktur dan mengandung noise. Proses pembersihan meliputi penanganan nilai yang hilang, penghapusan data duplikat, dan standarisasi format data (Dasu & Johnson, 2003).
3. Eksplorasi Data (*Exploratory Data Analysis*)
   Teknik eksplorasi digunakan untuk memahami pola dan hubungan dalam data. Statistik deskriptif dan visualisasi sering digunakan untuk menyoroti anomali dan tren (Tukey, 1977).

4. Analisis dan Model Prediktif
   Model statistik dan algoritma pembelajaran mesin diterapkan untuk memprediksi hasil dan mengidentifikasi pola. Model ini diuji dan divalidasi untuk memastikan performanya (Hastie et al., 2009).
5. Visualisasi dan Pelaporan
   Hasil analisis disajikan dalam bentuk visualisasi yang intuitif untuk mendukung pengambilan keputusan berbasis data.

2.2.3 Aplikasi Data Science di Berbagai Bidang

Data science telah menjadi kunci dalam mendorong inovasi dan efisiensi di berbagai sektor industri. Dengan kemampuannya untuk mengolah data dalam jumlah besar dan memberikan wawasan yang mendalam, data science memungkinkan organisasi untuk membuat keputusan yang lebih cerdas dan strategis. Penerapannya mencakup berbagai bidang, mulai dari bisnis, pemasaran, manufaktur, teknologi informasi, hingga kesehatan dan bioteknologi. Berikut ini adalah beberapa contoh aplikasi data science di berbagai sektor yang menunjukkan dampaknya yang signifikan.

Gambar 2.4: Contoh Aplikasi Data Science Di Berbagai Sektor

1. Bisnis dan Keuangan
   a. Analisis Risiko dan Deteksi Penipuan
      Algoritma pembelajaran mesin di bank digunakan untuk menganalisis pola transaksi dan mengidentifikasi anomali yang menunjukkan aktivitas mencurigakan, seperti penipuan kartu kredit atau transaksi ilegal (Ngai et al., 2011). Teknologi ini juga membantu dalam pemantauan transaksi real-time dan penilaian risiko kredit.
   b. Analisis Sentimen
      Data yang diperoleh dari ulasan pelanggan dan media sosial dianalisis untuk mengevaluasi sentimen publik terhadap produk dan layanan. Ini

memungkinkan perusahaan untuk merespons kebutuhan pelanggan secara proaktif dan meningkatkan strategi pemasaran (B. Liu, 2012).

2. Pemasaran dan E-commerce
   a. Sistem Rekomendasi
      Platform seperti Amazon dan Netflix memanfaatkan sistem rekomendasi yang memprediksi preferensi pelanggan berdasarkan pola perilaku mereka, meningkatkan personalisasi layanan dan mendorong penjualan (Aggarwal, 2016).
   b. Segmentasi Pelanggan
      Algoritma klastering mengelompokkan pelanggan berdasarkan karakteristik dan kebiasaan belanja mereka, memungkinkan strategi pemasaran yang lebih tepat sasaran (Wedel & Kamakura, 2000).
3. Manufaktur dan Logistik
   a. Prediksi Pemeliharaan (*Predictive Maintenance*)
      Model prediktif memonitor kondisi mesin untuk memperkirakan waktu kegagalan, memungkinkan tindakan preventif dan mengurangi downtime (Jardine et al., 2006).
   b. Optimasi Rantai Pasokan
      Analisis data real-time meningkatkan efisiensi logistik dan manajemen inventaris, memastikan pengiriman tepat waktu dan biaya minimal (Simchi-Levi et al., 2003).

2.3 Perbedaan Data Science, Data Analytics, dan Machine Learning

Dalam era digital, istilah Data Science, Data Analytics, dan Machine Learning sering kali digunakan secara bergantian. Namun, ketiga bidang ini memiliki perbedaan yang signifikan dalam pendekatan, metode, dan tujuan aplikasinya. Bagian ini akan menjelaskan definisi, hubungan, serta perbedaan ketiganya melalui tabel perbandingan dan studi kasus yang relevan.

2.3.1 Perbedaan Secara Konsep

Data science, data analytics, dan machine learning memiliki keterkaitan yang erat tetapi juga memiliki perbedaan yang mendasar. Data science mencakup keseluruhan proses untuk mengelola, menganalisis, dan memanfaatkan data dalam pengambilan keputusan. Data analytics berfokus pada analisis data untuk

memberikan wawasan spesifik dan mendukung keputusan yang telah ditetapkan. Sementara itu, machine learning menitikberatkan pada pengembangan algoritma yang dapat mempelajari pola dari data dan membuat prediksi secara otomatis.

1. Data Science
   Data Science adalah disiplin multidisiplin yang mencakup metode, proses, algoritma, dan sistem untuk mengekstraksi pengetahuan dan wawasan dari data terstruktur maupun tidak terstruktur (Provost & Fawcett, 2013). Ini mencakup seluruh proses mulai dari pengumpulan data, pembersihan, eksplorasi, analisis, dan pembuatan model prediktif.
2. Data Analytics
   Data Analytics berfokus pada penerapan metode statistik dan analisis kuantitatif untuk mengidentifikasi pola, tren, dan hubungan dalam data (Barton & Court, 2012). Ini terutama digunakan untuk memahami data historis dan mendukung pengambilan keputusan bisnis.
3. Machine Learning
   Machine Learning adalah subbidang dari Artificial Intelligence (AI) yang mempelajari algoritma yang memungkinkan komputer untuk belajar dari data dan membuat prediksi tanpa diprogram secara eksplisit (Goodfellow et al., 2016). Fokusnya adalah mengembangkan model yang dapat menggeneralisasi pola dari data yang diberikan.

Tabel di bawah ini memberikan perbandingan antara Data Science, Data Analytics, dan Machine Learning berdasarkan beberapa aspek utama. Perbandingan ini membantu memperjelas fokus dan aplikasi masing-masing bidang.

Tabel 2.1. Data Science, Data Analytics & Machine Learning

Aspek	Data Science	Data Analytics	Machine Learning
Fokus Utama	Mengolah data untuk mendapatkan wawasan mendalam	Analisis data untuk mendukung keputusan	Mengembangkan algoritma untuk prediksi
Pendekatan	Multidisiplin (Statistik, Machine Learning, AI)	Analisis kuantitatif dengan metode statistik	Model prediktif berbasis algoritma

Aspek	Data Science	Data Analytics	Machine Learning
Output	Model, insight, dan prediksi	Laporan, visualisasi, dan rekomendasi	Prediksi dan klasifikasi berdasarkan pola
Penggunaan Utama	Pengolahan data besar, pembelajaran mesin	Analisis deskriptif dan diagnostik	Automasi dan sistem prediktif
Alat yang Digunakan	Python, R, SQL, TensorFlow	Excel, Tableau, Power BI	Scikit-learn, TensorFlow, PyTorch

2.3.2 Hubungan dan Tumpang Tindih

Ketiga bidang ini saling terkait dan sering kali bekerja bersama dalam berbagai proyek data. Data Science mencakup Data Analytics dan Machine Learning sebagai bagian integral dari proses analisis data.

Hubungan antara Data Science dan Data Analytics

Data Analytics adalah bagian dari Data Science yang berfokus pada eksplorasi dan interpretasi data untuk memahami pola historis dan tren saat ini. Data Science memperluas cakupan ini dengan mencakup pemodelan prediktif dan teknik yang lebih kompleks seperti Machine Learning dan Artificial Intelligence. Contohnya, Data Analytics dapat digunakan untuk menganalisis penjualan tahunan, sementara Data Science melibatkan prediksi penjualan di masa depan dengan menggunakan model prediktif yang didukung oleh Machine Learning.

Hubungan antara Data Science dan Machine Learning

Machine Learning adalah komponen inti dalam Data Science yang mendukung pembangunan model prediktif dan sistem otomatisasi. Data Science mencakup seluruh siklus data, termasuk pengumpulan, pembersihan, analisis eksplorasi, dan implementasi model Machine Learning. Misalnya, dalam proyek deteksi penipuan, Data Science mengelola seluruh pipeline data dari ekstraksi hingga pengolahan fitur, sedangkan Machine Learning berfokus pada pembangunan dan pengujian model prediksi.

Hubungan antara Data Analytics dan Machine Learning

Data Analytics sering kali memanfaatkan Machine Learning untuk analisis yang lebih dalam, terutama dalam prediksi tren dan pengambilan keputusan otomatis.

Sebagai contoh, dalam analisis pelanggan, Data Analytics mengelompokkan pelanggan berdasarkan perilaku pembelian, sementara Machine Learning membangun model prediktif untuk merekomendasikan produk atau layanan yang relevan. Dengan demikian, Machine Learning memperkuat kemampuan Data Analytics dengan memberikan hasil yang lebih akurat dan mendalam.

2.3.3 Studi Kasus

Studi kasus berikut ini memberikan gambaran nyata tentang bagaimana Data Science, Data Analytics, dan Machine Learning digunakan untuk menyelesaikan berbagai masalah di dunia nyata. Setiap kasus menyoroti peran khusus dari masing-masing bidang dan bagaimana mereka berkontribusi pada pengambilan keputusan berbasis data.

Kasus 1: Prediksi Penjualan Produk (Data Science)

Sebuah perusahaan ritel menggunakan Data Science untuk mengintegrasikan data dari penjualan, media sosial, dan sensor IoT. Data tersebut dibersihkan, dianalisis, dan diproses menggunakan algoritma Machine Learning untuk memprediksi tren penjualan dan mengoptimalkan stok barang.

Kasus 2: Evaluasi Kampanye Pemasaran (Data Analytics)

Perusahaan pemasaran menggunakan Data Analytics untuk mengevaluasi efektivitas kampanye iklan mereka. Melalui analisis data historis, mereka mengidentifikasi pola perilaku pelanggan dan menyesuaikan strategi pemasaran berdasarkan temuan tersebut.

Kasus 3: Deteksi Penipuan Transaksi (Machine Learning)

Sebuah bank menerapkan algoritma Machine Learning untuk memantau transaksi keuangan secara real-time. Model tersebut melatih dirinya dari pola transaksi historis dan mampu mengenali anomali yang mencurigakan untuk mencegah penipuan.

Meskipun Data Science, Data Analytics, dan Machine Learning memiliki area fokus yang berbeda, ketiganya saling melengkapi dalam menganalisis dan memanfaatkan data untuk pengambilan keputusan. Data Science mencakup pendekatan yang lebih luas, Data Analytics berfokus pada analisis deskriptif dan diagnostik, sedangkan Machine Learning lebih menitikberatkan pada pengembangan model prediktif. Pemahaman tentang perbedaan ini membantu organisasi mengoptimalkan pemanfaatan data untuk berbagai kebutuhan bisnis.

2.4 Profesi dan Keterampilan dalam Data Science

Data science telah menjadi salah satu bidang yang paling dicari di era digital saat ini. Profesi yang berkaitan dengan data science berkembang pesat seiring dengan meningkatnya kebutuhan akan analisis data dan teknologi berbasis kecerdasan buatan. Bab ini akan membahas secara mendetail berbagai peran utama dalam data science, keterampilan yang dibutuhkan, serta tren karir dan prospek masa depan di bidang ini.

2.4.1 Peran Utama dalam Data Science

Profesi dalam data science melibatkan berbagai peran yang saling melengkapi dalam siklus hidup data. Setiap peran memiliki tanggung jawab dan keahlian khusus yang dibutuhkan untuk memproses, menganalisis, dan menginterpretasikan data guna mendukung pengambilan keputusan berbasis informasi. Berikut ini adalah penjelasan mendalam mengenai peran-peran utama yang ada dalam dunia data science.

2.4.1.1 Data Scientist

Data Scientist bertanggung jawab untuk merancang dan mengimplementasikan model analisis data guna mengekstrak wawasan yang berguna bagi pengambilan keputusan. Mereka memanfaatkan metode statistik, machine learning, dan algoritma kompleks untuk memprediksi hasil dan mengidentifikasi pola.

Tugas Utama:

- Mengumpulkan dan membersihkan data dari berbagai sumber.
- Mengembangkan model prediktif dan algoritma machine learning.
- Membuat visualisasi data yang informatif.
- Mengkomunikasikan temuan kepada pemangku kepentingan.
- Mengidentifikasi tren dan pola data yang signifikan.

Keterampilan Penting:

- Pemrograman: Python, R, SQL.
- Analisis Statistik: Pandas, NumPy, Scikit-learn.
- Machine Learning: TensorFlow, Keras, PyTorch.
- Visualisasi Data: Tableau, Power BI, Matplotlib.

- Soft Skills: Komunikasi, pemecahan masalah, dan pemikiran kritis.

2.4.1.2 Data Engineer

Data Engineer berfokus pada pembangunan dan pemeliharaan infrastruktur data. Mereka bertanggung jawab dalam merancang arsitektur data dan memastikan bahwa data tersedia, aman, dan dapat digunakan.

Tugas Utama:

- Mengembangkan pipeline data untuk mengumpulkan dan memproses data dalam jumlah besar.
- Mengelola basis data dan sistem penyimpanan data.
- Membangun dan mengoptimalkan sistem ETL (Extract, Transform, Load).
- Mengintegrasikan berbagai sumber data.
- Memastikan keamanan data dan kepatuhan terhadap regulasi.

Keterampilan Penting:

- Pemrograman: Python, Java, Scala.
- Basis Data: SQL, NoSQL, Hadoop, Apache Spark.
- Pengelolaan Data: AWS, Azure, Google Cloud Platform.
- Tools ETL: Apache Nifi, Talend.
- Soft Skills: Manajemen proyek dan pemecahan masalah teknis.

2.4.1.3 Data Analyst

Data Analyst berfokus pada analisis data untuk menghasilkan laporan dan wawasan bisnis. Mereka memvisualisasikan data dan menginterpretasikan hasil analisis untuk mendukung pengambilan keputusan.

Tugas Utama:

- Melakukan analisis deskriptif dan diagnostik.
- Menghasilkan laporan dan dashboard.
- Mengidentifikasi pola dan tren dalam data historis.
- Mengembangkan strategi berdasarkan hasil analisis.

Keterampilan Penting:

- Pemrograman: Python, R, SQL.
- Visualisasi Data: Excel, Tableau, Power BI.
- Statistik dan Analisis: SPSS, SAS.

- Soft Skills: Keterampilan komunikasi dan presentasi yang kuat.

2.4.1.4 4. Machine Learning Engineer

Machine Learning Engineer berfokus pada pembangunan, implementasi, dan pemeliharaan model machine learning untuk aplikasi prediktif.

Tugas Utama:

- Mengembangkan model machine learning dengan algoritma canggih.
- Menguji dan menyempurnakan model untuk akurasi yang lebih baik.
- Mengintegrasikan model ke dalam aplikasi dan sistem produksi.
- Mengelola siklus hidup model dan memperbaruinya secara berkala.

Keterampilan Penting:

- Pemrograman: Python, Java, C++.
- Machine Learning Framework: TensorFlow, Keras, PyTorch.
- Teknik Optimasi Model: Hyperparameter tuning, grid search.
- Deployment: Docker, Kubernetes, AWS Sagemaker.
- Soft Skills: Pemikiran analitis dan kolaborasi tim.

2.4.2 Keterampilan yang Dibutuhkan dalam Data Science

Data science menggabungkan berbagai keterampilan teknis dan non-teknis yang diperlukan untuk mengelola, menganalisis, dan mengekstrak wawasan dari data. Keterampilan ini mencakup kemampuan pemrograman, pengetahuan statistik, serta keahlian dalam teknologi big data dan machine learning. Selain keterampilan teknis, profesional data science juga memerlukan kemampuan komunikasi dan pemecahan masalah yang kuat untuk menginterpretasikan hasil analisis dan menyampaikan wawasan kepada pemangku kepentingan.

Tabel 2.2: Rangkuman Keterampilan Teknis dalam Data Science

Kategori	Teknologi dan Alat Utama
Bahasa Pemrograman	Python, R, SQL - untuk manipulasi data, analisis statistik, dan pengembangan algoritma.
Big Data Tools	Hadoop, Spark - untuk pemrosesan data dalam skala besar dan analitik terdistribusi.

Kategori	Teknologi dan Alat Utama
Machine Learning Frameworks	TensorFlow, Keras, PyTorch - untuk membangun dan melatih model prediktif serta deep learning.
Visualisasi Data	Tableau, Power BI, Matplotlib, Seaborn - untuk presentasi visual yang efektif dan dashboard interaktif.
Basis Data	SQL, NoSQL (MongoDB, Cassandra) - untuk manajemen data terstruktur dan tidak terstruktur.
Cloud Computing	AWS, Azure, Google Cloud - untuk penyimpanan data yang fleksibel dan skalabilitas komputasi berbasis cloud.

Keterampilan non-teknis dalam data science memang sangat penting karena dapat membantu menyampaikan temuan yang kompleks kepada audiens yang tidak memiliki latar belakang teknis, serta mendukung proses kolaborasi dalam tim lintas fungsi. Berikut adalah penjelasan mendalam mengenai keterampilan-keterampilan non-teknis yang dibutuhkan dalam data science.

2.4.2.1 Komunikasi Efektif

Komunikasi efektif adalah keterampilan yang sangat penting bagi seorang data scientist, terutama ketika mereka harus menyampaikan hasil analisis data kepada pemangku kepentingan yang tidak memiliki pengetahuan teknis mendalam. Keterampilan ini meliputi kemampuan untuk (Glass & Callahan, 2014):

- Menerjemahkan Hasil Analisis
  Seorang data scientist harus dapat mengubah temuan yang rumit menjadi informasi yang dapat dimengerti oleh orang-orang dari berbagai latar belakang, seperti manajer, eksekutif, atau klien. Ini termasuk menjelaskan model statistik, prediksi, atau rekomendasi dengan bahasa yang mudah dimengerti dan relevan dengan konteks bisnis atau keputusan strategis yang perlu diambil.
- Presentasi Visual
  Penyajian data dalam bentuk visual yang jelas, seperti grafik atau dashboard interaktif, sangat membantu dalam menyampaikan informasi. Visualisasi ini harus mudah dipahami dan menggambarkan informasi kunci secara

langsung. Dengan begitu, audiens dapat dengan cepat menangkap inti dari analisis yang disajikan.

- Menyesuaikan Pesan
  Data scientist harus dapat menyesuaikan gaya komunikasi mereka sesuai dengan audiens. Untuk audiens teknis, mungkin perlu menggunakan istilah teknis dan mendalam, sementara untuk audiens non-teknis, bahasa yang lebih sederhana dan aplikasi praktis dari temuan harus ditekankan.

2.4.2.2 Pemecahan Masalah

Pemecahan masalah adalah kemampuan untuk menghadapi dan mengatasi tantangan dalam proses analisis data. Dalam data science, tantangan ini bisa datang dalam berbagai bentuk, baik itu masalah teknis, seperti data yang tidak lengkap atau tidak konsisten, atau masalah analitis, seperti memilih model yang tepat untuk suatu tujuan.

- Pendekatan Sistematis
  Data scientist perlu pendekatan yang terstruktur dalam menganalisis masalah. Mereka harus dapat mengidentifikasi akar penyebab masalah, menyusun hipotesis, dan merancang eksperimen atau analisis untuk mengujinya. Pendekatan berbasis data sangat penting agar solusi yang ditemukan adalah yang paling tepat dan dapat diandalkan.
- Adaptasi terhadap Ketidakpastian
  Dalam dunia data, sangat mungkin untuk menghadapi data yang tidak sempurna atau masalah yang belum pernah dijumpai sebelumnya. Kemampuan untuk beradaptasi dan tetap menemukan solusi meskipun kondisi tidak ideal adalah keterampilan yang sangat dihargai. Hal ini termasuk kemampuan untuk memilih teknik atau model yang tepat meskipun ada ketidakpastian dalam data.

2.4.2.3 Pemikiran Kritis

Pemikiran kritis adalah kemampuan untuk berpikir secara logis dan objektif dalam mengevaluasi data dan informasi yang tersedia. Keterampilan ini sangat penting untuk memastikan bahwa analisis yang dilakukan benar-benar dapat dipercaya dan menghasilkan keputusan yang tepat.

- Evaluasi Data dan Model
  Data scientist harus mampu mengevaluasi kualitas data yang digunakan, serta memastikan bahwa metodologi analisis yang diterapkan sesuai dan valid. Mereka juga harus mampu menilai kelemahan dari model atau analisis yang digunakan dan mencari cara untuk memperbaikinya.
- Menghasilkan Rekomendasi Berdasarkan Bukti
  Pemikiran kritis juga mencakup kemampuan untuk membuat keputusan berdasarkan bukti yang ada, bukan hanya asumsi atau intuisi. Ini berarti data scientist harus dapat menginterpretasikan hasil analisis secara mendalam dan memastikan bahwa rekomendasi yang diberikan berbasis pada analisis yang objektif.
- Menghindari Bias
  Pemikiran kritis melibatkan kesadaran terhadap bias yang mungkin muncul dalam analisis, baik itu bias yang disebabkan oleh data yang tidak representatif atau bias yang berasal dari preferensi pribadi dalam memilih model. Data scientist yang baik selalu berusaha untuk mengurangi bias dalam proses analisis mereka.

2.4.2.4 Kolaborasi Tim

Bekerja dalam tim lintas fungsi adalah aspek yang tidak kalah penting dalam data science, karena banyak proyek data melibatkan kolaborasi antara berbagai departemen atau disiplin ilmu. Kemampuan untuk bekerja dengan orang-orang dari berbagai latar belakang, seperti tim pemasaran, tim pengembangan produk, atau manajemen, sangat penting untuk kesuksesan proyek (Shan et al., 2015).

- Berkomunikasi dengan Berbagai Pihak

  Dalam kolaborasi lintas fungsi, seorang data scientist perlu mengembangkan kemampuan untuk berkomunikasi dengan berbagai pihak yang mungkin memiliki tujuan, latar belakang, atau cara berpikir yang berbeda. Ini membutuhkan fleksibilitas dalam pendekatan dan kejelasan dalam menjelaskan bagaimana data dapat membantu mencapai tujuan bersama.

- Beradaptasi dengan Kebutuhan Tim

  Terkadang, data scientist harus menyesuaikan pendekatan analisis mereka dengan kebutuhan tim atau organisasi. Misalnya, seorang data scientist yang bekerja dengan tim pemasaran mungkin harus fokus pada analisis

perilaku pelanggan, sementara dengan tim produk, mereka mungkin perlu lebih berfokus pada prediksi kinerja produk baru.

- Mengelola Konflik

  Dalam tim lintas fungsi, perbedaan pendapat atau prioritas mungkin muncul. Seorang data scientist harus memiliki keterampilan dalam mengelola konflik dan bekerja untuk mencapai konsensus, serta memahami bagaimana data dan analisis yang mereka buat dapat mendukung keputusan yang lebih baik dalam organisasi.

  Keterampilan-keterampilan non-teknis ini membantu data scientist tidak hanya dalam melakukan pekerjaan teknis mereka dengan baik, tetapi juga dalam memastikan bahwa hasil analisis mereka dapat diterima dan digunakan oleh pemangku kepentingan dalam organisasi. Keterampilan ini juga memperkuat kemampuan mereka untuk berkolaborasi secara efektif dengan berbagai pihak yang terlibat dalam proyek data.

Bab 3
Statistika dan Matematika Dasar untuk Data Science

3.1 Konsep Dasar Statistika

Pada bagian ini dibahas mengenai beberapa konsep dasar statistika diantaranya definisi dan ruang lingkup statistika, data dalam statistika, sumber data dan penyajian data. Konsep tersebut mutlak dipahami sebagai landasan untuk implementasi statistika secara lebih mendalam.

3.1.1 Definisi dan Ruang Lingkup Statistika

3.1.1.1 Statistika Deskriptif

Pengertian Statistika Deskriptif

Statistika deskriptif adalah proses mengubah data penelitian menjadi bentuk yang lebih terstruktur seperti tabel, agar lebih mudah dipahami dan diinterpretasikan. informasi yang diperoleh dari statistika deskriptif mencakup ukuran pemusatan data, ukuran penyebaran data, serta kecenderungan yang ada pada suatu kelompok data (David Freedman Robert Pisani & Purves, 2007).

Ruang Lingkup Statistika Deskriptif

Ruang lingkup statistika deskriptif meliputi penyajian data, pengukuran tendensi sentral, pengukuran variabilitas, angka indeks serta deret waktu.

3.1.1.2 Statistika Inferensial

Statistika Inferensial adalah metode yang digunakan untuk memperoleh informasi tentang populasi berdasarkan sampel, dengan menganalisis dan menginterpretasikan data untuk menarik kesimpulan (Hatani, 2008). Statistika inferensial ini juga mencakup rangkaian metode yang berkaitan dengan analisis sebagai data, yang nantinya digunakan untuk meramalkan atau menarik kesimpulan mengenai data keseluruhan dari populasi tersebut. Ciri-ciri statistika inferensial antara lain: data yang dianalisis berasal dari sampel acak, digunakan

untuk menggeneralisasi dan meramalkan ciri penting suatu variabel serta hubungan antar variabel, generalisasi dan ramalan yang diterapkan pada seluruh populasi berdasarkan analisis data sampel dan generalisasi dan ramalan dilakukan melalui uji hipotesis atau pengecekan asumsi.

Ruang lingkup statistika inferensial meliputi probabilitas, metode pengambilan sampel, uji hipotesis, statistika parametrik (seperti korelasi dan regresi), serta statistika nonparametrik.

3.1.1.3 Perbedaan Statistika Deskriptif dengan Statistika Inferensial.

Statistika deskriptif terbatas pada penyajian data dalam bentuk tabel, diagram, grafik dan ukuran lainnya, serta bertujuan untuk menggambarkan karakteristik data. Sementara itu, statistika inferensial tidak hanya mencakup statistika deskriptif, tetapi juga digunakan untuk melakukan estimasi dan menarik kesimpulan mengenai populasi berdasarkan sampelnya. Statistika inferensial bertujuan untuk menarik kesimpulan mengenai populasi dengan menganalisis sampel.

3.1.2 Data dalam Statistika

3.1.2.1 Jenis Data

Berdasarkan bentuk dan sifatnya, data penelitian dapat dibedakan dalam dua jenis yaitu data kualitatif dan data kuantitatif (David S. Moore William I. Notz & Fligner, 2017).

1) Data Kualitatif

   Data kualitatif adalah data yang berbentuk kata-kata, bukan dalam bentuk angka. Data kualitatif diperoleh melalui berbagai macam teknik pengumpulan data misalnya wawancara, analisis dokumen, diskusi terfokus, atau observasi yang telah dituangkan dalam catatan lapangan.

   a) Data Kasus adalah data yang menjelaskan kasus tertentu. Data kasus hanya berlaku untuk kasus tertentu serta tidak bertujuan untuk menguji hipotesis tertentu.
   b) Data Pengalaman Individu adalah data yang berisi keterangan mengenai apa yang dialami oleh individu sebagai warga masyarakat tertentu yang menjadi objek penelitian.

2) Data Kuantitatif

Data Kuantitatif adalah data yang berbentuk angka atau bilangan. Sesuai dengan bentuknya, data kuantitatif dapat diolah atau dianalisis menggunakan teknik perhitungan matematika atau statistika.

a) Data Nominal atau sering disebut juga data kategori, data yang diperoleh melalui mengelompokkan objek berdasarkan kategori tertentu.
b) Data Diskrit, adalah data dalam bentuk angka (bilangan) yang diperoleh dengan cara membilang.
c) Data kontinu, adalah data dalam bentuk angka/bilangan yang diperoleh berdasarkan hasil pengukuran.

3.1.2.2 Skala Pengukuran

Skala pengukuran yang digunakan dalam dalam skala statistika yakni nominal, ordinal interval dan rasio.

Skala Nominal

Skala nominal merupakan skala pengukuran yang paling rendah dibandingkan dengan skala lainnya. Skala ini hanya berfungsi untuk membedakan suatu objek atau peristiwa berdasarkan nama atau kategori tanpa memiliki makna kuantitatif. Skala nominal digunakan untuk mengklasifikasikan objek, individu, atau kelompok ke dalam kategori tertentu.

Pemberian angka atau simbol dalam skala nominal tidak memiliki makna numerik, melainkan hanya menunjukkan keberadaan atau ketiadaan suatu atribut atau karakteristik pada objek yang diukur. Sebagai contoh, jenis kelamin dapat dikodekan dengan angka 1 untuk laki-laki dan 2 untuk perempuan. Namun, angka tersebut hanya berfungsi sebagai label kategori dan tidak memiliki makna matematis. Oleh karena itu, kita tidak dapat menyatakan bahwa perempuan memiliki nilai dua kali lebih besar daripada laki-laki. Kode yang digunakan dapat ditukar, misalnya laki-laki diberi kode 2 dan perempuan kode 1, selama setiap kategori memiliki kode yang berbeda.

Angka dalam skala nominal tidak memiliki nilai intrinsik, maka operasi matematika standar seperti penjumlahan, pengurangan, perkalian, atau pembagian tidak dapat diterapkan. Analisis statistik yang sesuai untuk skala nominal adalah metode yang berbasis jumlah dan proporsi, seperti modus, distribusi frekuensi, uji chi-kuadrat, serta teknik statistik non-parametrik lainnya.

Skala Ordinal

Skala ordinal memiliki tingkat yang lebih tinggi dibandingkan skala nominal dan sering disebut sebagai skala peringkat. Hal ini dikarenakan angka atau simbol dalam skala ordinal tidak hanya berfungsi untuk membedakan suatu objek, tetapi juga menunjukkan urutan atau tingkatan berdasarkan karakteristik tertentu.

Sebagai contoh, tingkat kepuasan seseorang terhadap suatu produk dapat diklasifikasikan menggunakan skala berikut: 5 untuk sangat puas, 4 untuk puas, 3 untuk kurang puas, 2 untuk tidak puas, dan 1 untuk sangat tidak puas. Dalam skala ordinal, tidak seperti skala nominal, penggantian angka harus tetap mempertahankan urutan dari yang terbesar ke yang terkecil atau sebaliknya. Oleh karena itu, tidak diperkenankan menukar urutan angka secara sembarangan, misalnya menetapkan 1 untuk sangat puas, 2 untuk tidak puas, 3 untuk puas, dan seterusnya. Urutan harus konsisten, misalnya 1 untuk sangat puas, 2 untuk puas, 3 untuk kurang puas, dan seterusnya.

Sama seperti skala nominal, skala ordinal juga tidak memungkinkan penerapan operasi matematika standar seperti penjumlahan, pengurangan, perkalian, atau pembagian. Analisis statistik yang sesuai untuk skala ordinal adalah metode berbasis jumlah dan proporsi, seperti modus, distribusi frekuensi, uji chi-kuadrat, serta berbagai teknik statistik non-parametrik lainnya.

Interval

Skala interval memiliki karakteristik yang mencakup sifat-sifat skala nominal dan ordinal, dengan tambahan karakteristik berupa interval yang tetap. Dengan demikian, skala interval sudah memiliki nilai intrinsik serta jarak yang dapat diukur, tetapi jarak tersebut belum berbentuk kelipatan.

Pernyataan bahwa "jarak belum merupakan kelipatan" sering kali diartikan sebagai ketiadaan nilai nol mutlak dalam skala interval. Artinya, meskipun terdapat nol dalam skala ini, nilai tersebut bukanlah nol absolut yang menunjukkan ketiadaan suatu karakteristik, melainkan hanya titik acuan dalam pengukuran.

Pada skala interval, angka yang digunakan benar-benar merepresentasikan besaran yang dapat dianalisis secara matematis. Oleh karena itu, berbagai operasi matematika serta metode statistik dapat diterapkan pada skala ini, kecuali metode yang bergantung pada rasio, seperti perhitungan koefisien variasi.

Rasio

Rasio merupakan skala pengukuran dengan tingkat kualitas tertinggi. Skala ini mencakup seluruh karakteristik yang terdapat pada skala nominal, ordinal, dan interval, dengan tambahan sifat adanya nilai nol yang bersifat mutlak.

Nilai nol mutlak dalam skala rasio berarti bahwa nol menunjukkan ketiadaan suatu karakteristik secara absolut dan tidak dapat berubah meskipun menggunakan skala yang berbeda. Oleh karena itu, dalam skala rasio, pengukuran memiliki makna perbandingan atau rasio yang jelas.

Salah satu contoh penggunaan skala rasio adalah dalam pengukuran tinggi dan berat. Sebagai ilustrasi, jika suatu benda memiliki berat 30 kg dan benda lainnya memiliki berat 60 kg, maka dapat dinyatakan bahwa benda kedua memiliki berat dua kali lipat dibandingkan benda pertama.

3.1.3 Sumber Data

Sumber data merupakan segala sesuatu yang memberikan informasi terkait data (Edi Riadi, 2016). Berdasarkan sumbernya, data dibagi menjadi data primer dan data sekunder.

Data Primer

Data primer adalah data informasi yang diperoleh langsung dari objek yang diteliti. Data primer ini bersifat asli dan belum mengalami proses statistik apa pun. Untuk mengumpulkan data primer, peneliti harus mengumpulkan secara langsung melalui teknik observasi, wawancara, diskusi terfokus, dan penyebaran kuesioner. Dalam penelitian, sumber data yang digunakan adalah data primer yang dikumpulkan melalui angket (kuesioner) sebagai penelitian.

Data Sekunder

Data Sekunder merupakan data yang diperoleh secara tidak langsung dari objek penelitian. Data sekunder yang diperoleh melalui sebuah situs internet atau referensi lain yang relevan dengan topik yang sedang diteliti oleh penulis.

3.1.4 Metode Pengumpulan Data

Metode pengumpulan data merupakan metode yang digunakan oleh peneliti dalam memperoleh data untuk penelitiannya (Arikunto, 2012). Adapun beberapa metode pengumpulan data adalah sebagai berikut:

Survei

Survei merupakan metode yang digunakan dalam evaluasi untuk menggambarkan fakta dan karakteristik populasi atau wilayah tertentu secara sistematis, faktual, dan akurat. Metode penelitian kuantitatif seperti survei sering dimanfaatkan untuk memperoleh atau mengumpulkan data dari populasi yang luas, biasanya dengan menggunakan sampel yang lebih kecil. Metode survei ini juga digunakan untuk menyelesaikan masalah atau isu skala besar yang melibatkan populasi sangat besar, sehingga diperlukan ukuran sampel yang cukup besar. Dalam penelitian survei, informasi dikumpulkan dari responden melalui kuesioner.

Eksperimen

Eksperimen adalah salah satu jenis metode penelitian kuantitatif yang bertujuan untuk menguji keefektifan suatu variabel eksperimen. Penelitian ini umumnya lebih sering diterapkan di bidang ilmu eksakta. Terdapat dua jenis penelitian eksperimen, yaitu eksperimen semu dan eksperimen sungguhan.

Metode eksperimen semu digunakan untuk evaluasi guna memperoleh informasi yang mendekati data sebenarnya ketika kondisi tidak memungkinkan untuk mengontrol atau memanipulasi variabel-variabel yang relevan.

Sementara itu, metode eksperimen sungguhan bertujuan untuk mengkaji hubungan sebab-akibat. Hal ini dilakukan dengan memberikan satu atau lebih perlakuan kepada kelompok eksperimen dan membandingkan hasilnya dengan kelompok kontrol yang tidak menerima perlakuan tersebut.

Observasi

Observasi atau pengamatan merupakan metode pengumpulan data yang dilakukan dengan cara mengamati secara langsung objek yang sedang diteliti di lapangan (Apriyanti et al., 2019)..

3.1.5 Penyajian Data

Penyajian data merupakan cara untuk menyusun dan menampilkan data agar lebih mudah dipahami. Berikut beberapa metode penyajian yang umum digunakan:

3.1.5.1 Tabel Frekuensi

Tabel frekuensi yaitu tabel yang menggambarkan atau mencantumkan jumlah kemunculan suatu kejadian atau frekuensi kejadian (Leni, 2017). Tabel ini digunakan untuk menunjukkan jumlah kemunculan suatu data dalam kategori tertentu. Tabel frekuensi biasanya digunakan untuk data diskrit dan dapat dilengkapi dengan frekuensi relatif dan kumulatif.

Contoh Tabel frekuensi:

Data Tinggi badan 20 Siswa (dalam cm):

150, 155, 160, 150, 165, 155, 160, 170, 165, 160,
155, 150, 160, 155, 170, 165, 160, 150, 155, 165

Tabel 3.1: Contoh Tabel frekuensi

Tinggi Badan	Frekuensi
150	4
155	5
160	5
165	4
170	2
Total	**20**

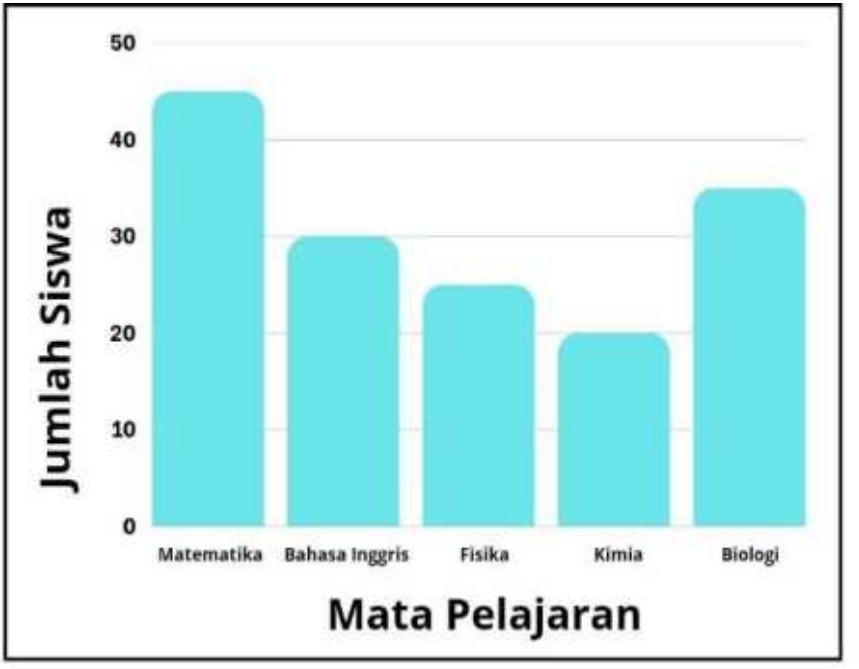
3.1.5.2 Diagram Batang

Diagram batang merupakan cara penyajian data secara visual dengan menggunakan batang berbentuk persegi panjang untuk menggambarkan nilai atau frekuensi dari berbagai kategori. Panjang atau tinggi batang tersebut tersebut mencerminkan nilai yang diwakilinya. Diagram batang sangat berguna untuk membandingkan data antar kategori serta menggambarkan perubahan nilai.

Contoh Diagram Batang

Jumlah siswa yang memilih mata pelajaran di kelas X

- Matematika: 45 Siswa
- Bahasa Inggris: 30 Siswa
- Fisika: 25 Siswa
- Kimia: 20 Siswa
- Biologi: 35 Siswa



Gambar 3.1: Contoh Diagram Batang

3.1.5.3 Histogram

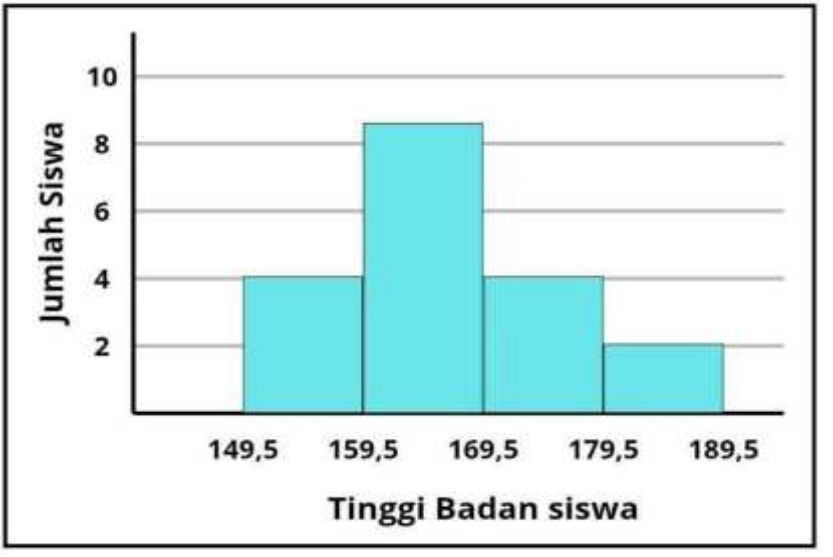
Histogram merupakan jenis diagram yang digunakan untuk menggambarkan distribusi frekuensi suatu data dalam bentuk batang vertikal. Setiap batang pada histogram mewakili sebuah interval (kelas) dan tinggi batang menunjukkan jumlah data yang jatuh dalam interval tersebut.

Histogram sering digunakan untuk menunjukkan seberapa sering suatu nilai muncul dalam kumpulan data kontinu. Histogram bermanfaat untuk melihat pola distribusi data seperti apakah data terpusat ditengah, miring ke kiri atau ke kanan, atau menyebar merata.

Contoh Histogram

Data tinggi badan 20 Siswa Kelas X dalam satuan cm
150,160,155,165,170,160,175,180,165,160,
155,170,160,160,175,160,165,170,155,180.
Berdasarkan data diperoleh interval dengan rentang 10 cm

a. 150 - 159
b. 160 - 169
c. 170 - 179
d. 180 – 189



Gambar 3.2: Contoh Histogram

3.2 Pengukuran Tendensi Sentral dan Dispersi

Pengukuran statistik memiliki peranan penting dalam analisis data. Dua konsep utama dalam statistik deskriptif adalah tendensi sentral dan dispersi. Tendensi sentral mengacu pada ukuran yang menunjukkan nilai tengah atau tipikal dalam sekumpulan data, sedangkan dispersi mengukur seberapa jauh data tersebar di sekitar nilai tengah tersebut (Johnson & Bhattacharyya, 2014).

3.2.1 Pengukuran Tendensi Sentral

Tendensi sentral merangkum kumpulan data dengan satu nilai yang mewakili seluruh distribusi data. Ada tiga ukuran utama tendensi sentral, yaitu (Johnson & Bhattacharyya, 2014):

3.2.1.1 Mean (Rata-rata)

Mean atau rata-rata adalah jumlah seluruh nilai dalam dataset dibagi dengan banyaknya nilai. Rumusnya:

$$\bar{X} = \frac{\sum X_i}{n}$$

Contoh:
Misalkan kita memiliki data nilai ujian 5 siswa: 70, 80, 90, 85, dan 75.

Jadi, rata-rata nilai ujian adalah 80.

$$\bar{X} = \frac{70 + 80 + 90 + 85 + 75}{5} = \frac{400}{5} = 80$$

Kelebihan:

- Mudah dihitung dan dipahami.
- Menggunakan semua nilai dalam dataset.

Kekurangan:

- Sensitif terhadap nilai ekstrem (outlier).

3.2.1.2 Median

Median adalah nilai tengah dari data yang telah diurutkan. Jika jumlah data ganjil, median adalah nilai di tengah. Jika jumlah data genap, median adalah rata-rata dari dua nilai tengah. Rumusnya:

$$Me = X_{(n+1)/2}, \text{ untuk } n \text{ ganjil}$$

$$Me = \frac{X_{n/2} + X_{(n/2)+1}}{2}, \text{ untuk } n \text{ genap}$$

Contoh:
Data: 50, 60, 70, 80, 90 (sudah diurutkan).
Karena jumlah data ganjil (5), median adalah nilai tengah, yaitu 70.
Jika data: 50, 60, 70, 80 (genap), maka median:

$$Me = \frac{60 + 70}{2} = 65$$

3.2.1.3 Modus

Modus adalah nilai yang paling sering muncul dalam dataset. Bisa ada satu modus (unimodal), dua modus (bimodal), atau lebih (multimodal).
Contoh:
Data: 5, 7, 8, 8, 9, 10, 8.
Nilai yang paling sering muncul adalah 8 sehingga modusnya adalah 8.

3.2.2 Pengukuran Dispersi

Dispersi mengukur seberapa tersebar data dalam suatu dataset. Beberapa ukuran utama dispersi adalah:

3.2.2.1 Rentang (*Range*)

Rentang adalah selisih antara nilai maksimum dan minimum dalam dataset:

$$R = X_{max} - X_{min}$$

Contoh:
Data: 15, 20, 35, 40, 50.

$$R = 50 - 15 = 35$$

3.2.2.2 Rata-rata Deviasi Mutlak (Mean Absolute Deviation, MAD)

MAD mengukur rata-rata penyimpangan absolut nilai data dari mean:

$$MAD = \frac{\sum |X_i - \bar{X}|}{n}$$

Contoh:
Data: 10, 20, 30.

$$\bar{X} = \frac{10 + 20 + 30}{3} = 20$$

$$MAD = \frac{|10 - 20| + |20 - 20| + |30 - 20|}{3} = \frac{10 + 0 + 10}{3} = 6.67$$

3.2.2.3 Variansi

Variansi adalah ukuran penyebaran data yang dihitung sebagai rata-rata kuadrat deviasi dari mean:

$$\sigma^2 = \frac{\sum(X_i - \bar{X})^2}{n}$$

Contoh:
Data: 2, 4, 6.

$$\bar{X} = \frac{2 + 4 + 6}{3} = 4$$

$$\sigma^2 = \frac{(2 - 4)^2 + (4 - 4)^2 + (6 - 4)^2}{3} = \frac{4 + 0 + 4}{3} = 2.67$$

3.2.2.4 Simpangan Baku (Standar Deviasi)

Simpangan baku adalah akar dari variansi:

$$\sigma = \sqrt{\sigma^2}$$

Contoh:
Dari variansi sebelumnya (2.67):

$$\sigma = \sqrt{2.67} = 1.63$$

3.2.3 Aplikasi dalam Berbagai Bidang

Pengukuran tendensi sentral dan dispersi memiliki berbagai aplikasi dalam berbagai bidang, terutama dalam analisis data untuk pengambilan keputusan

yang lebih baik. Berikut adalah beberapa contoh penerapan yang lebih mendalam:

1. Ekonomi

Dalam bidang ekonomi, pengukuran ini digunakan untuk memahami distribusi pendapatan, inflasi, dan pertumbuhan ekonomi.

- Rata-rata pendapatan penduduk: Mean digunakan untuk menghitung rata-rata pendapatan di suatu wilayah, membantu pemerintah dalam menentukan kebijakan ekonomi.
- Ketimpangan ekonomi: Dispersi seperti variansi dan simpangan baku dapat mengukur tingkat ketimpangan pendapatan dalam suatu populasi, misalnya dengan menghitung koefisien variasi atau indeks Gini.
- Analisis inflasi: Dengan melihat rata-rata harga barang dalam suatu periode dan standar deviasinya, ekonomi dapat memperkirakan tingkat kestabilan harga.

2. Kesehatan

Dalam dunia medis dan kesehatan masyarakat, statistik ini digunakan untuk menganalisis data pasien dan efektivitas pengobatan.

- Tekanan darah pasien: Mean tekanan darah dalam populasi digunakan untuk mengetahui batas normal, sementara standar deviasi membantu mengidentifikasi pasien dengan tekanan darah abnormal.
- Penyebaran penyakit: Rentang dan variansi digunakan dalam epidemiologi untuk memahami bagaimana penyakit menyebar di populasi yang berbeda.
- Efektivitas obat: Uji klinis sering menggunakan statistik ini untuk mengukur variasi respons pasien terhadap obat baru dibandingkan dengan standar pengobatan.

3. Pendidikan

Dalam bidang pendidikan, pengukuran ini berguna untuk mengevaluasi performa akademik siswa dan efektivitas metode pembelajaran.

- Distribusi nilai ujian: Rata-rata nilai ujian membantu menentukan tingkat kesulitan soal, sementara standar deviasi menunjukkan sebaran kemampuan siswa.

- Perbandingan prestasi siswa: Median sering digunakan untuk membandingkan nilai antar kelompok tanpa terpengaruh oleh outlier.
- Analisis efektivitas kurikulum: Dengan menggunakan variasi nilai sebelum dan sesudah perubahan kurikulum, pendidik dapat mengukur dampaknya terhadap siswa.

4. Manajemen dan Bisnis

Dalam dunia bisnis, pengukuran statistik ini sangat membantu dalam pengambilan keputusan strategis.

- Analisis kepuasan pelanggan: Survei kepuasan pelanggan menggunakan rata-rata dan standar deviasi untuk memahami variasi pengalaman pelanggan terhadap suatu produk atau layanan.
- Pengukuran produktivitas karyawan: Perusahaan menganalisis rata-rata output kerja karyawan dan variansinya untuk menentukan efektivitas strategi manajemen sumber daya manusia.
- Prediksi penjualan: Dengan menggunakan data historis, perusahaan dapat memperkirakan rata-rata penjualan dan fluktuasinya untuk perencanaan produksi dan pemasaran.

5. Teknik dan Ilmu Data

Dalam bidang teknik dan data science, pengukuran ini membantu dalam pemrosesan data, optimasi, dan analisis performa sistem.

- Kontrol kualitas produk: Variansi dalam ukuran atau berat produk di industri manufaktur digunakan untuk memastikan standar kualitas yang konsisten.
- Analisis performa algoritma: Dalam machine learning, mean dan variansi dari error model digunakan untuk mengevaluasi keakuratan prediksi.
- Reliabilitas sistem: Standar deviasi digunakan untuk mengukur stabilitas sistem, seperti waktu respons server dalam dunia IT.

3.3 Probabilitas dan Distribusi Probabilitas

Probabilitas dan distribusi probabilitas adalah konsep fundamental dalam statistik dan analisis data. Probabilitas mengukur kemungkinan terjadinya suatu peristiwa, sedangkan distribusi probabilitas menjelaskan bagaimana

probabilitas tersebar pada berbagai hasil yang mungkin terjadi dalam suatu eksperimen acak. Pemahaman yang baik tentang probabilitas dan distribusinya sangat penting dalam berbagai bidang seperti ekonomi, kesehatan, teknik, dan ilmu komputer.

3.3.1 Konsep Dasar Probabilitas

Probabilitas adalah angka antara 0 dan 1 yang menunjukkan kemungkinan terjadinya suatu kejadian . Probabilitas suatu kejadian A dinotasikan sebagai $P(A)$ dan dihitung menggunakan rumus dasar:

$$P(A) = \frac{n(A)}{n(S)}$$

Di mana:

- $n(A)$ adalah jumlah hasil yang mendukung kejadian A.
- $n(S)$ adalah jumlah total hasil dalam ruang sampel.

3.3.2 Jenis Probabilitas

Probabilitas dapat diklasifikasikan menjadi beberapa jenis:

- **Probabilitas Klasik**: Berdasarkan jumlah hasil yang mungkin terjadi, misalnya peluang muncul angka enam pada dadu adalah $\frac{1}{6}$.
- **Probabilitas Empiris**: Berdasarkan data historis atau observasi, misalnya peluang seorang pelanggan membeli produk setelah melihat iklan.
- **Probabilitas Subjektif**: Berdasarkan opini atau pengalaman seseorang, misalnya seorang analis saham memperkirakan harga saham akan naik.

3.3.3 Aturan Dasar Probabilitas

Beberapa aturan penting dalam probabilitas meliputi:

1. **Aturan Penjumlahan**: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ Digunakan ketika kita mencari probabilitas dari dua kejadian terjadi secara bersamaan.
2. **Aturan Perkalian**: $P(A \cap B) = P(A)P(B|A)$ Digunakan ketika dua kejadian terjadi secara berurutan dan saling bergantung.

3. **Probabilitas Komplemen**: $P(A^c) = 1 - P(A)$ Digunakan untuk mencari peluang kejadian yang tidak terjadi.

3.3.4 Distribusi Probabilitas

Distribusi probabilitas menunjukkan bagaimana probabilitas tersebar pada berbagai hasil yang mungkin terjadi dalam suatu eksperimen acak. Distribusi ini dapat dibedakan menjadi dua jenis utama:

3.3.4.1 Distribusi Probabilitas Diskrit

Distribusi probabilitas diskrit digunakan untuk variabel acak diskrit, yaitu variabel yang hanya memiliki nilai tertentu dan terpisah. Contoh umum dari distribusi ini meliputi:

Distribusi Binomial

Distribusi binomial digunakan ketika suatu percobaan memiliki dua kemungkinan hasil (sukses atau gagal) dan dilakukan berulang kali. Rumus probabilitas distribusi binomial adalah:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Di mana:

- n adalah jumlah percobaan,
- k adalah jumlah keberhasilan,
- p adalah probabilitas keberhasilan dalam satu percobaan.

Contoh: Jika peluang sukses dalam suatu ujian adalah 0,6 dan siswa mengikuti 5 ujian, maka peluang siswa berhasil dalam 3 ujian dihitung menggunakan distribusi binomial.

3.3.4.2 Distribusi Poisson

Distribusi Poisson digunakan untuk menghitung jumlah kejadian dalam interval waktu atau ruang tertentu, dengan rumus:

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

Di mana λ adalah rata-rata jumlah kejadian dalam satu interval.

Contoh: Jika rata-rata jumlah panggilan ke pusat layanan pelanggan adalah 10 per jam, maka peluang menerima tepat 5 panggilan dalam satu jam dapat dihitung dengan distribusi Poisson.

3.3.4.3 Distribusi Probabilitas Kontinu

Distribusi probabilitas kontinu digunakan untuk variabel acak kontinu, yaitu variabel yang dapat memiliki nilai dalam rentang tertentu.

Distribusi Normal

Distribusi normal adalah distribusi yang paling umum digunakan dalam statistik. Distribusi ini memiliki bentuk lonceng simetris dan dinyatakan dengan fungsi kepadatan probabilitas:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Di mana:

- μ adalah mean (rata-rata),
- σ adalah standar deviasi.

Contoh: Dalam ujian nasional, nilai siswa sering mengikuti distribusi normal dengan rata-rata 70 dan standar deviasi 10. Dengan menggunakan distribusi normal, kita bisa menghitung persentase siswa yang mendapat nilai di atas 80.

Distribusi Eksponensial

Distribusi eksponensial digunakan untuk menghitung waktu antara kejadian dalam suatu proses yang terjadi secara acak dengan tingkat kejadian konstan. Fungsi distribusinya:

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0$$

Di mana λ adalah laju kejadian rata-rata.

Contoh: Jika rata-rata waktu antar kedatangan pelanggan di restoran adalah 5 menit, maka peluang seorang pelanggan berikutnya tiba dalam waktu kurang dari 3 menit dapat dihitung menggunakan distribusi eksponensial.

3.4 Aljabar Linear dan Kalkulus untuk Data Science

Aljabar linear dan kalkulus adalah dua cabang matematika yang memiliki peran fundamental dalam Data Science. Aljabar linear digunakan untuk manipulasi data dalam bentuk vektor dan matriks, sementara kalkulus membantu dalam analisis perubahan dan optimasi fungsi. Keduanya sangat penting dalam berbagai aplikasi seperti machine learning, deep learning, dan analisis data.

3.5 Aljabar Linear dalam Data Science

3.5.1 Vektor dan Operasi Dasar

Vektor adalah elemen dasar dalam aljabar linear yang merepresentasikan data dalam ruang berdimensi banyak (Gilbert & Strang, 1991). Sebuah vektor dapat ditulis sebagai:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Operasi dasar pada vektor meliputi:

- **Penjumlahan vektor**: $a + b = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n)$
- **Perkalian skalar**: $a = (ca_1, ca_2, \dots, ca_n)$
- **Produk dot**: $a \cdot b = \sum_{i=1}^{n} a_i b_i$

3.5.2 Matriks dan Transformasi Linear

Matriks adalah kumpulan elemen yang tersusun dalam baris dan kolom, biasanya digunakan untuk merepresentasikan dataset dan transformasi linear dalam machine learning.

Bentuk umum matriks:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Operasi dasar matriks meliputi:

1. Penjumlahan dan Pengurangan Matriks

   Operasi ini dilakukan dengan menambahkan atau mengurangi elemen-elemen yang sesuai dari dua matriks dengan ukuran yang sama.

   Contoh:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

2. Perkalian Matriks

   Perkalian dua matriks dilakukan dengan mengambil dot product dari baris pertama matriks pertama dengan kolom pertama matriks kedua, dan seterusnya.

   Contoh:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$A \times B = \begin{bmatrix} (1 \times 5 + 2 \times 7) & (1 \times 6 + 2 \times 8) \\ (3 \times 5 + 4 \times 7) & (3 \times 6 + 4 \times 8) \end{bmatrix}$$

$$= \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

3. Determinan dan Invers Matriks

   - Determinan Matriks adalah nilai skalar yang dapat digunakan untuk menentukan apakah matriks memiliki invers atau tidak. Untuk matriks 2×2:

$$\det(A) = a_{11}a_{22} - a_{12}a_{21}$$

- Invers Matriks adalah matriks yang jika dikalikan dengan matriks aslinya menghasilkan matriks identitas. Invers dari matriks 2×2 dihitung dengan:

$$A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

Contoh:

Jika $A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$, maka:

$$\det(A) = (2 \times 4) - (3 \times 1) = 8 - 3 = 5$$

$$A^{-1} = \frac{1}{5} \begin{bmatrix} 4 & -3 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 0.8 & -0.6 \\ -0.2 & 0.4 \end{bmatrix}$$

4. Dekomposisi Matriks (SVD, PCA)
   - Singular Value Decomposition (SVD): Metode dekomposisi matriks menjadi tiga matriks: $A = U\Sigma V^T$ di mana U dan V adalah matriks ortogonal dan Σ adalah matriks diagonal berisi singular values. SVD digunakan dalam kompresi data dan reduksi dimensi.
   - Principal Component Analysis (PCA): Teknik yang menggunakan eigenvalues dan eigenvectors untuk menemukan kombinasi fitur yang paling signifikan dalam suatu dataset. PCA sering digunakan dalam machine learning untuk reduksi dimensi dan visualisasi data.

3.5.3 Eigenvalues dan Eigenvectors

Eigenvalues dan eigenvectors banyak digunakan dalam PCA (Principal Component Analysis) dan analisis fitur dalam machine learning. Jika AA adalah matriks persegi, maka eigenvector v dan eigenvalue λ memenuhi persamaan:

$$\mathrm{A}\, v = \lambda v$$

3.5.4 Kalkulus dalam Data Science

3.5.4.1 Diferensiasi dan Gradien

Diferensiasi digunakan untuk menentukan laju perubahan suatu fungsi. Dalam machine learning, diferensiasi digunakan untuk menghitung gradien dalam optimasi model. Jika $f(x)$ adalah fungsi, maka turunannya adalah:

$$\frac{d}{dx}f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Gradien dari fungsi multi-variabel f(x,y)f(x, y) diberikan oleh vektor turunan parsial:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

3.5.4.2 Integral dan Optimasi

Integral digunakan dalam probabilitas dan statistik, seperti distribusi probabilitas kumulatif dan metode Monte Carlo. Integral suatu fungsi diberikan oleh:

$$F(x) = \int f(x)dx$$

Optimasi dalam machine learning sering menggunakan metode **gradient descent**, di mana iterasi dilakukan untuk meminimalkan fungsi kehilangan (loss function):

$$\theta = \theta - \alpha \nabla J(\theta)$$

3.5.4.3 Aplikasi dalam Neural Networks

Dalam deep learning, backpropagation menggunakan diferensial dan gradien untuk memperbarui bobot model. Dengan menggunakan aturan rantai dalam kalkulus, turunan dari fungsi aktivasi seperti ReLU dan sigmoid digunakan dalam proses training model.

3.5.5 Implementasi dalam Data Science

Aljabar linear dan kalkulus diterapkan dalam berbagai aspek Data Science:

- **Regresi Linear**: Menggunakan persamaan matriks untuk menentukan parameter terbaik, mengurangi error dengan metode Ordinary Least Squares (OLS).
- **PCA (Principal Component Analysis)**: Menggunakan eigenvalues dan eigenvectors untuk reduksi dimensi, membantu menghilangkan redundansi data tanpa banyak kehilangan informasi.
- **Gradient Descent**: Menggunakan kalkulus untuk optimasi model machine learning, dengan teknik seperti Stochastic Gradient Descent (SGD) untuk konvergensi lebih cepat.
- **Transformasi Data**: Menggunakan matriks untuk normalisasi dan standardisasi data, seperti menggunakan **mean normalization** dan **min-max scaling** untuk memastikan distribusi data lebih seimbang.
- **Neural Networks**: Menggunakan turunan parsial dan backpropagation dalam optimasi model deep learning, memungkinkan pembaruan bobot yang lebih efisien.

Aljabar linear dan kalkulus merupakan pilar utama dalam Data Science. Dengan memahami konsep vektor, matriks, transformasi linear, diferensiasi, dan optimasi, kita dapat membangun dan mengoptimalkan model machine learning secara lebih efektif. Integrasi kedua konsep ini memungkinkan analisis data yang lebih mendalam dan implementasi algoritma yang lebih efisien dalam dunia nyata.

Bab 4
Pengumpulan dan Persiapan Data

4.1 Metode Pengumpulan Data

Metode pengumpulan data merupakan tahapan krusial dalam suatu penelitian, karena menentukan validitas dan reliabilitas temuan yang diperoleh. Dalam penelitian ini, metode pengumpulan data dilakukan melalui pendekatan yang sistematis guna memastikan bahwa informasi yang dikumpulkan relevan serta dapat dipertanggungjawabkan.

4.1.1 Sumber Data Primer dan Sekunder

Dalam proses pengumpulan data, identifikasi sumber data yang tepat merupakan langkah penting untuk menjamin bahwa data yang dikumpulkan relevan, akurat, dan dapat diandalkan. Sumber data umumnya dibagi menjadi dua kategori utama, yaitu data primer dan data sekunder. Pemahaman tentang perbedaan kedua jenis sumber ini sangat penting untuk menentukan metode pengumpulan data yang sesuai dengan tujuan penelitian.

4.1.1.1 Data Primer

Data primer adalah data yang dikumpulkan langsung oleh peneliti melalui berbagai metode seperti wawancara, survei, eksperimen, atau observasi lapangan. Data ini dihasilkan dari interaksi langsung dengan subjek penelitian dan belum pernah tersedia sebelumnya dalam bentuk yang terstruktur. Keuntungan utama dari data primer adalah kemampuannya untuk disesuaikan dengan kebutuhan spesifik penelitian. Sebagai contoh, survei atau kuesioner dapat dirancang untuk menjawab pertanyaan yang sangat spesifik terkait fenomena tertentu.

Metode pengumpulan data primer meliputi:

- Survei dan Kuesioner: Digunakan untuk memperoleh data kuantitatif dari responden dalam jumlah besar. Survei dapat disebarkan secara langsung

atau melalui platform digital seperti Google Forms atau SurveyMonkey (Groves et al., 2009).

- Wawancara: Dilakukan secara tatap muka, melalui telepon, atau daring untuk menggali informasi kualitatif dari subjek penelitian (Creswell, 2014).
- Observasi: Melibatkan pengamatan langsung terhadap aktivitas atau fenomena tertentu dalam lingkungan alamiah subjek. Metode ini sering digunakan dalam penelitian etnografi atau studi lapangan (Patton, 2002).

Meskipun data primer menawarkan fleksibilitas dan akurasi tinggi, pengumpulannya sering kali memerlukan waktu dan sumber daya yang besar. Oleh karena itu, penggunaannya perlu dipertimbangkan dengan cermat.

4.1.1.2 Data Sekunder

Berbeda dengan data primer, data sekunder merupakan data yang telah dikumpulkan, diolah, dan dipublikasikan oleh pihak lain sebelumnya. Sumber data sekunder meliputi laporan resmi, dokumen pemerintah, jurnal ilmiah, buku referensi, atau data dari lembaga statistik seperti Badan Pusat Statistik (BPS) atau organisasi internasional seperti World Bank. Data ini bersifat lebih mudah diakses dan ekonomis dibandingkan dengan data primer, tetapi penggunaannya sering kali memerlukan evaluasi yang cermat untuk memastikan relevansi dan validitasnya.

Contoh sumber data sekunder meliputi:

- Laporan Statistik: Menyediakan data kuantitatif yang telah dihimpun dari survei besar, seperti sensus penduduk.
- Artikel Ilmiah: Menyediakan data dan analisis dari penelitian sebelumnya yang dapat digunakan sebagai referensi atau titik awal penelitian lebih lanjut (Boslaugh, 2007).
- Basis Data Online: Seperti Google Scholar, Scopus, atau PubMed, yang memberikan akses ke ribuan artikel dan laporan penelitian.
- Keuntungan utama dari data sekunder adalah efisiensi waktu dan biaya. Namun, karena data ini dikumpulkan untuk tujuan yang mungkin berbeda dari penelitian yang sedang dilakukan, peneliti harus memperhatikan potensi bias atau ketidaksesuaian konteks (Johnston, 2017).

4.1.1.3 Data Statik

Data statik adalah jenis data yang sifatnya tetap atau tidak mengalami perubahan setelah dikumpulkan. Data ini biasanya dihasilkan dari pengukuran yang dilakukan pada titik waktu tertentu dan bersifat final. Contoh dari data statik meliputi hasil sensus penduduk, data hasil survei lapangan yang dilakukan dalam periode tertentu, atau arsip data masa lalu. Misalnya, hasil sensus tahun 2020 akan tetap relevan untuk penelitian yang berfokus pada kondisi penduduk saat itu tanpa perlu pembaruan.

Karakteristik data statik meliputi:

- Tidak berubah seiring waktu
  Data ini merepresentasikan suatu kondisi pada titik waktu tertentu, misalnya laporan penjualan triwulan tertentu.
- Mudah diarsipkan dan direferensikan
  Karena sifatnya tetap, data statik dapat digunakan sebagai rujukan jangka panjang tanpa risiko perubahan isi.
- Digunakan untuk analisis retrospektif
  Peneliti sering menggunakan data statik untuk memahami tren historis atau menguji hubungan antara variabel di masa lalu (Biemer & Lyberg, 2003).

Keunggulan data statik adalah stabilitasnya, sehingga peneliti dapat menganalisis tanpa khawatir akan adanya perubahan data selama proses penelitian berlangsung. Namun, data statik memiliki keterbatasan dalam konteks penelitian yang membutuhkan pembaruan data secara terus-menerus.

4.1.1.4 Data Dinamis

Berbeda dengan data statik, data dinamis adalah data yang terus diperbarui seiring waktu. Data ini dihasilkan dari proses yang berlangsung secara kontinu atau dalam interval waktu tertentu. Contoh dari data dinamis adalah data harga saham harian, data cuaca, atau data lalu lintas jaringan internet yang dikumpulkan secara real-time.

Karakteristik data dinamis meliputi:

- Berubah seiring waktu: Data ini menggambarkan fenomena yang bersifat dinamis atau terus berkembang. Misalnya, data pemakaian energi rumah tangga yang dikumpulkan secara harian.
- Memerlukan penyegaran berkala: Data dinamis harus diperbarui secara real-time atau periodik untuk mempertahankan relevansi.

- Digunakan untuk analisis prediktif dan monitoring: Karena sifatnya yang terus diperbarui, data dinamis sangat berguna dalam model prediksi dan deteksi perubahan pola (Gama, 2010).

Salah satu tantangan dalam penggunaan data dinamis adalah bagaimana mengelola dan memproses data dalam jumlah besar yang terus bertambah (big data). Hal ini sering membutuhkan infrastruktur teknologi seperti basis data terdistribusi atau cloud computing.

Tabel 4.1: Perbedaan Utama antara Data Statik dan Dinamis

Kriteria	Data Statik	Data Dinamis
Sifat perubahan	Tetap setelah dikumpulkan	Terus diperbarui seiring waktu
Contoh	Data sensus penduduk, laporan keuangan tahunan	Harga saham, data cuaca, data IoT
Kebutuhan pembaruan	Tidak memerlukan pembaruan	Membutuhkan pembaruan berkala atau real-time
Penggunaan utama	Analisis retrospektif, pengambilan keputusan jangka panjang	Monitoring real-time, prediksi, analisis perubahan jangka pendek

4.1.1.5 Bagaimana Memilih Antara Data Statik dan Dinamis

Pemilihan antara data statik dan dinamis bergantung pada tujuan dan konteks penelitian. Beberapa faktor yang perlu dipertimbangkan meliputi:

1. Tujuan Penelitian
    - Jika penelitian bersifat retrospektif atau membutuhkan analisis historis, data statik lebih sesuai.
    - Sebaliknya, untuk penelitian yang memerlukan pemantauan berkelanjutan atau prediksi waktu nyata, data dinamis adalah pilihan terbaik.

2. Ketersediaan Sumber Daya
   - Pengolahan data dinamis sering kali memerlukan infrastruktur dan teknologi yang canggih untuk menangani volume data besar dan pembaruan real-time.
   - Data statik cenderung lebih sederhana dan lebih hemat sumber daya dalam hal pengolahan.
3. Ketepatan Waktu

   Jika hasil penelitian tidak memerlukan data terbaru, maka data statik sudah cukup untuk memenuhi kebutuhan penelitian.

   Namun, jika hasil harus responsif terhadap perubahan kondisi saat ini, data dinamis menjadi sangat penting.

Contoh Aplikasi

- Data Statik: Digunakan dalam studi demografi yang menganalisis perubahan populasi berdasarkan data sensus beberapa dekade terakhir.
- Data Dinamis: Diterapkan dalam algoritma trading saham yang memantau pergerakan harga secara real-time untuk membuat keputusan beli/jual secara otomatis.

4.1.2 Pengumpulan Data Online

Dalam era digital, internet telah menjadi sumber data yang sangat kaya, beragam, dan mudah diakses. Pengumpulan data dari internet memungkinkan peneliti mengakses informasi yang tidak terbatas oleh ruang dan waktu. Data dapat berupa teks, angka, gambar, atau video, dan mencakup berbagai bidang seperti sosial, ekonomi, pendidikan, dan kesehatan. Ada tiga metode utama yang sering digunakan dalam pengumpulan data online, yaitu web scraping, pemanfaatan API (Application Programming Interface), dan akses ke basis data online. Ketiga metode ini memiliki keunggulan dan tantangan masing-masing yang perlu dipahami oleh peneliti sebelum menentukan metode yang paling sesuai.

4.1.2.1 Web Scraping

Web scraping adalah metode pengumpulan data dari halaman web secara otomatis menggunakan perangkat lunak atau skrip tertentu. Teknik ini memungkinkan peneliti untuk mengekstrak informasi yang tersimpan dalam berbagai elemen web, seperti teks artikel, tabel data, atau metadata gambar.

Metode ini menjadi pilihan populer dalam pengumpulan data besar (big data) karena efisiensinya dalam mengumpulkan data dalam jumlah besar dalam waktu yang relatif singkat.

Proses web scraping dimulai dengan mengidentifikasi struktur HTML dari halaman web yang menjadi target. Struktur HTML ini berisi elemen-elemen seperti div, span, atau p yang mengatur tampilan dan posisi konten di halaman. Setelah elemen yang berisi data target diidentifikasi, program scraping akan mengekstrak informasi tersebut dan menyimpannya dalam format yang lebih terstruktur, seperti CSV atau JSON.

Salah satu keunggulan utama web scraping adalah kemampuannya untuk mengumpulkan data dari berbagai sumber secara cepat dan otomatis. Sebagai contoh, perusahaan ritel dapat menggunakan web scraping untuk mengumpulkan informasi harga produk dari situs pesaing dalam waktu nyata. Namun, teknik ini tidak bebas dari tantangan. Perubahan struktur situs web dapat membuat program scraping menjadi tidak berfungsi, dan dalam beberapa kasus, kegiatan ini dapat melanggar kebijakan privasi atau hukum yang berlaku (Krotov & Silva, 2018).

4.1.2.2 Pemanfaatan API

API (Application Programming Interface) adalah antarmuka yang memungkinkan aplikasi atau program berkomunikasi dan berbagi data dengan sistem lain. API memberikan akses langsung ke data yang sudah terstruktur dan disimpan di server penyedia layanan, tanpa memerlukan interaksi langsung dengan antarmuka pengguna seperti pada web scraping. Banyak perusahaan teknologi besar menyediakan API untuk mengakses data mereka, seperti Twitter API untuk mengumpulkan data media sosial, atau Google Maps API untuk mendapatkan informasi lokasi.

Proses kerja API melibatkan pengiriman permintaan (*request*) dari pengguna ke server API menggunakan protokol HTTP. Permintaan ini dapat berupa instruksi untuk mengambil data tertentu, misalnya jumlah tweet yang mengandung kata kunci tertentu dalam rentang waktu tertentu. Setelah menerima permintaan, server API akan mengembalikan data yang diminta dalam format yang dapat dengan mudah diolah, seperti JSON atau XML.

Keunggulan utama dari pemanfaatan API adalah keandalannya dalam menyediakan data yang sudah terstruktur dan disaring sesuai kebutuhan pengguna. Selain itu, API biasanya dirancang untuk mendukung integrasi yang mulus dengan berbagai bahasa pemrograman, seperti Python atau JavaScript,

sehingga memudahkan peneliti dalam mengotomatisasi proses pengumpulan data.

Namun, API juga memiliki keterbatasan. Sebagian besar penyedia API menetapkan rate limit, yaitu batas jumlah permintaan yang dapat dikirim dalam periode waktu tertentu. Batasan ini dimaksudkan untuk mencegah beban berlebih pada server, tetapi juga bisa menjadi kendala bagi peneliti yang memerlukan data dalam jumlah besar atau waktu nyata. Beberapa API juga memerlukan biaya berlangganan untuk mengakses data premium atau fitur tambahan (Boslaugh, 2007).

4.1.2.3 Basis Data Online

Metode pengumpulan data online lainnya adalah melalui akses ke basis data online yang disediakan oleh berbagai institusi, baik pemerintah, akademik, maupun komersial. Basis data online merupakan kumpulan data yang disimpan di server dan dapat diakses melalui portal web atau sistem manajemen basis data. Beberapa basis data menyediakan data yang dapat diunduh dalam format terstruktur, seperti CSV atau Excel, sementara yang lain memerlukan kueri SQL untuk mengakses data tertentu.

Contoh basis data yang sering digunakan dalam penelitian meliputi:

- Google Scholar: Menyediakan akses ke artikel ilmiah, jurnal, dan konferensi akademik.
- World Bank Open Data: Menyediakan data statistik global terkait pembangunan ekonomi dan sosial.
- PubMed: Basis data yang berfokus pada penelitian biomedis dan kesehatan.

Keunggulan dari basis data online adalah ketersediaan data yang terverifikasi dan telah melalui proses validasi oleh penyedia data. Hal ini memberikan kepercayaan tinggi terhadap kualitas data yang diperoleh. Namun, tidak semua basis data bersifat terbuka. Beberapa basis data memerlukan lisensi atau biaya langganan untuk mengakses data tertentu. Selain itu, data yang tersedia mungkin tidak selalu terkini, terutama jika penyedia data memiliki keterbatasan dalam memperbarui konten mereka.

4.1.2.4 Etika dan Legalitas dalam Pengumpulan Data Online

Meskipun pengumpulan data online menawarkan fleksibilitas dan akses ke berbagai jenis data, penting bagi peneliti untuk mempertimbangkan aspek etika dan legalitas. Misalnya, penggunaan web scraping dapat melanggar hukum

perlindungan data jika dilakukan tanpa izin. Oleh karena itu, peneliti perlu memeriksa kebijakan privasi situs web atau layanan sebelum melakukan pengumpulan data. Selain itu, jika menggunakan data dari basis data online berlisensi, pastikan untuk mematuhi syarat dan ketentuan yang ditetapkan.

Beberapa langkah etis yang dapat diambil meliputi:

- Memastikan bahwa data yang dikumpulkan tidak melanggar privasi individu.
- Menghindari scraping data sensitif tanpa izin.
- Menghormati batasan akses dan rate limit yang ditetapkan oleh penyedia API.

4.1.3 Data Eksperimental dan Observasi

Dalam penelitian, pengumpulan data dapat dilakukan melalui berbagai metode, dua di antaranya yang sering digunakan adalah metode eksperimen langsung dan metode observasi. Kedua metode ini memiliki karakteristik yang berbeda dalam pengumpulan, jenis data yang dihasilkan, serta tujuan penggunaannya dalam penelitian. Dalam banyak kasus, peneliti dapat menggunakan salah satu atau kombinasi dari keduanya untuk mendapatkan hasil penelitian yang lebih komprehensif.

4.1.3.1 Data Eksperimental

Metode eksperimen adalah metode pengumpulan data yang melibatkan pengendalian dan manipulasi variabel tertentu untuk mengamati efek atau hasil dari perubahan tersebut. Data yang dihasilkan dari metode ini disebut data eksperimental, dan metode ini banyak digunakan dalam berbagai disiplin ilmu, termasuk sains, teknik, psikologi, dan ekonomi.

Pada metode eksperimen, peneliti mengatur kondisi penelitian secara sistematis dan mengubah satu atau beberapa variabel bebas (*independent variables*) untuk melihat pengaruhnya terhadap variabel terikat (*dependent variables*). Hal ini memungkinkan peneliti untuk mengidentifikasi hubungan sebab-akibat dengan lebih akurat. Misalnya, dalam sebuah penelitian medis, peneliti dapat mengontrol dosis obat tertentu yang diberikan kepada pasien untuk melihat dampaknya terhadap tingkat kesembuhan.

Ciri-ciri utama eksperimen meliputi:

- Manipulasi variabel
  Peneliti secara aktif mengubah variabel bebas untuk mengamati hasilnya.

- Pengendalian lingkungan
  Faktor-faktor eksternal yang dapat mempengaruhi hasil dijaga seminimal mungkin agar tidak menimbulkan bias.
- Pengulangan (*replicability*)
  Eksperimen dapat diulang di bawah kondisi yang sama untuk memverifikasi hasil.

Contoh metode eksperimen:

- Eksperimen laboratorium:
  Dilakukan dalam lingkungan terkontrol seperti laboratorium untuk memastikan bahwa variabel luar tidak mempengaruhi hasil. Misalnya, eksperimen tentang daya tahan material di bawah suhu ekstrem.
- Eksperimen lapangan
  Dilakukan dalam situasi nyata tetapi dengan kontrol tertentu. Contoh klasik adalah eksperimen pemasaran di mana harga produk diubah untuk mengukur pengaruhnya terhadap keputusan pembelian konsumen.
- Eksperimen alami
  Variabel bebas dikendalikan oleh alam atau faktor eksternal di luar kendali peneliti. Misalnya, meneliti efek bencana alam terhadap perilaku migrasi penduduk (Creswell, 2014).

Keunggulan dari metode eksperimen adalah kemampuannya dalam menentukan hubungan sebab-akibat secara langsung. Namun, metode ini juga memiliki keterbatasan, terutama ketika berhadapan dengan situasi kompleks yang tidak dapat dikontrol sepenuhnya. Selain itu, eksperimen dapat menjadi mahal dan memerlukan waktu yang lama, terutama jika melibatkan banyak subjek atau variabel.

4.1.3.2 Data Observasi

Berbeda dengan metode eksperimen, metode observasi adalah proses pengumpulan data dengan cara mengamati dan mencatat fenomena atau perilaku subjek penelitian secara langsung di lingkungan aslinya tanpa intervensi aktif dari peneliti. Data yang dikumpulkan disebut data observasi, dan metode ini sering digunakan dalam penelitian kualitatif maupun kuantitatif.

Observasi dapat dilakukan secara langsung (mengamati subjek secara fisik) atau tidak langsung (melalui perangkat rekaman video, audio, atau digital). Metode ini sangat berguna dalam penelitian yang bertujuan untuk memahami perilaku

alami, interaksi sosial, atau kondisi lingkungan tertentu yang tidak bisa dimanipulasi secara eksperimen.

Ciri-ciri utama metode observasi meliputi:

- Tanpa manipulasi
  Peneliti tidak mengubah variabel apa pun; hanya mencatat apa yang terjadi secara alami.
- Konteks alami
  Observasi dilakukan di lingkungan asli tempat fenomena terjadi.
- Bisa bersifat partisipatif atau non-partisipatif
  Dalam observasi partisipatif, peneliti turut serta dalam aktivitas subjek yang diamati. Sebaliknya, dalam observasi non-partisipatif, peneliti hanya mengamati tanpa terlibat.

Jenis-jenis observasi meliputi:

- Observasi partisipatif
  Peneliti terlibat dalam aktivitas subjek yang diamati untuk memahami konteks dari dalam. Misalnya, antropolog yang hidup bersama komunitas adat untuk memahami budaya mereka.
- Observasi non-partisipatif
  Peneliti hanya mengamati dari luar tanpa terlibat dalam kegiatan subjek. Contohnya, mengamati perilaku anak-anak di taman bermain tanpa ikut serta dalam permainan.
- Observasi terstruktur
  Peneliti menggunakan panduan observasi yang telah dirancang sebelumnya untuk mencatat kejadian yang relevan.
- Observasi tidak terstruktur
  Peneliti mengamati tanpa panduan khusus, membiarkan data mengalir secara alami. Hal ini umum dalam penelitian eksploratif atau studi kasus awal (Patton, 2002).

Metode observasi memiliki keunggulan dalam memahami fenomena secara mendalam dan kontekstual. Namun, metode ini juga memiliki beberapa tantangan, seperti adanya bias observasi, kesulitan mencatat data secara lengkap dalam situasi yang kompleks, serta potensi perubahan perilaku subjek yang sadar sedang diamati (disebut efek Hawthorne).

Tabel 4.2: Perbandingan Metode Eksperimen dan Observasi

Aspek	Metode Eksperimen	Metode Observasi
Manipulasi variabel	Peneliti mengontrol dan memanipulasi variabel	Tidak ada manipulasi, hanya pengamatan
Lingkungan	Dilakukan dalam lingkungan terkontrol atau semi-terkontrol	Dilakukan di lingkungan alami
Tujuan utama	Menentukan hubungan sebab-akibat	Memahami fenomena dalam konteks tertentu
Jenis data	Data kuantitatif yang terstruktur	Data kualitatif atau kuantitatif, tergantung metode observasi
Kelemahan	Biaya tinggi, tidak selalu merepresentasikan kondisi nyata	Bias observasi, perubahan perilaku subjek

Pemilihan antara eksperimen dan observasi bergantung pada tujuan penelitian dan kondisi yang dihadapi. Jika tujuan utama adalah untuk menentukan hubungan sebab-akibat secara eksplisit, metode eksperimen menjadi pilihan utama. Sebaliknya, jika peneliti ingin memahami perilaku alami atau situasi kompleks yang sulit dikontrol, metode observasi lebih cocok digunakan.

Dalam banyak kasus, kombinasi antara kedua metode ini memberikan hasil yang lebih kaya dan akurat. Misalnya, eksperimen awal dapat dilakukan untuk menguji hipotesis, sementara observasi lanjutan digunakan untuk memperkuat hasil dan menempatkannya dalam konteks nyata.

4.2 Teknik Sampling dan Cleaning Data

Teknik sampling dan cleaning data adalah dua langkah fundamental dalam pengumpulan dan pengolahan data yang bertujuan untuk memastikan bahwa data yang dianalisis merepresentasikan populasi secara akurat dan bebas dari kesalahan. Teknik sampling menentukan cara memilih subset data yang mewakili keseluruhan populasi, sementara cleaning data berfokus pada penghapusan atau koreksi kesalahan seperti data yang hilang, duplikasi, atau data yang tidak valid. Kombinasi dari kedua proses ini berkontribusi pada peningkatan kualitas dan keakuratan hasil analisis.

4.2.1 Teknik Sampling Data

Sampling adalah proses memilih sebagian data dari keseluruhan populasi untuk dianalisis, dengan tujuan membuat kesimpulan yang dapat digeneralisasikan. Ada berbagai metode sampling yang umum digunakan, antara lain random sampling, stratified sampling, dan systematic sampling. Setiap metode memiliki kelebihan dan kekurangannya sendiri, tergantung pada karakteristik populasi dan tujuan penelitian.

4.2.1.1 Random Sampling (Pengambilan Sampel Acak)

Random sampling adalah metode pengambilan sampel di mana setiap elemen dalam populasi memiliki peluang yang sama untuk dipilih. Metode ini dianggap sebagai metode sampling yang paling murni dan bebas bias karena pemilihan dilakukan secara acak. Misalnya, jika terdapat 1.000 siswa di sekolah, random sampling akan memilih siswa tersebut tanpa mempertimbangkan faktor-faktor demografis seperti usia atau jenis kelamin. Keunggulan metode ini adalah kemampuannya menghasilkan sampel yang representatif dalam populasi besar. Namun, random sampling bisa menjadi tidak efisien jika populasi sangat heterogen, karena variasi yang besar dalam populasi dapat menyebabkan hasil sampel kurang akurat (Lohr, 2010).

4.2.1.2 Stratified Sampling (Pengambilan Sampel Berstrata)

Stratified sampling digunakan ketika populasi memiliki subkelompok (strata) yang berbeda secara signifikan. Metode ini membagi populasi ke dalam strata-strata berdasarkan karakteristik tertentu, seperti jenis kelamin, kelompok umur, atau tingkat pendidikan. Setelah strata ditentukan, sampel diambil secara acak dari setiap strata. Keunggulan utama dari stratified sampling adalah kemampuannya menghasilkan sampel yang lebih representatif dibandingkan random sampling sederhana, terutama ketika perbedaan antarstrata signifikan. Misalnya, dalam survei rumah tangga nasional, stratified sampling dapat memastikan bahwa rumah tangga dari setiap wilayah geografis terwakili dengan proporsi yang sesuai (Thompson, 2012).

4.2.1.3 Systematic Sampling (Pengambilan Sampel Sistematis)

Systematic sampling adalah metode di mana elemen-elemen dipilih secara berkala dari daftar populasi. Proses ini dimulai dengan menentukan interval sampling, misalnya setiap elemen ke-5 dari daftar dipilih. Interval ini dihitung

dengan membagi ukuran populasi dengan ukuran sampel yang diinginkan. Metode ini sangat efisien dan mudah diterapkan, terutama jika data tersedia dalam bentuk daftar atau database. Namun, kelemahan dari systematic sampling adalah adanya potensi bias jika pola tertentu dalam populasi bertepatan dengan interval sampling. Misalnya, jika daftar populasi diurutkan berdasarkan pola tertentu (seperti abjad atau urutan waktu), hasil sampling mungkin tidak sepenuhnya acak (Cochran, 1977).

4.2.2 Identifikasi Outlier dan Data Hilang

Dalam pengolahan data, outlier dan data **yang** hilang merupakan dua tantangan utama yang dapat memengaruhi akurasi hasil analisis. Outlier adalah data yang berada jauh di luar pola umum atau distribusi utama data, sementara data hilang terjadi ketika nilai data tidak tercatat atau tidak tersedia dalam dataset. Mengidentifikasi dan menangani keduanya secara efektif sangat penting untuk mengurangi bias dan meningkatkan validitas hasil penelitian.

4.2.2.1 Identifikasi Outlier

Outlier dapat disebabkan oleh berbagai faktor, seperti kesalahan pengukuran, kesalahan entri data, atau fenomena langka yang memang terjadi dalam data. Kehadiran outlier bisa memberikan wawasan penting atau justru menciptakan distorsi dalam hasil analisis.

Metode umum untuk mendeteksi outlier meliputi:

1. Pendekatan Statistik
   - Metode Z-score: Menghitung jarak setiap data terhadap mean dalam satuan standar deviasi. Data yang memiliki Z-score lebih besar dari ±3 sering dianggap sebagai outlier.

     $$Z = (\mathrm{X} - \mu)\sigma\mathrm{Z} = \frac{(\mathrm{X} - \mu)}{\sigma}$$

     Di mana μ adalah mean dan σ adalah standar deviasi (Aggarwal, 2015).
   - Metode IQR (Interquartile Range): Data yang terletak di luar kisaran $Q1 - 1.5 \times IQR\mathrm{Q}$ atau $RQ3 + 1.5 \times IQR$ dianggap outlier, di mana IQR adalah selisih antara kuartil ke-3 (Q3) dan kuartil ke-1 (Q1).

     $$IQR = Q3 - Q1$$

2. Visualisasi Data
   - Boxplot: Alat sederhana yang menampilkan distribusi data dan menyoroti outlier sebagai titik di luar whisker.
   - Scatterplot: Berguna untuk mendeteksi outlier dalam data dua dimensi atau lebih.
3. Deteksi Berdasarkan Model

   Menggunakan model prediktif untuk mendeteksi data yang menyimpang dari prediksi normal, seperti model regresi atau clustering.

Setelah outlier diidentifikasi, langkah penanganannya bergantung pada penyebabnya:

- Menghapus outlier
  Jika diketahui bahwa outlier disebabkan oleh kesalahan teknis atau entri data, data tersebut bisa dihapus.
- Mengganti dengan estimasi
  Dalam beberapa kasus, nilai outlier dapat diganti dengan nilai estimasi, seperti mean atau median.
- Menganalisis secara terpisah
  Jika outlier dianggap sebagai fenomena penting, bisa dilakukan analisis terpisah untuk memahami penyebab dan dampaknya.

4.2.2.2 Identifikasi dan Penanganan Data Hilang

Data hilang terjadi ketika satu atau lebih nilai dalam dataset tidak tersedia. Hal ini dapat disebabkan oleh kegagalan pengukuran, kesalahan entri data, atau pengabaian responden dalam survei. Keberadaan data hilang dapat mengurangi akurasi hasil analisis jika tidak ditangani dengan benar.

Metode identifikasi data hilang meliputi:

- Inspeksi visual
  Menganalisis nilai kosong atau kosong yang ditandai dengan simbol tertentu seperti `NA`, `NULL`, atau `0`.
- Fungsi statistik
  Menggunakan fungsi bawaan dalam perangkat lunak analisis data seperti `isnull()` di Python atau `is.na()` di R untuk mendeteksi missing values.

Setelah data hilang diidentifikasi, langkah-langkah penanganannya meliputi:

1. Menghapus Data Hilang (*Listwise Deletion*)

   Menghapus seluruh baris atau kolom yang mengandung nilai hilang. Metode ini mudah diterapkan tetapi hanya disarankan jika jumlah data hilang sangat sedikit.
2. Pengisian (*Imputation*) dengan Nilai Statistik
   - Mean/Median Imputation
     Mengganti data hilang dengan nilai rata-rata atau median dari variabel tersebut. Cocok untuk variabel dengan distribusi normal.
   - Mode Imputation
     Digunakan untuk variabel kategorikal, di mana nilai yang paling sering muncul digunakan sebagai pengganti.
3. Pengisian Berdasarkan Model
   Menggunakan model prediktif seperti regresi untuk memprediksi nilai yang hilang berdasarkan variabel lain.
4. Multiple Imputation

   Metode yang lebih kompleks di mana beberapa set nilai imputed dibuat berdasarkan distribusi data, kemudian hasil analisis dari setiap set digabungkan untuk memperoleh hasil yang lebih akurat (Rubin, 2004).
5. Penanganan Khusus untuk Data Hilang pada Waktu (*Time Series*)
   - Forward Fill dan Backward Fill
     Mengisi nilai yang hilang dengan data sebelum atau sesudahnya dalam rangkaian waktu.
   - Interpolasi
     Menggunakan metode matematis untuk memperkirakan nilai yang hilang berdasarkan titik data yang diketahui.

4.2.3 Data Cleaning dan Validasi

Data cleaning adalah proses membersihkan dataset dari kesalahan, ketidakkonsistenan, dan data yang tidak valid sebelum dilakukan analisis lebih lanjut. Langkah-langkah dalam data cleaning meliputi penghapusan duplikasi, pengisian nilai kosong, dan pengoreksian kesalahan penulisan, yang semuanya bertujuan untuk meningkatkan kualitas data dan hasil analisis.

Salah satu langkah awal yang umum dilakukan adalah penghapusan duplikasi data. Duplikasi bisa terjadi ketika data yang sama tercatat lebih dari satu kali,

biasanya akibat kesalahan entri atau integrasi dari berbagai sumber data. Menghapus entri duplikat sangat penting untuk menghindari bias dalam perhitungan statistik. Langkah berikutnya adalah mengisi nilai kosong (*missing values)* yang ditemukan selama proses analisis. Hal ini bisa dilakukan dengan berbagai metode, seperti mengganti nilai kosong dengan rata-rata (mean), median, atau metode imputation yang lebih kompleks (yang akan dibahas pada sub-bab berikut).

Selain itu, data yang mengandung kesalahan penulisan juga perlu diperbaiki. Kesalahan penulisan dapat menyebabkan variasi yang tidak perlu dalam kategori data, seperti "Yogyakarta" yang ditulis sebagai "Jogja" atau "Yk". Penyatuan atau koreksi semantik ini dapat dilakukan menggunakan skrip pemrograman otomatis atau dengan pendekatan manual jika data berukuran kecil. Proses terakhir adalah validasi data, di mana dataset diuji untuk memastikan bahwa tidak ada kesalahan tambahan dan bahwa data tersebut memenuhi aturan dan logika yang telah ditetapkan, seperti format tanggal yang konsisten atau angka yang berada dalam rentang yang wajar.

Dengan melakukan data cleaning yang efektif, peneliti dapat meminimalkan kesalahan analisis dan memastikan bahwa hasil akhir penelitian didasarkan pada data yang akurat dan bersih (Rahm & Do, 2000).

4.2.4 Data Imputation dan Missing Values

Data imputation adalah proses mengganti nilai yang hilang dengan estimasi yang dapat diterima sehingga dataset tetap lengkap dan dapat dianalisis dengan akurat. Data hilang yang dibiarkan tanpa penanganan dapat menyebabkan bias dalam hasil analisis, mengurangi akurasi model prediktif, atau bahkan membuat algoritma pembelajaran mesin tidak dapat berjalan. Oleh karena itu, teknik-teknik pengisian data yang hilang memainkan peran penting dalam tahap persiapan data.

Berbagai pendekatan statistik sederhana dapat digunakan untuk mengisi data yang hilang, seperti mean imputation dan median imputation. Mean imputation menggantikan nilai yang hilang dengan rata-rata dari seluruh nilai pada variabel tersebut. Metode ini efektif jika data terdistribusi normal, tetapi dapat menjadi bias jika terdapat outlier. Di sisi lain, median imputation lebih tahan terhadap outlier karena menggunakan nilai tengah dari data. Untuk variabel kategorikal, mode imputation digunakan, di mana nilai yang paling sering muncul di dataset menjadi pengganti nilai yang hilang.

Untuk situasi yang lebih kompleks, metode imputation berbasis model prediktif dapat digunakan. Salah satu pendekatan populer adalah regression imputation, di mana model regresi dibangun untuk memprediksi nilai yang hilang berdasarkan variabel lainnya dalam dataset. Misalnya, jika nilai usia seorang responden hilang, model regresi dapat memprediksi usia berdasarkan variabel seperti pendidikan, pekerjaan, atau lokasi tempat tinggal. Metode lain yang lebih canggih adalah multiple imputation, di mana beberapa estimasi nilai yang hilang dihasilkan, dan hasil akhirnya diambil rata-rata dari semua estimasi untuk mengurangi bias dan ketidakpastian (Rubin, 2004).

Dalam konteks data deret waktu (*time series*), metode interpolasi juga sering digunakan untuk memperkirakan nilai yang hilang di antara dua titik waktu yang diketahui. Forward fill mengisi nilai yang hilang dengan nilai sebelumnya, sementara backward fill menggunakan nilai berikutnya. Metode ini sangat berguna dalam data kontinu seperti sensor IoT atau pengukuran suhu.

4.3 Transformasi dan Normalisasi Data

Transformasi dan normalisasi data merupakan langkah penting dalam proses analisis data, terutama dalam pembelajaran mesin dan analitik statistik. Langkah ini dilakukan untuk memastikan bahwa data berada dalam skala yang sesuai, distribusi yang stabil, dan tidak rentan terhadap variabel-variabel dominan yang dapat mempengaruhi hasil analisis. Proses ini melibatkan berbagai teknik, seperti standarisasi, normalisasi, encoding data kategorikal, transformasi non-linear, dan reduksi dimensi, yang dirancang untuk meningkatkan akurasi dan efisiensi algoritma analisis data.

4.3.1 Standarisasi dan Normalisasi

Standarisasi dan normalisasi adalah dua metode yang berbeda, meskipun keduanya bertujuan untuk menyelaraskan skala data.

Standarisasi adalah teknik transformasi data ke distribusi dengan rata-rata (mean) = 0 dan variansi (varian) = 1. Proses ini penting untuk algoritma yang peka terhadap skala data, seperti regresi linier dan metode berbasis jarak (misalnya KNN dan K-means clustering). Formula umum standarisasi adalah:

$$Z = \frac{(X - \mu)}{\sigma}$$

Di mana μ\mu adalah mean dan σ\sigma adalah standar deviasi data. Standarisasi cocok digunakan ketika data memiliki distribusi normal atau mendekati normal.

Normalisasi, di sisi lain, mengubah data sehingga semua nilai berada dalam skala antara [0, 1] atau kadang-kadang [-1, 1]. Teknik ini penting jika data memiliki berbagai rentang yang besar dan jika algoritma membutuhkan input dalam skala tertentu, seperti jaringan saraf dan algoritma berbasis gradien. Formula normalisasi umum adalah:

$$X' = \frac{(X - X_{\min})}{(X_{\max} - X_{\min})}$$

Perbedaan utama antara keduanya adalah bahwa standarisasi bekerja pada distribusi data dengan mempertimbangkan statistik deskriptif (mean dan standar deviasi), sedangkan normalisasi bekerja pada rentang nilai absolut.

4.3.2 Encoding Data Kategorikal

Dalam dataset yang berisi variabel kategorikal, seperti warna, jenis kelamin, atau status pekerjaan, encoding data diperlukan agar data tersebut dapat digunakan dalam algoritma pembelajaran mesin yang hanya bekerja dengan angka. Berikut tabel perbandingan untuk metode encoding data kategorikal:

Tabel 4.3: Metode encoding data kategorikal

Metode Encoding	Deskripsi	Kelebihan	Kekurangan	Contoh
One-Hot Encoding	Mengonversi setiap kategori unik menjadi variabel dummy biner (0 dan 1).	Tidak memberikan urutan antar-kategori sehingga cocok untuk kategori non-hierarkis.	Dapat menyebabkan peningkatan dimensi data (curse of dimensionality) jika kategori sangat banyak.	Merah → [1, 0, 0], Hijau → [0, 1, 0], Biru → [0, 0, 1]
Label Encoding	Setiap kategori diberikan nilai numerik unik secara berurutan.	Hemat ruang memori karena tidak perlu membuat kolom tambahan.	Rentan terhadap bias jika algoritma menganggap urutan numerik memiliki arti khusus.	Merah = 0, Hijau = 1, Biru = 2
Ordinal Encoding	Digunakan jika kategori memiliki urutan tertentu, seperti tingkat pendidikan atau tingkat risiko.	- Menghormati urutan hierarkis antar-kategori, berguna untuk analisis berbasis urutan.	- Tidak cocok untuk kategori yang tidak memiliki urutan alami, bisa menyebabkan interpretasi keliru.	SMA = 1, S1 = 2, S2 = 3

4.3.3 Teknik Transformasi Non-Linear

Untuk dataset yang tidak terdistribusi normal atau mengandung outlier, teknik transformasi non-linear sering digunakan untuk menormalkan distribusi data dan mengurangi pengaruh outlier. Beberapa teknik umum meliputi:

1. Transformasi Logaritma
   Mengurangi skala data yang sangat besar dan menekan efek outlier. Misalnya, digunakan dalam data keuangan seperti pendapatan dan keuntungan.
   $$X' = \log(X + 1)$$
2. Transformasi Akar Kuadrat
   Digunakan untuk data yang bernilai kecil atau bervariasi rendah. Teknik ini sering digunakan dalam data yang mengandung variabel diskrit, seperti jumlah pelanggan.
   $$X' = \sqrt{X}$$
3. Transformasi Box-Cox
   Transformasi yang lebih fleksibel yang dapat mengubah berbagai bentuk distribusi ke bentuk mendekati normal dengan parameter lambda (λ).
   $$y(\lambda) = \begin{cases} \frac{X^{\lambda}-1}{\lambda}, & \text{jika } \lambda \neq 0 \\ \log(X), & \text{jika } \lambda = 0 \end{cases}$$

Teknik ini sering digunakan dalam regresi linier dan model statistik lainnya.

4.3.4 Reduksi Dimensi (PCA dan LDA)

Reduksi dimensi adalah proses mengurangi jumlah variabel dalam dataset dengan tetap mempertahankan informasi penting. Hal ini penting ketika dataset memiliki ratusan atau ribuan fitur, yang dapat menyebabkan overfitting dan memperlambat algoritma.

Principal Component Analysis (PCA)

PCA adalah teknik reduksi dimensi yang bekerja dengan mengubah fitur asli ke dalam sekumpulan fitur baru yang tidak saling berkorelasi, yang disebut principal components. PCA berfokus pada mengoptimalkan variabilitas data dan sering digunakan dalam masalah clustering dan visualisasi data (Jolliffe, 2011).

Langkah-langkah:

- Menghitung matriks kovarian dari data.
- Menghitung eigenvalue dan eigenvector dari matriks tersebut.
- Memilih sejumlah principal components dengan variansi tertinggi.

Linear Discriminant Analysis (LDA)

LDA digunakan untuk mengurangi dimensi sekaligus memisahkan kelas dalam dataset secara optimal. Teknik ini berfokus pada memaksimalkan perbedaan antara kelas-kelas dan meminimalkan variasi di dalam kelas. LDA cocok digunakan dalam masalah klasifikasi dengan data yang memiliki banyak fitur (McLachlan, 2004).

Dengan reduksi dimensi, peneliti dapat mengurangi kompleksitas data besar, mempercepat proses komputasi, dan mengurangi risiko overfitting dalam model pembelajaran mesin.

4.4 Penyimpanan dan Format Data

Dalam pengelolaan data, penyimpanan dan format data memainkan peran penting dalam menentukan seberapa efisien data dapat diakses, diolah, dan dilindungi. Dengan berbagai jenis data yang dihasilkan, mulai dari data terstruktur hingga tidak terstruktur, penting bagi peneliti dan praktisi data untuk memahami format yang tepat serta metode penyimpanan yang sesuai, baik secara lokal maupun melalui cloud.

4.4.1 Format Data Populer

Berbagai format penyimpanan data memiliki kelebihan masing-masing berdasarkan tujuan penggunaannya. Empat format populer yang umum digunakan adalah CSV, JSON, Parquet, dan SQL.

Tabel 4.4: Perbandingan Format Penyimpanan Data:

Format	Deskripsi	Kapan Digunakan
CSV (Comma-Separated Values)	Format teks sederhana di mana setiap baris mewakili satu rekaman, dan setiap nilai dipisahkan oleh koma. Mudah dibaca oleh manusia dan mesin.	Cocok untuk data kecil dan sederhana yang tidak memiliki hierarki kompleks. Tidak efisien untuk data besar karena tidak memiliki indeks bawaan.

Format	Deskripsi	Kapan Digunakan
JSON (JavaScript Object Notation)	Format data berbasis teks untuk menyimpan data terstruktur dalam bentuk pasangan atribut-nilai. Populer dalam aplikasi web dan API.	Ideal untuk data semi-terstruktur seperti hasil API. Efektif dalam menangani objek bersarang, tetapi tidak efisien untuk dataset besar.
Parquet	Format penyimpanan berbasis kolom yang dioptimalkan untuk analisis data besar. Digunakan dalam sistem big data seperti Apache Spark dan Hadoop.	Sangat efektif untuk analisis big data karena mampu menyimpan data terkompresi dan mendukung pencarian kolom tertentu tanpa membaca seluruh dataset.
SQL (Structured Query Language)	Format untuk mengelola dan mengambil data dari database relasional yang disusun dalam tabel. Mendukung kueri yang kompleks.	Cocok untuk data terstruktur dengan skema tetap yang sering diakses dan diolah menggunakan kueri kompleks. Banyak digunakan di aplikasi skala besar seperti MySQL, PostgreSQL, atau Oracle.

4.4.2 Penyimpanan Data Lokal dan Cloud

Penyimpanan data dapat dilakukan secara lokal atau melalui layanan cloud. Masing-masing memiliki kelebihan dan kekurangannya.

1. Penyimpanan Lokal (HDD/SSD):

   Data disimpan di perangkat keras seperti hard disk drive (HDD) atau solid-state drive (SSD) pada komputer atau server lokal. Penyimpanan lokal menawarkan kontrol penuh atas data, tanpa memerlukan koneksi internet. Namun, kelemahannya adalah keterbatasan kapasitas dan risiko kehilangan data jika terjadi kerusakan perangkat.

2. Penyimpanan Cloud:

   Penyimpanan cloud, seperti Google Drive, Amazon Web Services (AWS), atau Microsoft Azure, memungkinkan pengguna menyimpan data di server jarak jauh yang dapat diakses melalui internet. Keunggulan utama cloud adalah skalabilitas dan fleksibilitas, di mana pengguna dapat menambah kapasitas penyimpanan dengan mudah sesuai kebutuhan. Penyimpanan

cloud juga menawarkan keunggulan dalam kolaborasi dan backup otomatis, tetapi ketergantungan pada koneksi internet bisa menjadi kendala.

Tabel 4.5: Perbandingan Penyimpanan Data Lokal dan Cloud

Aspek	Penyimpanan Lokal	Penyimpanan Cloud
Kapasitas	Terbatas pada perangkat	Skalabel sesuai kebutuhan
Aksesibilitas	Terbatas pada perangkat fisik	Dapat diakses dari mana saja
Biaya	Biaya awal tinggi (pembelian perangkat)	Biaya berbasis penggunaan (pay-as-you-go)
Keamanan	Bergantung pada keamanan lokal	Bergantung pada penyedia layanan cloud

4.4.3 Manajemen Data Terstruktur dan Tidak Terstruktur

Data terstruktur adalah data yang dapat diorganisasikan dalam tabel dengan baris dan kolom yang jelas, seperti data keuangan atau catatan transaksi. Data ini biasanya disimpan dalam database relasional seperti MySQL atau PostgreSQL. Di sisi lain, data tidak terstruktur mencakup teks, gambar, video, audio, atau dokumen yang tidak memiliki format atau struktur yang jelas.

Tabel 4.6: Data Terstruktur dan Tidak Terstruktur

Aspek	Data Terstruktur	Data Tidak Terstruktur
Contoh	Tabel pelanggan, data penjualan	Dokumen teks, video, rekaman audio
Penyimpanan	Database relasional (SQL)	Sistem file, database NoSQL (MongoDB)
Kemudahan Analisis	Mudah dianalisis menggunakan kueri SQL	Memerlukan teknik khusus seperti NLP atau analisis gambar
Ukuran Data	Biasanya lebih kecil	Ukurannya besar dan terus bertambah

Manajemen data terstruktur memanfaatkan kueri SQL untuk manipulasi data, sementara data tidak terstruktur memerlukan teknologi khusus seperti mesin pencari berbasis dokumen atau pemrosesan citra digital.

4.4.4 Keamanan dan Backup Data

Keamanan data adalah aspek penting dalam pengelolaan data, terutama ketika data bersifat sensitif atau rahasia. Untuk menjaga integritas data, beberapa langkah strategis dapat diambil:

- Enkripsi: Mengubah data menjadi bentuk yang tidak dapat dibaca oleh pihak yang tidak memiliki kunci dekripsi. Enkripsi digunakan baik dalam penyimpanan (data at rest) maupun selama pengiriman (data in transit).
- Kontrol Akses: Mengatur hak akses pengguna untuk memastikan bahwa hanya pihak yang berwenang yang dapat mengakses data tertentu. Kontrol akses dapat diterapkan melalui sistem manajemen identitas dan akses (IAM).
- Sistem Backup: Backup data secara rutin ke lokasi berbeda penting untuk menghindari kehilangan data akibat kegagalan perangkat keras, serangan siber, atau kesalahan manusia. Backup dapat dilakukan secara lokal maupun melalui cloud. Strategi backup seperti full backup, incremental backup, dan differential backup dapat disesuaikan dengan kebutuhan.

Bab 5
Eksplorasi dan Visualisasi Data

5.1 Pengantar Eksplorasi dan Visualisasi Data

Eksplorasi dan visualisasi data merupakan tahap kritis dalam analisis data yang bertujuan untuk memahami pola, struktur, dan karakteristik dataset sebelum melakukan pemodelan atau pengambilan keputusan. Seiring dengan meningkatnya volume dan kompleksitas data dalam berbagai bidang seperti bisnis, kesehatan, sains, dan kecerdasan buatan, teknik eksplorasi data menjadi semakin penting untuk mengidentifikasi anomali, hubungan antar variabel, serta informasi tersembunyi dalam data.

Eksplorasi data (*Data Exploration*) adalah proses pemeriksaan awal terhadap dataset untuk memahami distribusi variabel, mengidentifikasi data yang hilang (*missing values*), menemukan pola umum, dan mendeteksi anomali. Teknik eksplorasi melibatkan metode statistik deskriptif, analisis korelasi, serta teknik reduksi dimensi untuk menyederhanakan kompleksitas dataset (W. M. Lim, 2024).

Sementara itu, visualisasi data (*Data Visualization*) adalah representasi grafis dari informasi dan data dengan tujuan untuk memudahkan interpretasi serta mengungkap wawasan yang sulit diperoleh hanya melalui angka atau tabel (Hudiburgh & Garbinsky, 2020). Visualisasi memungkinkan pengguna untuk memahami data secara lebih intuitif, melihat tren, membandingkan variabel, dan mendeteksi pola yang relevan dalam data.

Proses eksplorasi dan visualisasi data memiliki beberapa tujuan utama, di antaranya:

- Memahami struktur data: Mengetahui distribusi nilai, jenis variabel, serta kualitas dataset.
- Mengidentifikasi anomali dan outlier: Mendeteksi nilai ekstrem yang mungkin berdampak pada hasil analisis.
- Menemukan hubungan antar variabel: Menggunakan teknik statistik dan visualisasi untuk memahami keterkaitan antar fitur.

- Membantu dalam pemilihan fitur (feature selection): Menentukan variabel yang paling berkontribusi dalam analisis atau pemodelan.
- Menyajikan hasil secara intuitif: Menyediakan visualisasi yang memudahkan pemangku kepentingan dalam memahami temuan utama.

Manfaat eksplorasi dan visualisasi data sangat luas, mencakup berbagai disiplin ilmu, seperti:

- Dalam bisnis, membantu dalam analisis tren pasar, perilaku pelanggan, dan optimalisasi strategi pemasaran.
- Dalam kesehatan, digunakan untuk menganalisis data pasien, mendeteksi pola penyakit, dan meningkatkan pengambilan keputusan klinis.
- Dalam kecerdasan buatan dan pembelajaran mesin, membantu dalam pemilihan fitur dan evaluasi kinerja model.
- Dalam penelitian ilmiah, mendukung analisis eksperimen dan penyajian hasil yang lebih mudah dipahami.

5.2 Metode Eksplorasi Data

Metode eksplorasi data merupakan langkah awal dalam analisis data yang bertujuan untuk memahami karakteristik dataset sebelum dilakukan pemodelan atau pengambilan keputusan lebih lanjut. Eksplorasi data bertujuan untuk mengidentifikasi pola, hubungan antar variabel, mendeteksi anomali, serta menentukan fitur-fitur yang paling relevan dalam dataset (Bouwer et al., 2022).

Metode eksplorasi data umumnya dapat dibagi menjadi beberapa kategori utama, yaitu analisis statistik deskriptif, analisis korelasi, pembersihan data, serta teknik reduksi dimensi.

5.2.1 Analisis Statistik Deskriptif

Analisis statistik deskriptif merupakan langkah pertama dalam eksplorasi data yang bertujuan untuk menggambarkan distribusi data melalui ringkasan statistik. Teknik ini dapat dilakukan pada variabel kuantitatif maupun kualitatif dengan menggunakan berbagai metrik (Yellapu, 2018).

5.2.1.1 Statistik Sentral

Statistik Sentral adalah konsep dalam statistik yang digunakan untuk menggambarkan nilai pusat atau kecenderungan umum dalam suatu distribusi data. Dengan kata lain, statistik sentral membantu menentukan nilai yang mewakili dataset secara keseluruhan. Ada tiga ukuran utama dari statistik

sentral, yaitu Mean (Rata-rata), Median (Nilai Tengah), dan Modus (Nilai yang Paling Sering Muncul) (Downey, 2011).

5.2.1.2 Mean

Mean atau rata-rata adalah salah satu ukuran statistik deskriptif yang digunakan untuk menentukan nilai pusat dari sekumpulan data numerik. Mean dihitung dengan menjumlahkan semua nilai dalam dataset lalu membaginya dengan jumlah elemen dalam dataset tersebut.

Mean dari sekumpulan data X yang terdiri dari n elemen dinotasikan sebagai $\bar{X}$ dan dihitung menggunakan rumus:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

di mana:

- $\bar{X}$ = mean (rata-rata)
- Xi = nilai ke-i dalam dataset
- n = jumlah total data dalam dataset

Contoh Perhitungan Mean

Misalkan kita memiliki dataset berikut yang berisi nilai ujian lima siswa, $X = \{80,90,85,75,95\}$. Untuk menghitung nilai mean dari nilai ujian dapat dihitung sebagai:

$$\bar{X} = \frac{80 + 90 + 85 + 75 + 95}{5} = 85$$

Jadi, rata-rata nilai ujian dari lima siswa tersebut adalah 85.

5.2.1.3 Mean untuk Data Berbobot

Jika setiap data memiliki bobot yang berbeda (*Weighted Mean*), rumusnya menjadi:

$$\overline{X_w} = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i}$$

di mana: w_i = bobot dari setiap elemen X_i

Contoh:

Misalkan ada tiga produk dengan harga dan jumlah pembelian sebagai berikut:

Produk	Harga per Unit ($)	Jumlah Dibeli
A	10	5
B	20	3
C	15	2

Maka mean berbobot dari harga produk adalah:

$$\overline{X_w} = \frac{(10 \times 5) + (20 \times 3) + (15 \times 2)}{5 + 3 + 2} = 14$$

Jadi, rata-rata harga berbobot adalah $14 per unit.

5.2.1.4 Median

Median adalah ukuran pemusatan data yang menunjukkan nilai tengah dari sekumpulan data yang telah diurutkan. Median digunakan untuk menentukan titik tengah dari data, di mana separuh nilai berada di bawah median dan separuh lainnya berada di atasnya. Median sangat berguna, terutama saat data memiliki *outlier* atau distribusi yang tidak simetris karena tidak terpengaruh oleh nilai ekstrem. Median dapat ditentukan dengan menggunakan rumus:

$$\text{Median} = X_{\left(\frac{n+1}{2}\right)}$$, Jika jumlah data (n) ganjil

Artinya, median adalah nilai yang terletak di posisi ke-$\frac{n+1}{2}$ setelah data diurutkan.

$$Median = \frac{X_{\left(\frac{n}{2}\right)} + X_{\left(\frac{n}{2}+1\right)}}{2}$$, Jika jumlah data (n) genap

Artinya, median adalah rata-rata dari dua nilai tengah.

Contoh Median untuk Jumlah Data Ganjil

Misalkan dimiliki data berikut (belum terurut): $X = \{45,20,35,50,40\}$. Langkah awal adalah Urutkan data dari yang terkecil ke yang terbesar: $\{20,35,40,45,50\}$. Karena jumlah data $n = 5$ (ganjil), posisi median adalah: $\frac{5+1}{2} = 3$. Sehingga nilan median apa pada posisi ke-3, yaitu 40.

Contoh Median untuk Jumlah Data Genap

Misalkan dataset berikut:$X = \{60,55,70,65,75,80\}$, setelah data diurutkan menjadi $\{55,60,65,70,75,80\}$. Karena jumlah data $n = 6$ (genap), dua nilai tengah terletak di posisi ke-3 dan ke-4, yaitu: Nilai ke-3 = 65 dan Nilai ke-4 = 70. Sehingga median adalah

$$Median = \frac{65 + 70}{2} = \frac{135}{2} = 67.5$$

Median umumnya digunakan dalam beberapa kondisi khusus yang membuatnya lebih unggul dibandingkan dengan mean. Pertama, median lebih stabil saat digunakan pada data yang mengandung outlier atau nilai ekstrem. Hal ini disebabkan karena median tidak terpengaruh oleh nilai yang sangat besar atau sangat kecil, sehingga tetap memberikan gambaran yang akurat mengenai pusat data. Kedua, median sangat cocok untuk menganalisis data dengan distribusi yang tidak normal atau miring (*skewed distribution*), di mana data cenderung condong ke satu sisi. Dalam kasus seperti ini, median mampu merepresentasikan nilai tengah data secara lebih representatif dibandingkan mean. Terakhir, median juga efektif digunakan untuk data berskala ordinal, karena hanya memerlukan informasi tentang urutan atau peringkat data tanpa memperhitungkan jarak antar nilai. Dengan demikian, median menjadi pilihan yang tepat untuk menganalisis data dalam berbagai konteks yang memerlukan ketahanan terhadap distorsi nilai ekstrem dan distribusi yang tidak simetris.

5.2.1.5 Modus (*Mode*)

Modus adalah ukuran pemusatan data yang menunjukkan nilai atau kategori yang paling sering muncul dalam sebuah dataset. Berbeda dengan mean dan median yang berfokus pada posisi data dalam distribusi, modus mengukur frekuensi kemunculan suatu nilai. Modus dapat diterapkan pada data numerik maupun kategorikal. Modus sangat berguna untuk menganalisis data diskrit atau kategorikal, di mana informasi tentang nilai yang paling dominan dalam dataset diperlukan. Selain itu, modus juga dapat digunakan untuk data kontinu, meskipun interpretasinya lebih umum dalam bentuk distribusi frekuensi.

Tidak ada rumus matematis sederhana seperti pada mean atau median, karena modus bergantung pada frekuensi kemunculan data. Namun, untuk data yang berbentuk distribusi frekuensi (berkelompok), terdapat rumus untuk menghitung Modus Kelas Interval:

$$Modus = L + \left(\frac{(f_1 - f_0)}{(2f_1 - f_0 - f_2)} \right) \times w$$

Di mana:

- $L = batas\ bawah\ kelas\ modus$
- $f_1 = frekuensi\ kelas\ modus (frekuensi\ tertinggi)$
- $f_0 = frekuensi\ kelas\ sebelum\ kelas\ modus$
- $f_2 = frekuensi\ kelas\ setelah\ kelas\ modus$

- $w = lebar\ kelas\ interval$

Untuk data tak berkelompok (data mentah), modus cukup ditentukan dengan mengidentifikasi nilai yang muncul paling sering.

Contoh Perhitungan Modus

Modus untuk data tak berkelompok**.** Misalkan kita memiliki dataset: $X = \{4,2,5,3,2,6,2,7,3,5\}$. Hitung frekuensi kemunculan setiap angka, dan dapat dilihat bahwa angka 2 muncul 3 kali**.** Sehingga Modus dari dataset tersebut adalah **2**.

Modus untuk Data Berkelompok (Distribusi Frekuensi)

Tabel 5.1: Data Berkelompok

Kelas Interval	Frekuensi (f)
10–20	5
20–30	8
30–40	12
40–50	7
50–60	4

Langkah-langkah:

1. Identifikasi kelas modus: Kelas dengan frekuensi tertinggi adalah 30–40 dengan frekuensi $f_1 = 12$.
2. Batas bawah kelas modus (L): 30
3. Frekuensi kelas sebelum dan sesudahnya:
   - $f_0 = 8$ (kelas sebelum: 20–30)
   - $f_2 = 7$ (kelas setelah: 40–50)
4. Lebar kelas (w): 10 (selisih 40 - 30)

Gunakan rumus modus:

$$Modus = 30 + \left(\frac{12 - 8}{(2 \times 12) - 8 - 7}\right) \times 10$$
$$= 30 + (424 - 15) \times 10$$
$$= 30 + \left(\frac{4}{24 - 15}\right) \times 10$$
$$= 30 + (49) \times 10$$

$$= 30 + \left(\frac{4}{9}\right) \times 10$$
$$= 30 + 4.44$$
$$= 30 + 4.44$$
$$\approx 34.44$$

Modus digunakan dalam berbagai situasi, terutama ketika fokus analisis adalah pada frekuensi kemunculan suatu nilai. Salah satu penerapannya yang paling umum adalah untuk data kategorikal, di mana modus membantu mengidentifikasi kategori yang paling dominan, seperti preferensi warna favorit, jenis produk terpopuler, atau pilihan layanan yang paling sering digunakan. Selain itu, modus juga berguna dalam distribusi data yang skewed (tidak simetris), karena mampu memberikan gambaran mengenai puncak distribusi data tanpa terpengaruh oleh nilai ekstrem atau outlier. Dalam konteks ini, modus membantu memahami kecenderungan utama dalam data. Lebih lanjut, modus sangat efektif untuk menganalisis data dengan frekuensi dominan, seperti pola perilaku konsumen, di mana mengetahui nilai yang paling sering muncul dapat memberikan wawasan penting untuk pengambilan keputusan strategis. Dengan demikian, modus menjadi alat yang sederhana namun kuat untuk memahami karakteristik utama dari suatu dataset.

5.2.2 Statistik Dispersi (Variabilitas Data)

Statistik dispersi atau variabilitas data adalah ukuran yang digunakan untuk mengetahui seberapa besar sebaran atau penyebaran data dalam suatu dataset. Jika statistik pemusatan data seperti mean, median, dan **modus** digunakan untuk menggambarkan titik pusat data, maka statistik dispersi membantu memahami tingkat variasi data terhadap titik pusat tersebut. Ukuran variabilitas penting karena dua dataset bisa memiliki rata-rata yang sama, tetapi tingkat penyebaran data yang sangat berbeda. Dengan memahami variabilitas data, kita dapat mengukur konsistensi, volatilitas, dan risiko dalam analisis data, seperti dalam konteks keuangan, sains, atau penelitian sosial.

5.2.2.1 Rentang (*Range*)

Rentang adalah ukuran paling sederhana dari variabilitas data yang menunjukkan perbedaan antara nilai maksimum dan minimum dalam sebuah dataset.

$$Range = X_{\max} - X_{\min}$$

Kelebihan penggunaan rentang adalah mudah dihitung dan cepat memberikan gambaran kasar tentang sebaran data. Kekurangannya adalah sangat sensitif terhadap *outlier*, karena hanya mempertimbangkan dua nilai ekstrem.

Dipunyai dataset: $\{10,15,20,25,30\}$, $Range = 30 - 10 = 20$. Artinya, data tersebar dalam rentang 20 satuan.

5.2.2.2 Varians (*Variance*)

Varians mengukur seberapa jauh setiap nilai dalam dataset menyimpang dari rata-rata (mean). Varians dihitung dengan mengambil rata-rata dari kuadrat selisih antara setiap nilai data dengan rata-rata dataset.

Rumus Varians:

- Untuk populasi:

$$\sigma^2 = \frac{\sum_{i=1}^{N}(X_i - \mu)^2}{N}$$

- Untuk sampel:

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

Di mana:

- $X_i = nilaike - i\ dalam\ dataset$
- $\mu = mean\ populasi$
- $\bar{X} = mean\ sampel$
- $N = jumlah\ data\ untuk\ populasi$
- $n = jumlah\ data\ untuk\ sampel$

Contoh:

Dataset: 4,8,6,5,3

1. Mean $(\bar{X}) = 5.2(4 + 8 + 6 + 5 + 3)/5 = 5.2$
2. Hitung selisih kuadrat: $(4 - 5.2)2 = 1.44, (8 - 5.2)2 = 7.84, (6 - 5.2)2 = 0.64(4 - 5.2)^2 = 1.44, \quad (8 - 5.2)^2 = 7.84, \quad (6 - 5.2)^2 = 0.64(5 - 5.2)2 = 0.04, (3 - 5.2)2 = 4.84(5 - 5.2)^2 = 0.04, \quad (3 - 5.2)^2 = 4.84$
3. Jumlahkan: $1.44 + 7.84 + 0.64 + 0.04 + 4.84 = 14.81.44 + 7.84 + 0.64 + 0.04 + 4.84 = 14.8$

4. Varians sampel: $s^2 = \frac{14.8}{5-1} = \frac{14.8}{4} = 3.7$

Jadi, varians dari dataset adalah 3.7.

5.2.2.3 Simpangan Baku (*Standard Deviation*)

Simpangan baku atau standard deviation (SD) adalah akar kuadrat dari varians. SD digunakan untuk mengukur seberapa jauh data tersebar dari rata-rata. Karena menggunakan satuan yang sama dengan data aslinya (tidak dikuadratkan seperti varians), SD lebih mudah diinterpretasikan.

Rumus Simpangan Baku:

- Untuk populasi:

$$\sigma = \sqrt{\sigma^2}$$

- Untuk sampel:

$$s = \sqrt{s^2}$$

Contoh:

Dari perhitungan varians sebelumnya ($s^2 = 3.7$): $s = \sqrt{3.7} \approx 1.92$. Jadi, simpangan baku dari dataset tersebut adalah 1.92. Artinya, sebagian besar data berada sekitar 1.92 satuan dari nilai rata-rata (5.2).

5.2.2.4 Jangkauan Antar Kuartil (*Interquartile Range* - IQR)

IQR adalah ukuran variabilitas yang menggambarkan sebaran 50% data tengah, yaitu selisih antara kuartil ketiga ($Q3$) dan kuartil pertama ($Q1$).

$$\text{IQR} = Q3 - Q1$$

- Q1 (Kuartil 1): Nilai yang membagi 25% data terendah.
- Q3 (Kuartil 3): Nilai yang membagi 75% data terendah.

IQR sering digunakan untuk mendeteksi outlier karena data di luar batas:

$$[Q1 - 1.5 \times IQR,\ Q3 + 1.5 \times IQR]$$

biasanya dianggap sebagai outlier.

Contoh:

Dataset: $\{3,5,7,9,11,13,15\}$

- Q1 = 5, Q3 = 13

$$IQR = 13 - 5 = 8$$

Artinya, setengah dari data berada dalam rentang 8 satuan di tengah distribusi.

5.2.2.5 Koefisien Variasi (*Coefficient of Variation* - CV)

Koefisien Variasi adalah ukuran relatif variabilitas data yang menunjukkan perbandingan antara simpangan baku dengan mean, biasanya dinyatakan dalam persentase.

$$CV = \frac{\sigma}{\mu} \times 100\%$$

CV digunakan untuk membandingkan variabilitas antara dua dataset dengan skala atau unit yang berbeda.

Contoh:
Misalkan data A memiliki μ=50\mu = 50 dan σ=5\sigma = 5: $CV = \frac{5}{50} \times 100\% = 10\%$
Ini berarti tingkat variabilitas data A adalah 10% dari rata-ratanya.

5.2.3 Distribusi Data

Distribusi data adalah cara untuk menggambarkan bagaimana nilai-nilai dalam sebuah dataset tersebar atau terdistribusi. Distribusi ini menunjukkan frekuensi kemunculan dari setiap nilai atau kelompok nilai dalam data. Dengan memahami distribusi data, kita dapat memperoleh wawasan penting tentang karakteristik dataset, seperti pola, kecenderungan pusat (mean, median, modus), sebaran (varian, simpangan baku), dan potensi anomali (*outlier*).

Distribusi data sangat penting dalam statistik dan analisis data karena menentukan metode analisis yang tepat, pemilihan model statistik, serta interpretasi hasil yang lebih akurat.

5.2.3.1 Distribusi Normal (*Normal Distribution*)

Distribusi normal, juga dikenal sebagai distribusi Gaussian, adalah jenis distribusi data yang paling umum ditemukan di alam dan berbagai bidang penelitian. Distribusi ini membentuk kurva lonceng (*bell curve*) yang simetris terhadap mean.

Ciri-ciri Distribusi Normal:

- Simetris di sekitar rata-rata (mean = median = modus).
- Sebagian besar data berada di sekitar mean, dengan sedikit data di ekor distribusi.
- Mengikuti aturan 68-95-99.7 (Aturan Empiris):

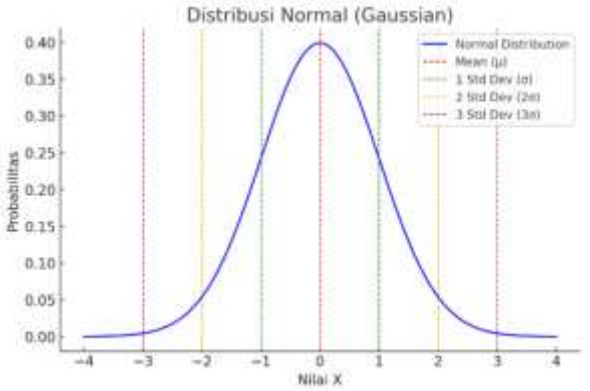
- 68% data berada dalam 1 simpangan baku dari mean.
- 95% data berada dalam 2 simpangan baku dari mean.
- 99.7% data berada dalam 3 simpangan baku dari mean.

Rumus Fungsi Distribusi Normal:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Di mana:

- $\mu = mean(rata - rata)$
- $\sigma = \ simpangan\ baku$
- $e = bilangan\ Euler(\sim 2.718)$
- $\pi = konstanta\ pi(\sim 3.1416)$



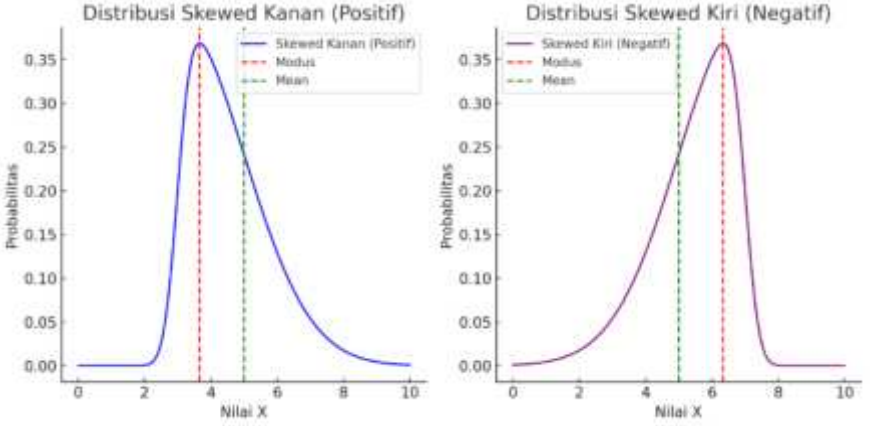
Gambar 5.1: Distribusi Normal

Gambar 5.1 menunjukkan **Distribusi Normal (Gaussian)** yang ditandai dengan kurva berbentuk **lonceng (*bell curve*)** yang simetris terhadap **mean** ($\mu = 0$). Kurva ini menggambarkan bagaimana data tersebar di sekitar nilai rata-rata, dengan sebagian besar data berada di tengah dan semakin berkurang menuju ekor distribusi. **Garis merah putus-putus** menunjukkan **mean** (μ), yang berfungsi sebagai pusat distribusi. Sementara itu, **garis hijau putus-putus** menunjukkan **1 simpangan baku** (σ) dari mean, yang mencakup sekitar **68%** dari total data. **Garis oranye putus-putus** menandai **2 simpangan baku** (2σ), yang mencakup sekitar **95%** dari data, menunjukkan bahwa sebagian besar data terdistribusi dalam rentang ini. Selanjutnya, **garis ungu putus-putus**

menunjukkan **3 simpangan baku** (3σ), yang mencakup sekitar **99.7%** dari seluruh data dalam distribusi. Dengan karakteristiknya yang universal, distribusi normal banyak digunakan dalam berbagai bidang penelitian seperti **statistik, kecerdasan buatan, ekonomi, dan ilmu sosial** untuk memahami pola data serta memperkirakan probabilitas kejadian dalam berbagai skenario.

5.2.3.2 Distribusi *Skewed* (Miring)

Distribusi *skewed* terjadi ketika data tidak simetris, melainkan condong ke satu sisi yaitu *Skewed* Kanan (Positif) dan *Skewed* Kiri (Negatif).



Gambar 5.2: Distribusi *Skewed*

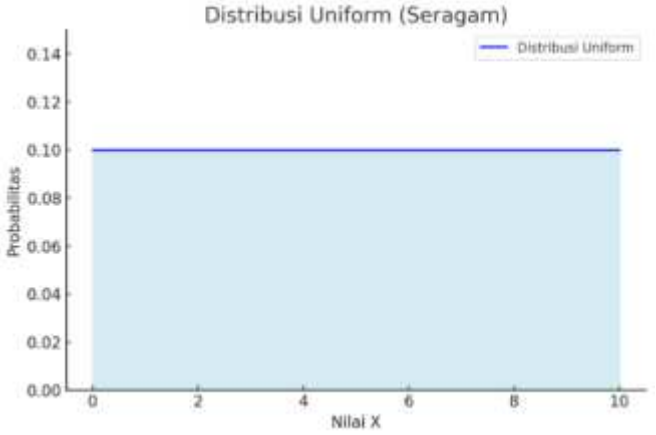
Gambar 5.2 menunjukkan Distribusi *Skewed* yang menggambarkan distribusi data yang tidak simetris dan condong ke satu sisi. Grafik sebelah kiri menampilkan Distribusi *Skewed* Kanan (Positif), di mana ekor distribusi lebih panjang di sisi kanan. Dalam distribusi ini, sebagian besar data terkonsentrasi di sisi kiri, sedangkan beberapa nilai ekstrem yang lebih besar menyebabkan ekor panjang ke arah kanan. Contoh umum dari distribusi ini adalah pendapatan individu, di mana sebagian besar orang memiliki penghasilan dalam kisaran tertentu, tetapi ada sedikit individu dengan penghasilan yang sangat tinggi. Modus (nilai yang paling sering muncul) berada di sebelah kiri (ditandai dengan garis merah putus-putus), sedangkan mean (rata-rata) cenderung tertarik ke arah ekor kanan (ditandai dengan garis hijau putus-putus).

Grafik sebelah kanan menunjukkan Distribusi *Skewed* Kiri (Negatif), di mana ekor distribusi lebih panjang di sisi kiri. Dalam distribusi ini, sebagian besar data terkonsentrasi di sisi kanan, sedangkan beberapa nilai ekstrem yang lebih kecil

menyebabkan ekor panjang ke arah kiri. Contoh distribusi ini dapat ditemukan pada usia pensiun, di mana sebagian besar individu pensiun pada usia yang relatif sama, tetapi ada beberapa yang pensiun lebih awal sehingga menyebabkan ekor distribusi yang panjang ke kiri. Seperti pada distribusi skewed kanan, modus berada di puncak distribusi, sedangkan mean cenderung tertarik ke arah ekor kiri.

5.2.3.3 Distribusi Uniform

Distribusi uniform menunjukkan bahwa semua nilai memiliki peluang yang sama untuk muncul. Bentuk distribusinya datar karena frekuensi kemunculan setiap nilai hampir sama.



Gambar 5.3: Distribusi Uniform

Gambar 5.3 menunjukkan Distribusi Uniform, yang ditandai dengan bentuk datar, di mana semua nilai memiliki peluang yang sama untuk muncul. Dalam distribusi ini, tidak ada nilai yang lebih sering muncul dibandingkan yang lain, sehingga grafiknya membentuk garis horizontal.

Distribusi ini sering digunakan dalam kasus di mana setiap kemungkinan hasil memiliki probabilitas yang sama. Contoh umum dari distribusi seragam adalah hasil lemparan dadu, di mana setiap angka dari 1 hingga 6 memiliki peluang yang sama untuk muncul, atau pengacakan angka dalam suatu rentang tertentu.

Karena setiap nilai dalam rentang tertentu memiliki peluang yang sama, distribusi ini tidak memiliki puncak yang jelas, berbeda dengan distribusi

normal atau distribusi skewed. Distribusi seragam sering digunakan dalam simulasi Monte Carlo, pengambilan sampel acak, dan berbagai aplikasi probabilitas.

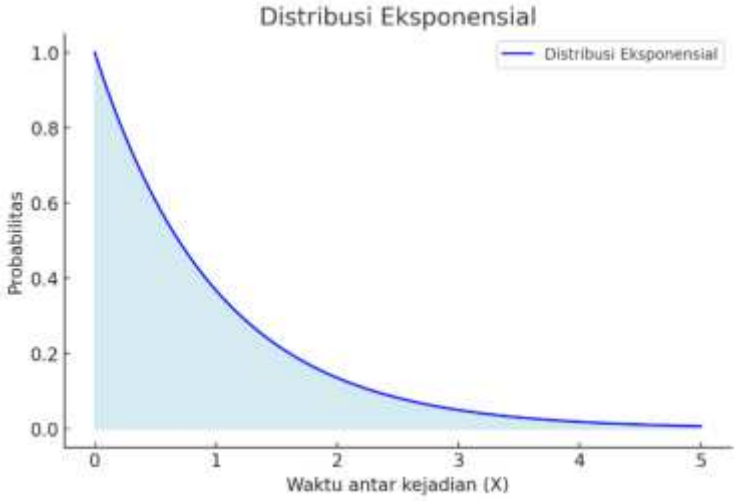
5.2.3.4 Distribusi Eksponensial (*Exponential Distribution*)

Distribusi eksponensial menggambarkan waktu antara kejadian-kejadian dalam proses Poisson (kejadian yang terjadi secara acak dalam interval waktu tertentu). Rumus Distribusi Eksponensial:

$$f(x;\lambda) = \lambda e^{-\lambda x}$$

Di mana:

- $\lambda = laju\ kejadian(rate\ parameter)$
- $x = waktu\ antar\ kejadian$



Gambar 5.4: Distribusi Eksponensial

Gambar 5.4 menunjukkan Distribusi Eksponensial, yang digunakan untuk menggambarkan waktu antara kejadian-kejadian dalam proses Poisson, yaitu kejadian yang terjadi secara acak dalam suatu interval waktu tertentu. Distribusi ini memiliki bentuk menurun secara eksponensial, dengan probabilitas kejadian yang tinggi pada awal waktu dan semakin kecil seiring bertambahnya waktu.

Hal ini mencerminkan bahwa sebagian besar kejadian terjadi dalam waktu yang relatif singkat, sementara hanya sedikit kejadian yang memiliki selang waktu yang lebih panjang.

Distribusi eksponensial memiliki parameter laju kejadian (λ), yang menentukan seberapa cepat kejadian terjadi. Jika nilai λ lebih besar, maka kejadian akan lebih sering terjadi dalam waktu yang lebih singkat.

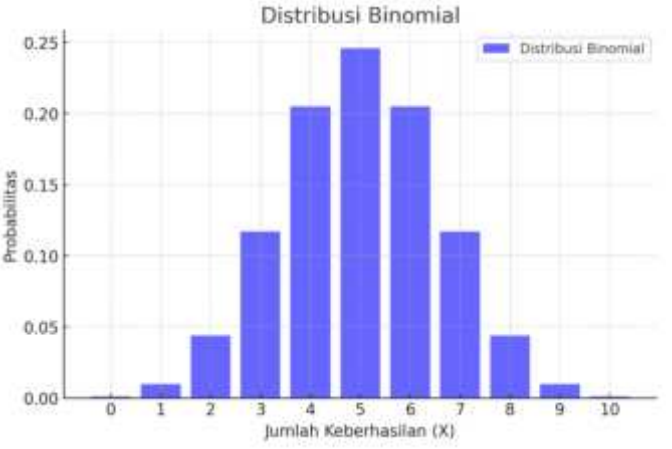
5.2.3.5 Distribusi Binomial (*Binomial Distribution*)

Distribusi binomial digunakan untuk menghitung probabilitas dari jumlah keberhasilan dalam sejumlah percobaan yang tetap, di mana setiap percobaan hanya memiliki dua hasil: berhasil (success) atau gagal (failure). Rumus Distribusi Binomial:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Di mana:

- $n = jumlah\ percobaan$
- $k = jumlah\ keberhasilan$
- $p = probabilitas\ keberhasilan$
- $\binom{n}{k} = kombinasi(cara\ memilih\ k\ dari\ n)$



Gambar 5.5: Distribusi Binomial

Gambar 5.5 menunjukkan Distribusi Binomial, yang digunakan untuk menghitung probabilitas dari jumlah keberhasilan dalam sejumlah percobaan

yang tetap. Setiap percobaan hanya memiliki dua kemungkinan hasil, yaitu berhasil (*success*) atau gagal (*failure*).

Dalam gambar 5, sumbu X mewakili jumlah keberhasilan (X) dalam 10 percobaan ($n = 10$), sedangkan sumbu Y menunjukkan probabilitas terjadinya jumlah keberhasilan tersebut. Distribusi binomial sering digunakan dalam berbagai situasi di mana ada percobaan independen dengan probabilitas keberhasilan tetap (p). Distribusi binomial memiliki karakteristik diskrit, karena hasilnya hanya berupa bilangan bulat yang menunjukkan jumlah keberhasilan dalam percobaan yang tetap.

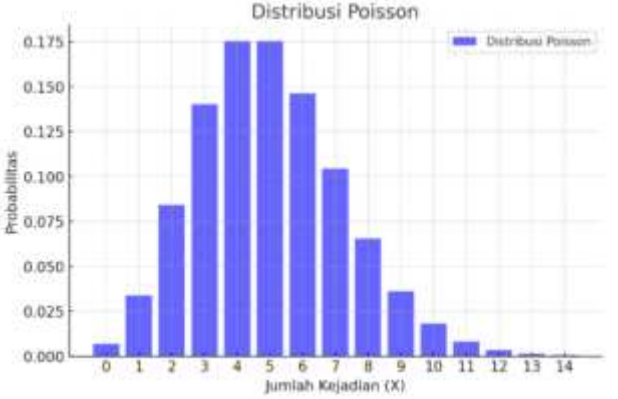
5.2.3.6 Distribusi Poisson

Distribusi Poisson digunakan untuk menghitung probabilitas terjadinya jumlah kejadian tertentu dalam suatu interval waktu atau ruang yang tetap. Rumus Distribusi Poisson:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Di mana:

- $\lambda = rata - rata\ jumlah\ kejadian\ dalam\ satu\ interval$
- $k = jumlah\ kejadian$
- $e = bilangan\ Euler (\sim 2.718)$



Gambar 5.6: Distribusi Poisson

Gambar 5.6 menunjukkan Distribusi Poisson, yang digunakan untuk menghitung probabilitas terjadinya jumlah kejadian tertentu dalam suatu

interval waktu atau ruang yang tetap. Distribusi ini sering digunakan untuk memodelkan kejadian yang terjadi secara acak namun dengan rata-rata tertentu dalam suatu periode waktu. Pada Gambar 6, sumbu X menunjukkan jumlah kejadian (X), sedangkan sumbu Y menunjukkan probabilitas terjadinya jumlah kejadian tersebut dalam interval tertentu. Distribusi ini dikendalikan oleh parameter rata-rata kejadian (λ), yang dalam contoh ini diatur ke 5, artinya rata-rata 5 kejadian terjadi dalam suatu interval waktu.

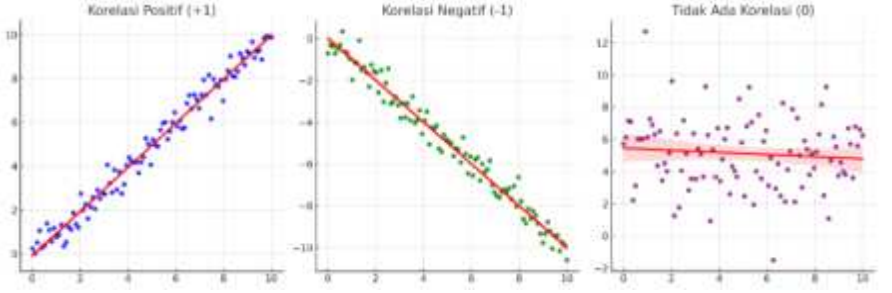
Distribusi Poisson bersifat diskrit, dan semakin besar nilai λ, bentuk distribusinya semakin mendekati distribusi normal. Distribusi ini banyak digunakan dalam statistik inferensial, keandalan sistem, dan analisis risiko.

5.2.4 Analisis Korelasi Antar Variabel

Analisis korelasi digunakan untuk mengidentifikasi hubungan antara dua atau lebih variabel dalam dataset. Korelasi dapat membantu dalam proses feature selection, karena variabel yang memiliki korelasi tinggi dapat menyebabkan multikolinearitas dalam pemodelan. Beberapa teknik yang digunakan antara lain:

5.2.4.1 *Pearson Correlation* (Korelasi Pearson)

Mengukur hubungan linier antara dua variabel numerik. Nilai korelasi berkisar antara -1 hingga 1.



Gambar 5.7: Pearson Correlation

Pada Gambar 5.7, grafik di sebelah kiri menunjukkan korelasi positif (+1), di mana ketika satu variabel meningkat, variabel lainnya juga ikut meningkat. Contohnya dapat ditemukan dalam hubungan antara suhu dan konsumsi es krim, di mana semakin panas cuaca, semakin tinggi pula penjualan es krim. Hal

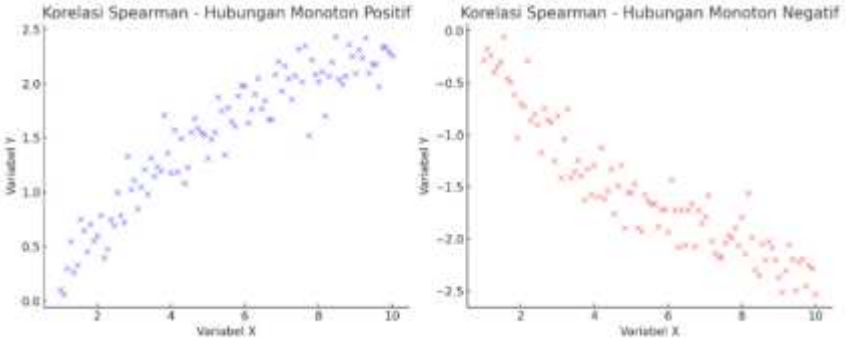
ini ditunjukkan oleh garis merah yang menggambarkan tren linier positif, yang berarti adanya hubungan yang kuat dan searah antara kedua variabel.

Sementara itu, grafik di tengah menunjukkan korelasi negatif (-1), di mana kenaikan pada satu variabel menyebabkan penurunan pada variabel lainnya. Contoh yang umum adalah hubungan antara kecepatan kendaraan dan waktu tempuh, di mana semakin cepat kendaraan melaju, semakin singkat waktu yang dibutuhkan untuk mencapai tujuan. Korelasi negatif ini divisualisasikan melalui garis merah yang menurun, menunjukkan tren linier negatif antara kedua variabel.

Di sisi kanan, grafik menunjukkan tidak adanya korelasi (0), yang berarti tidak terdapat pola linier yang jelas antara kedua variabel. Contoh yang mencerminkan kondisi ini adalah hubungan antara nomor telepon seseorang dan berat badannya, di mana tidak ada keterkaitan logis antara keduanya. Pada grafik ini, titik-titik data tersebar secara acak tanpa membentuk pola tertentu, yang menunjukkan bahwa tidak ada hubungan linier antara variabel-variabel tersebut.

5.2.4.2 Spearman Rank Correlation

Digunakan jika hubungan antara dua variabel tidak linier tetapi memiliki hubungan monoton (misalnya, pertumbuhan populasi vs konsumsi energi).



Gambar 5.8: Korelasi Spearman Rank

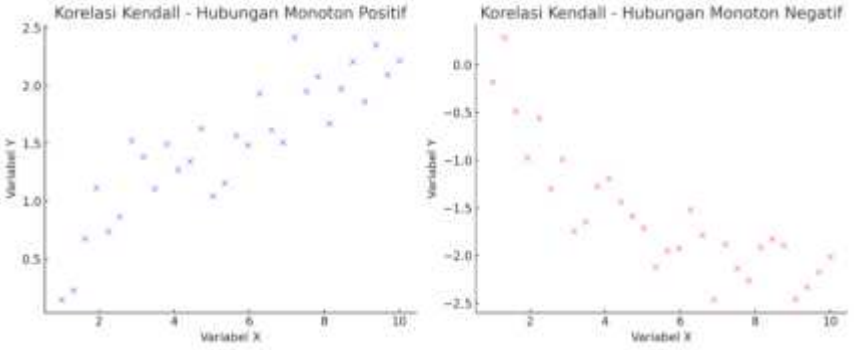
Gambar 5.8 menunjukkan Korelasi Spearman Rank, yang digunakan ketika hubungan antara dua variabel tidak linier tetapi tetap memiliki hubungan monoton. Korelasi ini mengukur sejauh mana hubungan antara dua variabel tetap meningkat atau menurun, meskipun bentuknya tidak mengikuti pola linier seperti pada Korelasi Pearson.

- Grafik kiri: Korelasi Spearman - Hubungan Monoton Positif
  Pada grafik ini, nilai Y meningkat seiring dengan meningkatnya X, tetapi hubungannya tidak berbentuk garis lurus. Contohnya adalah hubungan antara pengalaman kerja dan gaji, di mana kenaikan gaji tidak selalu proporsional terhadap jumlah tahun pengalaman, tetapi tetap menunjukkan tren kenaikan secara keseluruhan.
- Grafik kanan: Korelasi Spearman - Hubungan Monoton Negatif
  Di grafik ini, nilai Y menurun ketika X meningkat, tetapi tidak dalam pola linier sempurna. Contoh kasusnya adalah hubungan antara usia mobil dan harga jualnya, di mana semakin tua usia mobil, harga jualnya cenderung menurun, meskipun laju penurunannya tidak selalu seragam.

Korelasi Spearman sering digunakan dalam analisis data ordinal, penelitian sosial, dan hubungan tidak linier dalam sains karena dapat menangkap pola monotonicity tanpa memerlukan asumsi linieritas seperti pada Korelasi Pearson.

5.2.5 Kendall's Tau

Kendall's Tau hampir sama dengan Spearman tetapi lebih robust terhadap dataset kecil dan memiliki lebih sedikit asumsi terhadap distribusi data.



Gambar 5.9: Korelasi Kendall's

Gambar 5.9 menunjukkan Korelasi Kendall's Tau, yang digunakan untuk mengukur hubungan antara dua variabel yang memiliki hubungan monoton dengan lebih sedikit asumsi terhadap distribusi data. Kendall's Tau mirip dengan Spearman Rank, tetapi lebih robust untuk dataset kecil dan lebih tahan terhadap data dengan distribusi yang tidak teratur.

- Grafik kiri: Korelasi Kendall - Hubungan Monoton Positif

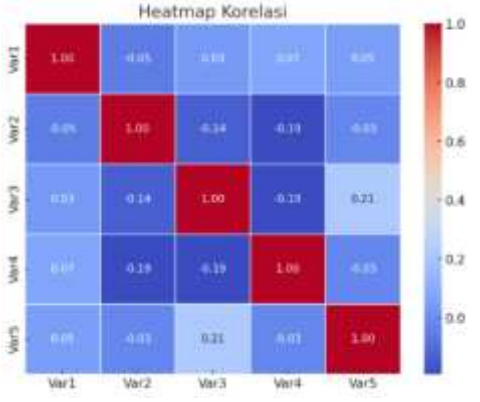
  Grafik ini menunjukkan bahwa ketika X meningkat, Y juga meningkat, meskipun tidak dalam garis lurus yang sempurna. Contoh aplikasinya adalah hubungan antara jumlah latihan dan peningkatan keterampilan, di mana lebih banyak latihan cenderung meningkatkan keterampilan seseorang, meskipun dengan variasi yang kecil.
- Grafik kanan: Korelasi Kendall - Hubungan Monoton Negatif

  Grafik ini menunjukkan bahwa ketika X meningkat, Y menurun, tetapi tidak dalam hubungan linier sempurna. Contohnya adalah hubungan antara usia produk elektronik dan nilai jualnya, di mana semakin tua produk, semakin rendah harganya, tetapi ada variasi harga tergantung pada kondisi dan permintaan pasar.

Korelasi Kendall sering digunakan dalam analisis data ordinal, peringkat dalam penelitian sosial, dan hubungan non-parametrik karena tidak terlalu bergantung pada asumsi distribusi normal. Ini membuatnya lebih andal dalam situasi dengan data kecil atau tidak beraturan dibandingkan dengan Korelasi Pearson atau Spearman.

5.2.5.1 Heatmap Korelasi

Korelasi antara variabel dapat divisualisasikan menggunakan heatmap, yang mempermudah dalam mengidentifikasi variabel yang berkorelasi tinggi dalam dataset.



Gambar 5.10: Heatmap Korelasi

Gambar 5.10 menunjukkan Heatmap Korelasi, yang digunakan untuk memvisualisasikan hubungan antara berbagai variabel dalam suatu dataset. Warna dalam heatmap menunjukkan tingkat korelasi antara variabel-variabel tersebut, dengan nilai berkisar dari -1 hingga 1:

- Nilai +1 (merah tua): Korelasi positif sempurna, artinya ketika satu variabel meningkat, variabel lainnya juga meningkat.
- Nilai -1 (biru tua): Korelasi negatif sempurna, artinya ketika satu variabel meningkat, variabel lainnya menurun.
- Nilai 0 (putih atau abu-abu muda): Tidak ada hubungan linier yang signifikan antara kedua variabel.

Heatmap ini membantu mengidentifikasi variabel yang memiliki korelasi tinggi dalam dataset, sehingga dapat digunakan untuk seleksi fitur dalam machine learning, deteksi multikolinearitas dalam regresi, atau memahami hubungan antara variabel dalam analisis data.

5.3 Pembersihan Data (*Data Cleaning*)

Pembersihan data merupakan tahap penting untuk memastikan kualitas data sebelum dilakukan analisis lebih lanjut. Proses ini melibatkan deteksi dan penanganan *missing values*, duplikasi, *outlier*, serta transformasi tipe data (Cielen et al., 2016).

5.3.1 Menangani Data Hilang (*Missing Values*)

Menangani data hilang (*Missing Values*) merupakan langkah penting dalam proses analisis data untuk memastikan kualitas dan akurasi hasil analisis. Data yang hilang dapat terjadi karena berbagai alasan, seperti kesalahan input, keterbatasan sensor, atau masalah dalam proses pengumpulan data. Jika tidak ditangani dengan baik, data yang hilang dapat menyebabkan bias dalam analisis dan mengurangi efektivitas model prediktif (Shumeiko & Rozora, 2021).

Ada beberapa metode untuk menangani data hilang tergantung pada jumlah dan pola hilangnya data. Salah satu pendekatan sederhana adalah menghapus baris atau kolom yang mengandung nilai hilang, terutama jika jumlahnya sangat sedikit dan tidak signifikan terhadap keseluruhan dataset. Namun, jika data yang hilang cukup banyak, metode ini bisa mengurangi informasi yang berharga. Alternatif lainnya adalah imputasi data, yaitu menggantikan nilai yang hilang dengan nilai statistik seperti mean, median, atau modus untuk variabel numerik. Pendekatan ini cocok untuk dataset dengan distribusi yang tidak terlalu *skewed*.

Selain metode sederhana, teknik yang lebih canggih seperti imputasi berbasis model dapat digunakan untuk memperkirakan nilai yang hilang. Contohnya adalah *K-Nearest Neighbors* (KNN) *Imputation*, yang menggantikan nilai yang hilang dengan rata-rata nilai dari titik data terdekat berdasarkan kemiripan fitur. Metode lainnya adalah regresi atau machine learning, di mana model dibangun untuk memprediksi nilai yang hilang berdasarkan hubungan antar variabel dalam dataset.

Dalam situasi tertentu, jika nilai yang hilang memiliki pola yang tidak acak, pendekatan seperti *Multiple Imputation* dapat digunakan untuk menghasilkan beberapa kemungkinan imputasi, sehingga meningkatkan keakuratan estimasi. Selain itu, metode seperti Interpolasi Linier dapat diterapkan pada data yang memiliki pola waktu (*time-series*) untuk memperkirakan nilai berdasarkan tren historis.

Pemilihan metode yang tepat sangat bergantung pada karakteristik dataset dan tujuan analisis. Oleh karena itu, penting untuk mengevaluasi distribusi data yang hilang dan memilih pendekatan yang paling sesuai untuk mengurangi dampak negatif terhadap hasil analisis dan prediksi.

5.3.2 Menangani Data Duplikat

Menangani Data Duplikat adalah langkah penting dalam proses pembersihan data untuk memastikan keakuratan dan efisiensi analisis. Data duplikat terjadi ketika satu atau lebih entri dalam dataset muncul lebih dari sekali, yang dapat disebabkan oleh kesalahan dalam proses pengumpulan data, penggabungan dataset dari berbagai sumber, atau kesalahan sistem dalam perekaman data. Keberadaan data duplikat dapat menyebabkan bias dalam analisis dan mempengaruhi performa model prediktif dalam machine learning (Elmagarmid et al., 2007).

Proses deteksi data duplikat dapat dilakukan dengan menggunakan metode sederhana seperti **identifikasi berdasarkan nilai yang sama dalam semua kolom** atau dengan **pencocokan sebagian (*fuzzy matching*)** jika terdapat kemungkinan variasi kecil dalam data. Dalam bahasa pemrograman seperti Python, fungsi `duplicated()` pada pustaka `pandas` dapat digunakan untuk menemukan dan menghapus entri yang sama. Jika dataset berasal dari SQL, perintah `DISTINCT` dapat diterapkan untuk menghapus duplikasi.

Setelah deteksi, langkah berikutnya adalah **menghapus atau mengelola duplikasi** berdasarkan kebutuhan analisis. Jika data duplikat tidak

diperlukan, entri tersebut dapat dihapus menggunakan drop_duplicates(). Namun, dalam beberapa kasus, seperti data transaksi atau sistem keanggotaan, beberapa entri yang terlihat serupa mungkin tetap valid, sehingga perlu dianalisis lebih lanjut dengan mempertimbangkan kolom tambahan seperti ***timestamp*** **atau ID unik** untuk membedakan entri yang sebenarnya berbeda.

Selain itu, jika duplikasi terjadi karena kesalahan penginputan, teknik **penyamaan format (*standardization*)** dapat digunakan untuk memastikan nilai yang seharusnya sama memiliki format yang seragam, seperti menyamakan huruf kapital, menghapus spasi ekstra, atau mengonversi format tanggal ke standar tertentu. Dengan menerapkan strategi yang tepat dalam menangani data duplikat, dataset dapat menjadi lebih bersih dan akurat, sehingga menghasilkan analisis yang lebih terpercaya dan efisien.

5.3.3 Deteksi dan Penanganan *Outlier*

Deteksi dan Penanganan *Outlier* adalah langkah krusial dalam analisis data untuk memastikan bahwa data yang digunakan dalam pemodelan atau pengambilan keputusan tidak dipengaruhi oleh nilai ekstrem yang tidak wajar. *Outlier* adalah titik data yang secara signifikan berbeda dari sebagian besar data lainnya dalam dataset. *Outlier* dapat muncul karena kesalahan pengukuran, variasi alami dalam data, atau faktor luar yang memengaruhi proses pengumpulan data. Jika tidak ditangani dengan baik, outlier dapat menyebabkan hasil analisis yang bias dan mengurangi keakuratan model prediktif.

Proses deteksi *outlier* dapat dilakukan menggunakan berbagai metode statistik dan visualisasi. Salah satu cara yang paling umum adalah menggunakan *Interquartile Range* (IQR), di mana outlier didefinisikan sebagai nilai yang berada di luar rentang [Q1 - 1.5 × IQR, Q3 + 1.5 × IQR]. Alternatif lainnya adalah metode *Z-score*, di mana titik data dengan nilai lebih dari 3 standar deviasi dari mean dianggap sebagai outlier. Selain itu, teknik visualisasi seperti *boxplot* dan *scatter plot* dapat digunakan untuk mengidentifikasi pola dan anomali dalam distribusi data.

Setelah mendeteksi *outlier*, langkah berikutnya adalah memutuskan bagaimana menanganinya. Jika *outlier* merupakan hasil dari kesalahan pengukuran atau data yang tidak valid, maka dapat dihapus dari dataset. Namun, jika *outlier* adalah bagian alami dari fenomena yang diamati, seperti dalam kasus analisis keuangan atau ilmu eksperimental, maka perlu dilakukan pendekatan yang lebih hati-hati. Salah satu pendekatan adalah transformasi data, seperti *log transformation* atau *Winsorization*, yang membatasi nilai ekstrem agar tidak

terlalu memengaruhi analisis. Alternatif lain adalah menggunakan model yang lebih tahan terhadap *outlier*, seperti *Median Absolute Deviation* (MAD) atau algoritma berbasis pohon keputusan dalam machine learning yang lebih robust terhadap nilai ekstrem.

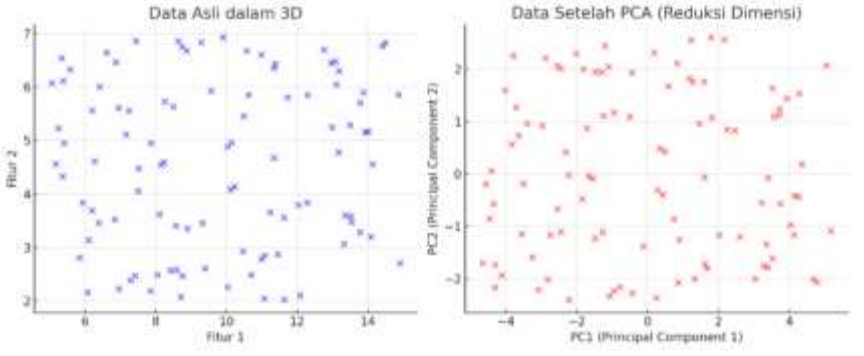
Pemilihan metode penanganan *outlier* harus disesuaikan dengan konteks data dan tujuan analisis. Jika data yang dianalisis memerlukan akurasi tinggi, seperti dalam bidang kesehatan atau keuangan, maka evaluasi yang lebih mendalam terhadap outlier sangat diperlukan sebelum mengambil keputusan. Dengan menangani outlier secara tepat, kualitas data dapat ditingkatkan, sehingga menghasilkan hasil analisis yang lebih akurat dan dapat diandalkan.

5.3.4 Reduksi Dimensi

Dalam dataset yang memiliki banyak fitur (dimensi tinggi), reduksi dimensi dapat membantu dalam meningkatkan efisiensi analisis dengan menghilangkan variabel yang kurang informatif. Beberapa metode yang umum digunakan adalah:

5.3.4.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah teknik reduksi dimensi yang digunakan untuk menyederhanakan dataset dengan banyak variabel menjadi beberapa komponen utama (*Principal Components* - PCs), tanpa kehilangan terlalu banyak informasi (Jolliffe, 2011). PCA bekerja dengan mengubah sumbu koordinat sehingga dataset dapat direpresentasikan dalam dimensi yang lebih rendah, sambil tetap mempertahankan sebanyak mungkin variasi dari data asli.



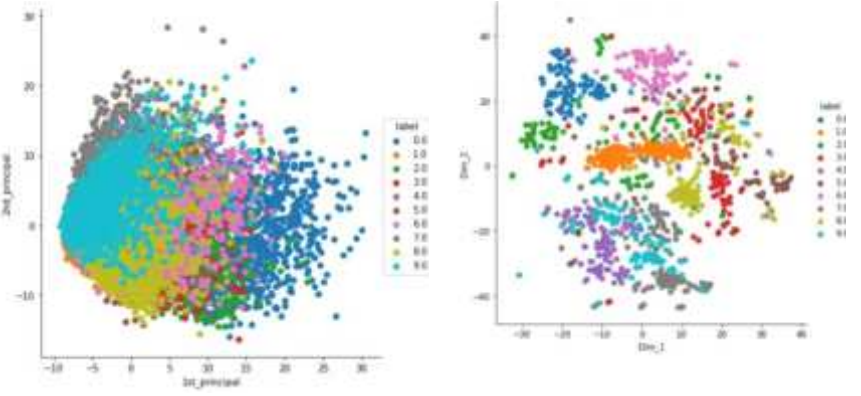
Gambar 5.11: Principal Component Analysis

Gambar 11 menunjukkan dua representasi data sebelum dan sesudah PCA diterapkan. Grafik kiri menampilkan data asli dalam ruang berdimensi tiga, di mana fitur-fitur masih saling berkorelasi. Grafik kanan menunjukkan hasil setelah PCA diterapkan, di mana data telah direduksi menjadi dua komponen utama (PC1 dan PC2). PCA mengekstrak arah dengan variasi terbesar dalam data dan menggunakannya untuk membentuk dimensi baru yang lebih optimal untuk analisis.

PCA sangat berguna dalam *machine learning* dan analisis data, terutama ketika bekerja dengan dataset berdimensi tinggi, karena dapat membantu mengurangi kompleksitas model, mengatasi multikolinearitas, serta meningkatkan efisiensi pemrosesan data. Namun, perlu diperhatikan bahwa PCA hanya menangkap variasi linier dalam data dan dapat kehilangan informasi penting jika tidak digunakan dengan bijak.

5.3.4.2 t-Distributed Stochastic Neighbor Embedding (t-SNE)

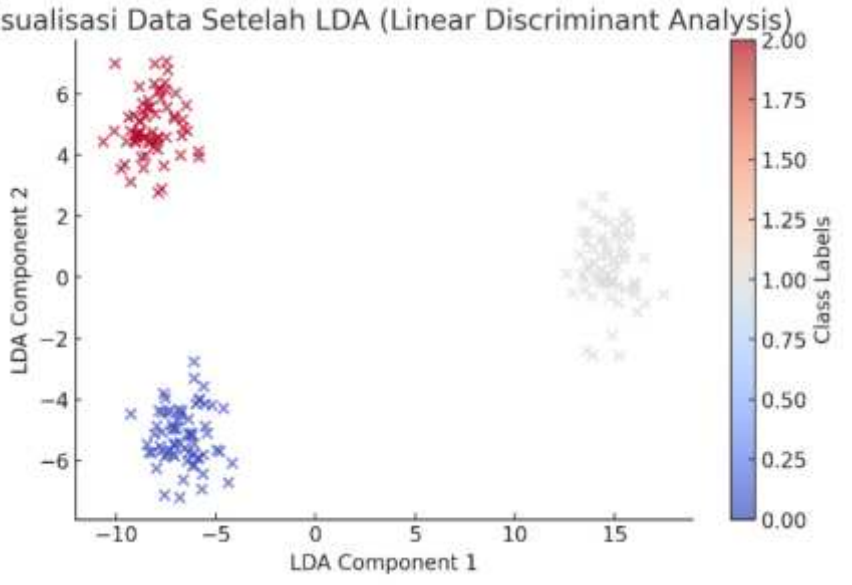
t-*Distributed Stochastic Neighbor Embedding* (t-SNE) adalah teknik reduksi dimensi non-linier yang digunakan untuk memvisualisasikan data berdimensi tinggi dalam bentuk dua atau tiga dimensi. Berbeda dengan PCA, yang mengubah variabel ke dalam komponen linier baru, t-SNE memfokuskan pada pelestarian hubungan lokal antar data, sehingga lebih efektif dalam menampilkan pola klasterisasi dan struktur kompleks dalam dataset.



Gambar 5.12: Reduksi data dengan PCA dan t-SNE

Gambar 5.12 memperlihatkan perbedaan kemampuan reduksi data dengan menggunakan PCA (gambar kiri) dan t-SNE (gambar kanan). t-SNE bekerja dengan menghitung probabilitas kesamaan antara titik data dalam ruang berdimensi tinggi, kemudian memetakan titik-titik tersebut ke dalam ruang berdimensi rendah dengan menjaga hubungan kedekatannya. Teknik ini sering digunakan dalam *machine learning*, *computer vision*, dan bioinformatika untuk mengeksplorasi pola dalam data yang sangat kompleks.

5.3.4.3 Linear Discriminant Analysis (LDA)



Gambar 5.13: Linear Discriminant Analysis

Gambar 5.13 menunjukkan *Linear Discriminant Analysis* (LDA), yang merupakan teknik reduksi dimensi yang sering digunakan dalam klasifikasi. Berbeda dengan PCA yang memaksimalkan varians dalam data, LDA berfokus pada pemisahan antar kelas, sehingga sangat berguna dalam machine learning dan pengenalan pola. LDA bekerja dengan mencari kombinasi linier dari fitur-fitur yang memaksimalkan perbedaan antar kelas dalam data (Tharwat et al., 2017). Proses ini melibatkan:

- Menghitung mean dari setiap kelas untuk memahami bagaimana data tersebar.
- Menghitung scatter matrices untuk mengukur penyebaran data dalam dan antar kelas.
- Mencari vektor eigen yang memaksimalkan rasio antara varians antar kelas dan varians dalam kelas.
- Memproyeksikan data ke dalam dimensi yang lebih rendah, di mana kelas dapat dipisahkan lebih baik.

Dalam grafik di atas, data awal memiliki 5 fitur, tetapi telah direduksi menjadi 2 komponen LDA yang tetap mempertahankan pemisahan antar kelas dengan optimal. Titik-titik dengan warna berbeda menunjukkan kelas yang berbeda, di mana LDA membantu memisahkan kelompok data dengan lebih jelas.

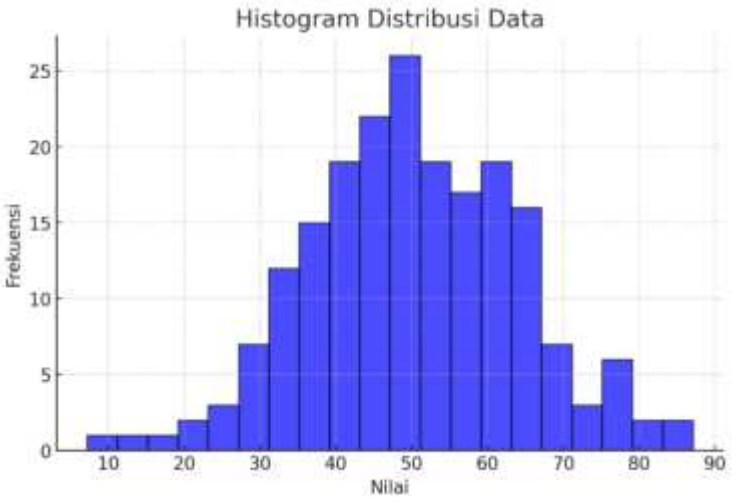
LDA banyak digunakan dalam klasifikasi gambar, analisis sentimen, dan pengenalan wajah, karena membantu mengurangi dimensi data sekaligus mempertahankan informasi yang relevan untuk pemisahan kelas. Dengan menggunakan LDA, model klasifikasi dapat bekerja lebih efisien dan akurat, terutama dalam dataset dengan fitur yang saling berkorelasi.

5.4 Teknik Visualisasi Data

Teknik visualisasi data adalah metode untuk menyajikan informasi dalam bentuk grafis agar lebih mudah dipahami dan dianalisis. Visualisasi ini membantu mengidentifikasi pola, tren, dan anomali dalam dataset yang mungkin sulit dikenali hanya dengan melihat angka-angka mentah. Berbagai teknik visualisasi tersedia, masing-masing digunakan sesuai dengan jenis data dan tujuan analisis.

5.4.1 Histogram

Histogram digunakan untuk melihat distribusi suatu variabel numerik. Histogram menunjukkan seberapa sering nilai tertentu muncul dalam suatu dataset.



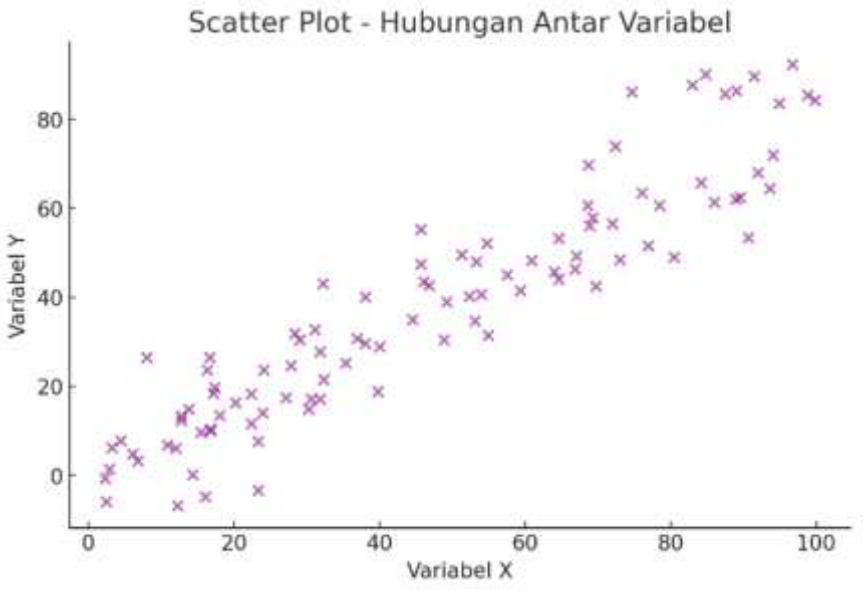
Gambar 5.14: Histogram

Gambar 5.14 menunjukkan Histogram, yang digunakan untuk menampilkan distribusi suatu variabel numerik. Histogram membagi data ke dalam beberapa interval (bins) dan menghitung frekuensi kemunculan data dalam setiap interval tersebut. Pada histogram ini Sumbu X mewakili nilai variabel yang diamati. Sumbu Y menunjukkan frekuensi atau jumlah kemunculan nilai dalam setiap interval.

Bentuk histogram dapat memberikan wawasan tentang distribusi data, seperti apakah data mengikuti distribusi normal, skewed (miring ke kanan/kiri), atau memiliki multimodal peaks (lebih dari satu puncak).

5.4.2 Scatter Plot

Scatter Plot membantu memahami hubungan antara dua variabel numerik. Titik-titik dalam scatter plot menunjukkan apakah ada pola hubungan antara kedua variabel, seperti korelasi positif atau negatif.



Gambar 5.15: Scatter Plot

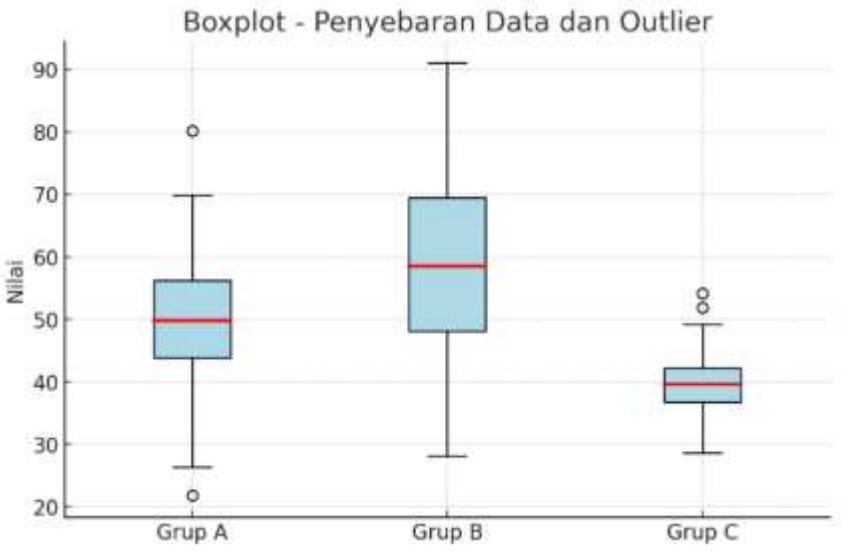
Gambar 5.15 menunjukkan Scatter Plot, yang digunakan untuk menampilkan hubungan antara dua variabel numerik. Dalam scatter plot ini: Sumbu X mewakili variabel pertama (*X*). Sumbu Y mewakili variabel kedua (*Y*). Setiap titik pada plot mewakili satu pasangan nilai dari kedua variabel.

Scatter plot membantu dalam mengidentifikasi pola hubungan antara variabel, seperti: Korelasi positif (ketika *X* meningkat, *Y* juga meningkat). Korelasi negatif (ketika *X* meningkat, *Y* menurun). Tidak ada korelasi (jika titik tersebar secara acak tanpa pola yang jelas).

Scatter plot sering digunakan dalam analisis data, statistik, dan machine learning untuk memahami hubungan antara fitur dalam dataset sebelum melakukan pemodelan lebih lanjut.

5.4.3 Boxplot

Boxplot digunakan untuk memahami penyebaran data dan mendeteksi outlier. Boxplot menunjukkan median, kuartil, dan nilai ekstrem dalam suatu dataset.



Gambar 5.16: Boxplot

Gambar 5:16 menunjukkan Boxplot, yang digunakan untuk menampilkan distribusi dan penyebaran data, serta mendeteksi outlier. Boxplot sangat berguna dalam analisis statistik karena dapat memberikan gambaran visual tentang bagaimana data tersebar tanpa harus melihat seluruh distribusi angka.

Pada diagram ini:

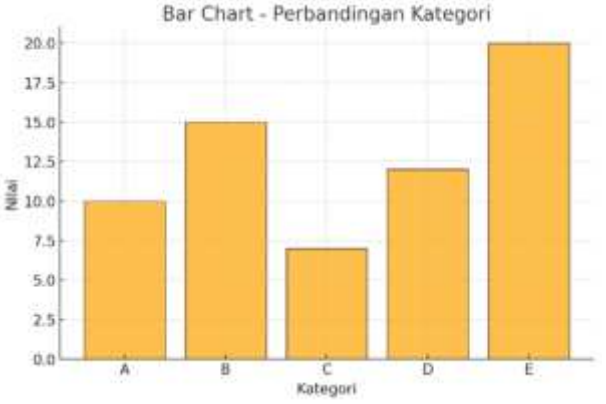
- Kotak biru mewakili rentang interkuartil (Interquartile Range - IQR), yaitu 50% data tengah (antara kuartil pertama $Q1$ dan kuartil ketiga $Q3$).
- Garis merah di dalam kotak menunjukkan median ($Q2$) atau nilai tengah data.
- Garis horizontal (whiskers) di atas dan bawah kotak menunjukkan rentang data tanpa outlier.
- Titik di luar whiskers adalah outlier, yaitu nilai yang berada di luar batas $Q1 - 1.5 \times IQR$ atau $Q3 + 1.5 \times IQR$.

Boxplot ini menampilkan tiga grup data: Grup A memiliki median sekitar 50 dengan variasi data yang cukup kecil. Grup B memiliki median lebih tinggi di sekitar 60, tetapi penyebaran datanya lebih luas. Grup C memiliki median di sekitar 40 dengan penyebaran yang lebih sempit, menunjukkan data yang lebih konsisten.

Boxplot sering digunakan dalam eksplorasi data dan statistik deskriptif untuk membandingkan distribusi beberapa kelompok data, mengidentifikasi pencilan (*outlier*), serta memahami variabilitas dalam dataset.

5.4.4 Bar Chart (Diagram Batang)

Bar Chart menampilkan perbandingan antara kategori dalam bentuk batang, di mana panjang batang menunjukkan nilai masing-masing kategori.



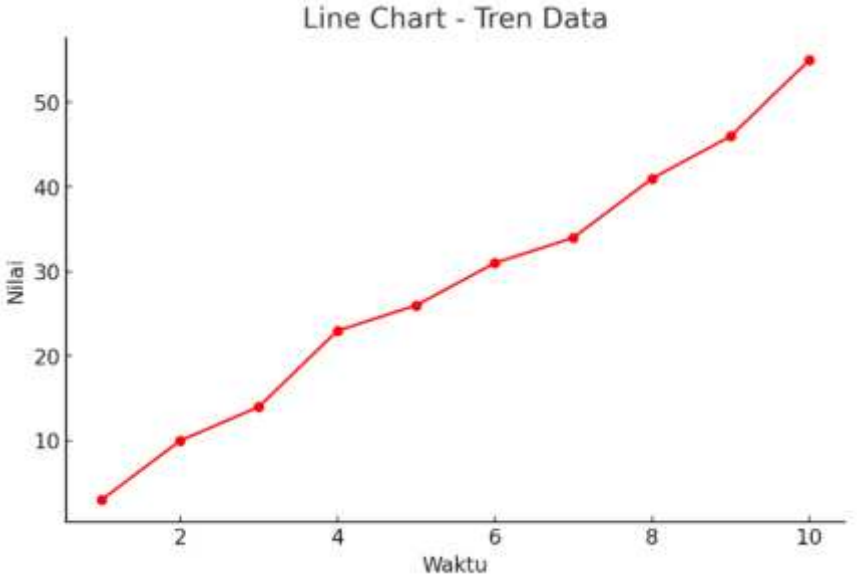
Gambar 5.17: Bar Chart

Gambar 5.17 menunjukkan Bar Chart, yang digunakan untuk membandingkan nilai antar kategori. Diagram batang sangat berguna dalam analisis data kategorikal, di mana kita ingin melihat bagaimana setiap kategori memiliki distribusi nilai yang berbeda.

Pada diagram ini: Sumbu X mewakili kategori yang diamati (A, B, C, D, dan E). Sumbu Y menunjukkan nilai atau frekuensi dari setiap kategori. Panjang batang menunjukkan jumlah atau nilai masing-masing kategori. Warna oranye digunakan untuk membedakan kategori dengan garis tepi hitam agar lebih jelas. Dengan menggunakan Bar Chart, dapat dengan mudah melihat kategori mana yang memiliki nilai tertinggi dan bagaimana distribusi data antar kategori secara visual.

5.4.5 Line Chart

Line Chart digunakan untuk menunjukkan perubahan suatu variabel terhadap waktu. Biasanya digunakan dalam analisis tren seperti penjualan per bulan atau suhu harian.



Gambar 5.18:Line Chart

Gambar 5.18 menunjukkan Line Chart, yang digunakan untuk menampilkan perubahan suatu variabel terhadap waktu. Diagram ini sangat berguna dalam menganalisis tren dan pola dalam data time-series.

Pada diagram ini: Sumbu X mewakili unit waktu, seperti hari, bulan, atau tahun. Sumbu Y menunjukkan nilai yang diukur dalam setiap periode waktu. Garis merah menghubungkan titik-titik data, menunjukkan pola perubahan dari waktu ke waktu. Titik-titik (marker 'o') membantu memperjelas nilai pada setiap periode waktu. Dengan Line Chart, dapat dengan mudah mengidentifikasi tren naik, tren turun, pola siklus, atau anomali dalam data, yang sangat berguna dalam pengambilan keputusan berbasis data.

5.4.6 Pie Chart

Pie Chart memvisualisasikan proporsi suatu kategori dalam bentuk lingkaran, cocok untuk menunjukkan pembagian dalam suatu kelompok.

Gambar 5.19: Pie Chart

Gambar 5.19 menunjukkan Pie Chart, yang digunakan untuk memvisualisasikan proporsi atau persentase dari berbagai kategori dalam satu dataset. Diagram ini sangat berguna dalam menunjukkan kontribusi relatif dari setiap kategori terhadap total keseluruhan.

Pada diagram ini: Setiap irisan lingkaran (*wedge*) mewakili satu kategori, dengan ukuran yang proporsional terhadap nilainya. Persentase (%) ditampilkan pada setiap kategori, menunjukkan seberapa besar kontribusi masing-masing terhadap total. Warna berbeda digunakan untuk membedakan setiap kategori agar lebih mudah diinterpretasikan. Start angle (140°) digunakan untuk mengatur rotasi diagram agar tampil lebih seimbang. Meskipun Pie Chart memberikan representasi visual yang mudah dipahami, penggunaannya lebih efektif jika jumlah kategori tidak terlalu banyak. Untuk dataset yang memiliki banyak kategori, Bar Chart biasanya lebih disarankan.

Bab 6
Basis Data dan Manajemen Data

6.1 Pengantar Basis Data

Basis data adalah kumpulan data yang terorganisir dan tersimpan secara sistematis di dalam komputer, yang dapat diakses, dikelola, dan diperbarui dengan mudah. Basis data dirancang untuk menyimpan informasi secara terstruktur sehingga memudahkan pengguna dalam melakukan operasi seperti pencarian, penyisipan, pembaruan, dan penghapusan data (Elmasri & Navathe, 2016).

Penggunaan basis data memungkinkan organisasi untuk memelihara sejumlah besar informasi dan memastikan integritas data terjaga. Kemampuan untuk menjalankan transaksi dengan andal, dan pemeliharaan keamanan data adalah aspek utama dalam pengelolaan basis data (Silberschatz et al., 2019). Dengan memanfaatkan basis data, pengguna tidak hanya dapat menyimpan data tetapi juga menganalisisnya, menarik kesimpulan, dan akhirnya membuat keputusan yang tepat, yang menunjukkan peran penting basis data dalam operasi dan penelitian modern.

Contoh penggunaan basis data dalam kehidupan sehari-hari adalah aplikasi perbankan untuk mengelola data nasabah dan data transaksi, e-commerce untuk mengelola data pelanggan, produk dan pesanan.

6.1.1 Komponen Basis Data

Komponen basis data terdiri dari beberapa elemen utama yang bekerja bersama untuk menyimpan, mengelola, dan memproses data. Berikut adalah komponen utama dalam basis data:

- Data : adalah informasi yang disimpan dalam bentuk tabel, record, atau file.
- Perangkat lunak: Database Management System (DBMS) adalah sistem yang mengelola basis data seperti MySQL, PostgreSQL, SQL Server, dan Oracle. Aplikasi pendukung yang berinteraksi dengan basis data, seperti BI tools (Power BI, Tableau) atau aplikasi berbasis web.

- Perangkat keras: seperti server, komputer, storage, dan perangkat jaringan yang digunakan untuk menjalankan sistem basis data.
- Pengguna: orang yang menggunakan atau mengelola data, seperti administrator, pengembang, dan pengguna akhir.

6.1.2 Model Data

Model data adalah struktur konseptual yang menentukan bagaimana data diorganisasi, disimpan, dan dihubungkan dalam sebuah sistem basis data. Model data berfungsi sebagai panduan dalam desain basis data, memastikan data dapat diakses dan dikelola dengan efisien. Barikut ini perbandingan jenis-jenis dari model data:

1. Model Data Relasional (Relational Model)
   Data disimpan dalam bentuk tabel yang terdiri dari baris dan kolom. Menggunakan kunci primer (primary key) dan kunci asing (foreign key) untuk menghubungkan tabel.
   Contoh produk basis data relasional seperti ditunjukan gambar 6.1.

Gambar 6.1 Produk RDBMS (Husain, 2023)

   Kelebihan: Fleksibel, mudah dimengerti, mendukung standar SQL.
   Kekurangan: Performa bisa menurun jika jumlah data sangat besar tanpa optimasi indeks.
2. Model Data Hierarkis
   Data disusun dalam bentuk pohon dengan hubungan parent-child. Setiap parent bisa memiliki banyak child, tetapi satu child hanya punya satu parent.
   Contoh: Sistem manajemen file di komputer.
   Kelebihan: Akses cepat jika struktur sesuai hierarki.

Kekurangan: Sulit dimodifikasi jika ada perubahan hubungan antar data.

3. Model Data Jaringan (*Network Model*)
   Mirip dengan model hierarkis, tetapi satu child bisa memiliki banyak parent.
   Menggunakan konsep graph dengan node dan edge untuk hubungan antar data.
   Contoh: Database industri perbankan dan supply chain.
   Kelebihan: Fleksibel dalam mendeskripsikan hubungan kompleks.
   Kekurangan: Kompleks dalam implementasi dan maintenance.
4. Model Data Berorientasi Objek (*Object-Oriented Model*)
   Menggabungkan konsep basis data dengan pemrograman berorientasi objek (OOP). Data disimpan dalam bentuk objek yang memiliki atribut dan metode.
   Kelebihan: Mendukung objek kompleks dan data multimeda.
   Kekurangan: Kurang umum dibanding model relasional.
5. Model Data NoSQL
   Digunakan untuk data yang tidak terstruktur atau semi-terstruktur.
   Kelebihan: Skalabilitas tinggi dan cocok untuk big data.
   Kekurangan: Tidak selalu mendukung transaksi ACID seperti model relasional.

6.2 Sistem Manajemen Basis Data

Sistem Manajemen Basis Data (*DBMS - Database Management System*) adalah perangkat lunak yang digunakan untuk membuat, mengelola, dan mengakses basis data secara efisien. DBMS memungkinkan pengguna untuk menyimpan, mengambil, memperbarui, dan menghapus data dengan mudah serta memastikan integritas dan keamanan data (Elmasri & Navathe, 2015). Sebelum DBMS digunakan secara luas dikalangan industri, data disimpan dalam file secara manual, sehingga dapat menyebabkan:

- Redundansi Data: data yang sama dapat disimpan di beberapa file yang berbeda.
- Inkonsistensi Data: karena data disimpan di berbagai file, perubahan pada satu file mungkin tidak terupdate pada file lain.
- Kesulitan dalam Akses Data: mengakses data yang tersebar di berbagai file memerlukan waktu dan usaha yang lebih besar, terutama jika data tersebut perlu digabungkan atau dianalisis.

- Masalah Keamanan: sulit untuk mengontrol akses ke data, meningkatkan risiko kebocoran atau penyalahgunaan data.
- Kesulitan dalam Backup dan Recovery: karena system sistem tidak terpusat, proses backup dan recovery data menjadi lebih rumit dan rentan terhadap kesalahan.

6.2.1 Karakteristik SMBD Relasional

Sistem Manajemen Basis Data Relasional adalah suatu sistem dengan struktur dan metode yang jelas untuk menyimpan dan mengambil isi data (Roberts, 2024). DBMS relasional memiliki karakteristik seperti berikut ini (Ramez Elmasri, Shamkant Navathe, 2015):

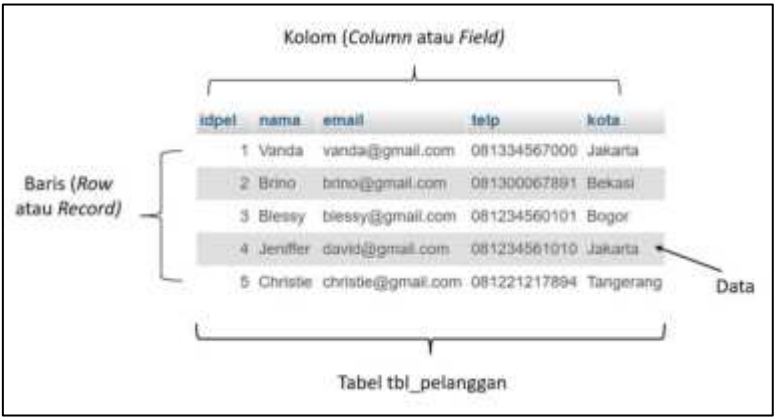
- Data disimpan dalam tabel: data diorganisasi dalam tabel yang terdiri dari baris dan kolom.
- Primary Key: setiap tabel memiliki primary key yang unik untuk mengidentifikasi setiap baris.
- Relasi antar tabel: tabel-tabel dapat dihubungkan menggunakan foreign key.
- ACID Compliance: RDBMS mendukung transaksi yang memenuhi prinsip ACID (Atomicity, Consistency, Isolation, Durability).
- Skema terstruktur: struktur database (tabel, kolom, tipe data) harus didefinisikan sebelum data dimasukkan.

6.2.2 Struktur Penyimpanan DBMS Relasional

Sebagai pengelola data, sistem database relasional memiliki struktur hirarki objek penyimpanan sebagai berikut:

1. Database
2. Tabel
3. Kolom

Database biasanya terdiri dari beberapa tabel, dan setiap tabel terdiri dari beberapa kolom atau field (Roberts, 2024). Di setiap database, tabel dan kolom memiliki nama sendiri sebagai identitas mereka. Tabel dan kolom inilah yang akan diisi data yang kemudian membentuk baris data (row atau record). Gambar 6.2 adalah contoh suatu tabel dalam database. Setiap tabel dalam database harus memiliki nama.



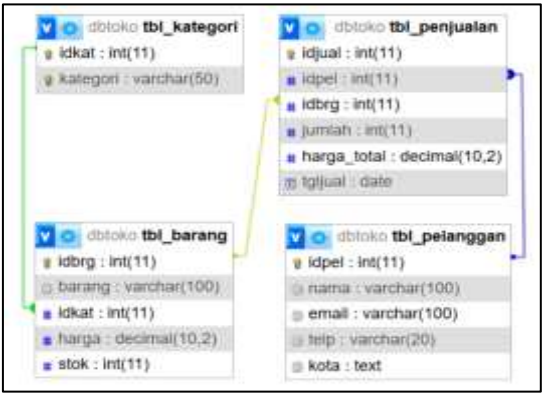
Gambar 6.2 Tabel dan kolom

Pada gambar 6.2 nama tabel adalah tbl_pelanggan, nama kolom adalah idpel, nama, email, telp dan kota.

6.3 Query SQL untuk Analisis Data

SQL adalah singkatan dari *Structured Query Language*. SQL adalah bahasa yang digunakan untuk mengakses, mengelola, dan menganalisis data dalam database. Dalam analisis data, SQL membantu pengguna mengambil, memfilter, dan memproses data dengan lebih efisien (Teate, 2021).

Agar memahami penggunaan query SQL untuk keperluan analisis data maka akan menggunakan tabel-tabel seperti ditunjukkan gambar 6.3 berikut ini:



Gambar 6.3: Desain tabel

Masing-masing tabel tersebut di isi data sesuai dengan kebutuhan, berikut adalah data yang sudah di input ke masing-masing tabel.

1. Data pada tabel tbl_kategori

idkat	kategori
1	Elektronik
2	Furnitur
3	Pakaian

2. Data pada tabel tbl_barang

idbrg	barang	idkat	harga	stok
1	Laptop	1	10000000.00	10
2	Smartphone	1	5000000.00	20
3	Sofa	2	3000000.00	10
4	Meja	2	1500000.00	8
5	Jaket	3	500000.00	15
6	Kaos	3	200000.00	30

3. Data pada tabel tbl_pelanggan

idpel	nama	email	telp	kota
1	Vanda	vanda@gmail.com	081334567000	Jakarta
2	Brino	brino@gmail.com	081300067891	Bekasi
3	Blessy	blessy@gmail.com	081234560101	Bogor
4	Jeniffer	david@gmail.com	081234561010	Jakarta
5	Christie	christie@gmail.com	081221217894	Tangerang

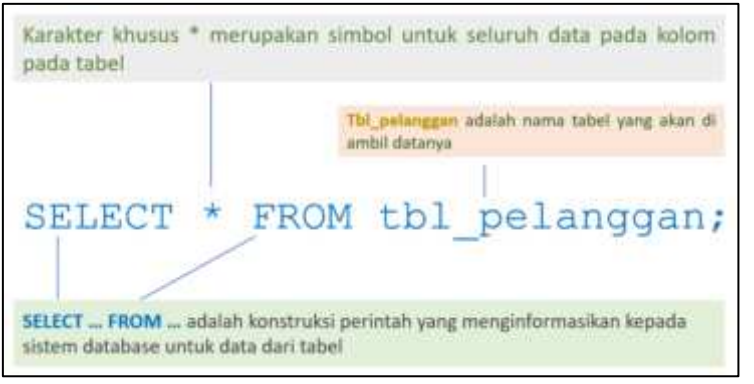
4. Data pada tabel tbl_penjualan

idjual	idpel	idbrg	jumlah	harga_total	tgljual
1	1	1	1	10000000.00	2024-01-01
2	2	2	2	10000000.00	2024-01-02
3	3	3	3	9000000.00	2024-01-03
4	4	4	1	1500000.00	2024-01-04
5	5	5	3	1500000.00	2024-01-05
6	1	6	2	400000.00	2024-01-06
7	2	1	1	10000000.00	2024-01-07
8	3	2	1	5000000.00	2024-01-08
9	4	3	2	6000000.00	2024-01-09
10	5	4	1	1500000.00	2024-01-10

Desain tabel pada gambar 6.3 dan data pada masing-masing table akan digunakan untuk analisis data pada bagian 6.3.1-6.3.8, serta contoh visuaalisasi data pada bagian 6.4.

6.3.1 Query Dasar

Untuk mengakses data di database, dapat menggunakan pernyataan `SELECT`. Pernyataan `SELECT` menyatakan kolom-kolom mana saja yang ingin tampilkan dari suatu tabel di database. Pernyataan `SELECT` tidak berdiri sendiri. Setelah menyatakan kolom - kolom yang ingin ditampilkan, dilanjutkan dengan FROM. Di FROM inilah dinyatakan dari tabel mana data yang ingin ditampilkan. `SELECT... FROM...` adalah pernyataan paling sederhana di SQL, dan merupakan bagian utama dari query (MySQL Fundamental, 2021). Query dasar dan sederhana pernyataan `SELECT` yang berfungsi untuk menampilkan seluruh kolom, ditunjukan pada gambar 6.3 sebagai berikut:

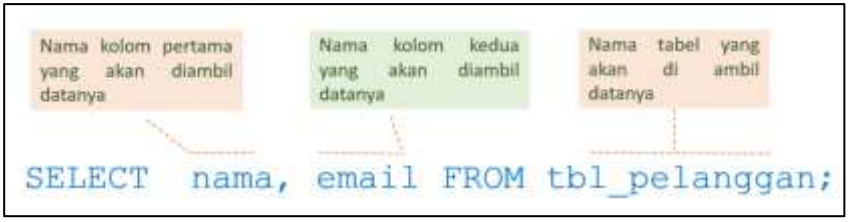


Gambar 6.4 Pernyataan SELECT

Penjelasan:

- Pernyataan `SELECT` digunakan untuk menginformasikan kepada sistem basis data untuk ingin mengambil data.
- Tanda * (bintang) artinya seluruh kolom perlu diambil dari tabel yang dirujuk. Tanda ini sering juga disebut sebagai wildcard.
- `FROM [NAMA_TABEL]`, artinya tabel yang akan diambil datanya.
- Tanda `;` (titik koma) adalah tanda yang menyatakan akhir dari perintah `SELECT` atau perintah SQL lainnya.

Tabel tbl_pelanggan memiliki lebih dari satu kolom data. Kadang-kadang hanya ingin menampilkan data tertentu saja. Gambar 6.4 menunjukan perintah untuk mengambil kolom tertentu dari suatu tabel.



Gambar 6.5 Menampilkan kolom tertentu

Data juga dapat ditampilkan menggunakan kriteria tertentu, misalnya menampilkan daftar pelanggan yang berasal dari kota "Jakarta".

Gambar 6.6 Klause WHERE

Dalam SQL, `WHERE` adalah klausa yang digunakan untuk memfilter data dalam perintah `SELECT`, `UPDATE`, `DELETE` dan `INSERT`. `WHERE` berfungsi untuk menentukan **kondisi** yang harus dipenuhi oleh baris data yang akan diproses. Hanya baris yang memenuhi kondisi ini yang akan ditampilkan.

6.3.2 Pengelompokan Data

Pengelompokan data (grouping) dalam SQL adalah proses mengelompokkan baris data berdasarkan nilai tertentu dari satu atau beberapa kolom. Pengelompokan data biasanya digunakan bersama dengan fungsi agregasi. Berikut adalah penerapan query SQL untuk pengelompokan data:

1. Menggunakan klausa `GROUP BY` dan fungsi agregasi `COUNT()` untuk mengelompokkan pelanggan berdasarkan kota dan menghitung jumlah pelanggan di setiap kota.

```
SELECT kota, COUNT(*) AS total_pelanggan
FROM tbl_pelanggan
GROUP BY kota;
```

Output:

kota	total_pelanggan
Bekasi	1
Bogor	1
Jakarta	2
Tangerang	1

2. Menggunakan klausa HAVING untuk memfilter hasil pengelompokan berdasarkan kondisi tertentu. HAVING bekerja setelah mengeksekusi fungsi agregasi. Contoh menghitung jumlah pelanggan per kota, lalu menampilkan hanya kota yang memiliki lebih dari 1 pelanggan.

```
SELECT kota, COUNT(*) AS total_pelanggan
FROM tbl_pelanggan
GROUP BY kota
HAVING total_pelanggan > 1;
```

Output:

kota	total_pelanggan
Jakarta	2

6.3.3 Agregasi Data

Fungsi agregat dalam SQL digunakan untuk melakukan perhitungan pada sekumpulan nilai dalam sebuah kolom dan mengembalikan satu nilai ringkasan. Fungsi ini sering digunakan bersama `GROUP BY` untuk mengelompokkan data berdasarkan kategori tertentu (Teate, 2021). Berikut adalah penerapan fungsi-fungsi agregat:

1. Query untuk menghitung total pendapatan dari semua transaksi di tabel tbl_penjualan:

```
SELECT SUM(harga_total) AS total_pendapatan
FROM tbl_penjualan;
```

Output:

total_pendapatan
54900000.00

2. Query untuk menghitung rata-rata harga barang yang dijual:

```
SELECT AVG(harga) AS rata_rata_harga
FROM tbl_barang;
```

Output:

rata_rata_harga
3366666.666667

3. Query untuk menghitung jumlah pelanggan unik yang pernah melakukan transaksi:

```
SELECT COUNT(DISTINCT idpel) AS total_pelanggan_unik
FROM tbl_penjualan;
```

Output:

total_pelanggan_unik
5

4. Query untuk menampilkan barang yang paling banyak terjual:

```
SELECT b.barang AS "Nama Barang", SUM(p.jumlah) AS
Total_Terjual
FROM tbl_penjualan p
JOIN tbl_barang b ON p.idbrg = b.idbrg
GROUP BY b.barang
ORDER BY Total_Terjual DESC
LIMIT 1;
```

Output:

Nama Barang	Total_Terjual
Sofa	5

5. Query untuk menghitung total pendapatan berdasarkan kota pelanggan:

```
SELECT pl.kota, SUM(p.harga_total) AS total_pendapatan
FROM tbl_penjualan p
JOIN tbl_pelanggan pl ON p.idpel = pl.idpel
GROUP BY pl.kota
ORDER BY total_pendapatan DESC;
```

Output:

kota	total_pendapatan 1
Bekasi	20000000.00
Jakarta	17900000.00
Bogor	14000000.00
Tangerang	3000000.00

6. Query untuk menampilkan harga barang tertinggi dan terendah di tabel tbl_barang:

```
SELECT
    MAX(harga) AS harga_tertinggi,
    MIN(harga) AS harga_terendah
FROM tbl_barang;
```

Output:

harga_tertinggi	harga_terendah
10000000.00	200000.00

6.3.4 Join

Pada dasarnya sebuah aplikasi dibangun dengan menggunakan banyak tabel. Tabel-tabel ini akan berelasi antara satu dengan yang lainnya. JOIN digunakan dalam SQL untuk menggabungkan data dari dua atau lebih table. Jenis-jenis `JOIN` ditunjukan pada table 6.1 berikut ini:

Tabel 6.1: Jenis-jenis JOIN

Jenis	Deskripsi
`INNER JOIN`	Mengambil data yang memiliki kecocokan di kedua tabel
`LEFT JOIN`	Mengambil semua data dari tabel kiri dan data yang cocok dari tabel kanan
`RIGHT JOIN`	Mengambil semua data dari tabel kanan dan data yang cocok dari tabel kiri

Jenis	Deskripsi
FULL JOIN	Mengambil semua data dari kedua tabel (kombinasi LEFT JOIN dan RIGHT JOIN)

Berikut adalah penerapan query SQL untuk join tabel:

1. Query untuk menampilkan semua data transaksi dengan detail. Sumber data yang ditampilkan berasal dari empat tabel yaitu tabel tbl_penjualan, tbl_pelanggan, tbl_barang dan tbl_kategori.

```
SELECT j.idpel AS "Kode Pelanggan", p.nama AS "Nama
Pelanggan", b.barang AS Barang, k.kategori AS Kategori,
j.jumlah AS Jumlah, j.harga_total AS "Harga Total",
j.tgljual AS "Tanggal Jual"
FROM tbl_penjualan j
JOIN tbl_pelanggan p ON j.idpel = p.idpel
JOIN tbl_barang b ON j.idbrg = b.idbrg
JOIN tbl_kategori k ON b.idkat = k.idkat;
```

Output:

Kode Pelanggan	Nama Pelanggan	Barang	Kategori	Jumlah	Harga Total	Tanggal Jual
1	Vanda	Laptop	Elektronik	1	10000000.00	2024-01-01
1	Vanda	Kaos	Pakaian	2	400000.00	2024-01-06
2	Brino	Smartphone	Elektronik	2	10000000.00	2024-01-02
2	Brino	Laptop	Elektronik	1	10000000.00	2024-01-07
3	Blessy	Sofa	Furnitur	3	9000000.00	2024-01-03
3	Blessy	Smartphone	Elektronik	1	5000000.00	2024-01-08
4	Jeniffer	Meja	Furnitur	1	1500000.00	2024-01-04
4	Jeniffer	Sofa	Furnitur	2	6000000.00	2024-01-09
5	Christie	Jaket	Pakaian	3	1500000.00	2024-01-05
5	Christie	Meja	Furnitur	1	1500000.00	2024-01-10

2. Query untuk menampilkan total penjualan per pelanggan.

```
SELECT p.nama AS "Nama Pelanggan", SUM(j.harga_total) AS
Total_Belanja
FROM tbl_penjualan j
JOIN tbl_pelanggan p ON j.idpel = p.idpel
GROUP BY p.nama;
```

Output:

Nama Pelanggan	Total_Belanja
Blessy	14000000.00
Brino	20000000.00
Christie	3000000.00
Jeniffer	7500000.00
Vanda	10400000.00

3. Query untuk menampilkan data barang dengan penjualan tertinggi

```
SELECT b.barang AS Barang, SUM(j.jumlah) AS
Total_Penjualan
FROM tbl_penjualan j
JOIN tbl_barang b ON j.idbrg = b.idbrg
GROUP BY b.barang
ORDER BY Total_Penjualan DESC
LIMIT 1;
```

Output:

Barang	Total_Penjualan
Sofa	5

4. Query untuk menampilkan stok barang yang tersisa setelah penjualan

```
SELECT b.barang, b.stok - COALESCE(SUM(j.jumlah), 0) AS
Sisa_Stok
FROM tbl_barang b
LEFT JOIN tbl_penjualan j ON b.idbrg = j.idbrg
GROUP BY b.barang, b.stok;
```

Output:

barang	Sisa_Stok
Jaket	12
Kaos	28
Laptop	8
Meja	6
Smartphone	17
Sofa	5

6.3.5 Subquery

Subquery atau query bersarang adalah query di dalam query. Subquery digunakan untuk mendapatkan hasil yang digunakan dalam bagian lain dari query utama, seperti di bagian `SELECT`, `FROM` atau `WHERE`. Berikut adalah beberapa contoh penggunaan subquery:

1. Subquery untuk menampilkan pelanggan yang telah melakukan transaksi lebih dari 12 juta:

```
SELECT nama, email
FROM tbl_pelanggan
WHERE idpel IN (
    SELECT idpel
    FROM tbl_penjualan
    GROUP BY idpel
    HAVING SUM(harga_total) > 12000000
);
```

Subquery pada bagian WHERE mencari ID pelanggan yang total belanjanya lebih dari 12 juta, dan query utama menampilkan data pelanggan yang memenuhi kondisi tersebut.

Output:

nama	email
Brino	brino@gmail.com
Blessy	blessy@gmail.com

2. Subquery untuk menampilkan nama barang beserta total pendapatan dari penjualan barang tersebut.

```
SELECT b.barang,
       (SELECT SUM(harga_total)
        FROM tbl_penjualan
        WHERE idbrg = b.idbrg) AS total_pendapatan
FROM tbl_barang b;
```

Subquery di bagian SELECT menghitung total pendapatan untuk setiap barang berdasarkan ID barang yang ada di query utama.

Output:

barang	total_pendapatan
Laptop	20000000.00
Smartphone	15000000.00
Sofa	15000000.00
Meja	3000000.00
Jaket	1500000.00
Kaos	400000.00

3. Subquery untuk melihat total transaksi per pelanggan dan kemudian hanya menampilkan pelanggan yang total transaksinya lebih dari 10.000.000.

```
SELECT tmp.idpel, tmp.total_transaksi
FROM (
    SELECT idpel, SUM(harga_total) AS total_transaksi
    FROM tbl_penjualan
    GROUP BY idpel
) AS tmp
WHERE tmp.total_transaksi > 10000000;
```

Subquery di dalam FROM menghasilkan tabel sementara yang berisi ID pelanggan dan total transaksi, yang kemudian disaring di query utama untuk mendapatkan pelanggan yang total transaksinya lebih dari 10.000.000.

Output:

idpel	total_transaksi
1	10400000.00
2	20000000.00
3	14000000.00

6.3.6 Common Table Expressions (CTE)

Common Table Expression (CTE) adalah fitur SQL yang digunakan untuk membuat query sementara yang diberi nama dan dapat digunakan ulang dalam query utama. CTE sangat membantu untuk menyederhanakan query yang kompleks dan meningkatkan keterbacaan kode SQL (Tanimura, 2021). Misalnya menggunakan subquery kompleks atau ketika ingin menjalankan

query rekursif. CTE didefinisikan menggunakan kata kunci WITH, diikuti oleh nama CTE, dan kemudian diikuti oleh query yang mendefinisikan data yang ingin Anda gunakan. Hasil CTE dapat digunakan seperti tabel sementara dalam query SQL. Sintaks dasar CTE adalah:

```
WITH nama_cte AS (
    SELECT kolom1, kolom2, ...
    FROM nama_tabel
    WHERE kondisi
)
SELECT *
FROM nama_cte;
```

Berikut adalah query SQL menggunakan Common Table Expressions (CTE) untuk menganalisis total penjualan per kategori barang dan jumlah pelanggan unik yang bertransaksi:

```
WITH PenjualanPerKategori AS (
    SELECT
        k.kategori,
        SUM(p.jumlah) AS total_terjual,
        SUM(p.harga_total) AS total_pendapatan
    FROM tbl_penjualan p
    JOIN tbl_barang b ON p.idbrg = b.idbrg
    JOIN tbl_kategori k ON b.idkat = k.idkat
    GROUP BY k.kategori
),
PelangganPerKategori AS (
    SELECT
        k.kategori,
        COUNT(DISTINCT p.idpel) AS total_pelanggan
    FROM tbl_penjualan p
    JOIN tbl_barang b ON p.idbrg = b.idbrg
    JOIN tbl_kategori k ON b.idkat = k.idkat
    GROUP BY k.kategori
)
SELECT
    ppk.kategori,
    ppk.total_terjual,
    ppk.total_pendapatan,
    ppc.total_pelanggan
FROM PenjualanPerKategori ppk
```

```
JOIN PelangganPerKategori ppc ON ppk.kategori = ppc.kategori
ORDER BY ppk.total_pendapatan DESC;
```

Hasil query SQL menggunakan Common Table Expressions (CTE) ditunjukan pada gambar 6.5 berikut ini:

kategori	total_terjual	total_pendapatan	total_pelanggan
Elektronik	5	35000000.00	3
Furnitur	7	18000000.00	3
Pakaian	5	1900000.00	2

Gambar 6.7 Output query CTE

Penjelasan:

- CTE PenjualanPerKategori: menghitung total barang terjual dan total pendapatan per kategori barang.
- CTE PelangganPerKategori: menghitung jumlah pelanggan unik yang membeli barang dalam kategori tersebut.
- Query Utama: menggabungkan kedua CTE berdasarkan kategori barang dan menampilkan data dengan total pendapatan tertinggi di urutan pertama.

6.3.7 SQL Window Functions

SQL Window Functions adalah fungsi yang memungkinkan untuk melakukan perhitungan di atas subset data tanpa mengubah hasil query utama. Fungsi ini dihitung untuk setiap baris dalam hasil query, tetapi hasilnya tidak mengubah jumlah baris yang dikembalikan(Tanimura, 2021).

Window Functions sering digunakan dalam analisis data yang melibatkan agregasi per kelompok tetapi tetap ingin mempertahankan data baris individu (Zhiyanov, 2023). Beberapa fungsi jendela yang sering digunakan adalah:

1. `ROW_NUMBER()` – Memberikan nomor urut untuk setiap baris. Contoh: query untuk menampilkan semua pelanggan dengan nomor urut berdasarkan urutan nama pelanggan.

```
SELECT idpel, nama, email,
       ROW_NUMBER() OVER (ORDER BY nama) AS row_num
FROM tbl_pelanggan;
```

Fungsi ROW_NUMBER() memberikan nomor urut berdasarkan abjad nama pelanggan, dimulai dari 1.

Output:

idpel	nama	email	row_num
3	Blessy	blessy@gmail.com	1
2	Brino	brino@gmail.com	2
5	Christie	christie@gmail.com	3
4	Jeniffer	david@gmail.com	4
1	Vanda	vanda@gmail.com	5

2. RANK() – Memberikan peringkat, tetapi ada kemungkinan peringkat yang sama jika ada nilai yang sama.

   Contoh: query untuk menampilkan barang beserta harga dan peringkatnya berdasarkan harga.

```
SELECT idbrg, barang, harga,
       RANK() OVER (ORDER BY harga DESC) AS rank
FROM tbl_barang;
```

Fungsi RANK() memberi peringkat pada barang berdasarkan harga. Jika ada dua barang dengan harga yang sama, keduanya akan mendapat peringkat yang sama, dan peringkat berikutnya akan terlewat (misalnya, dua barang dengan peringkat 1, maka peringkat berikutnya adalah 3).

Output:

idbrg	barang	harga	rank
1	Laptop	10000000.00	1
2	Smartphone	5000000.00	2
3	Sofa	3000000.00	3
4	Meja	1500000.00	4
5	Jaket	500000.00	5
6	Kaos	200000.00	6

3. DENSE_RANK() – Memberikan peringkat, tetapi tidak ada celah jika ada nilai yang sama.
4. NTILE(n) – Membagi hasil menjadi n grup yang hampir sama besar.
5. SUM(), AVG(), MIN(), MAX() – Fungsi agregasi yang digunakan dalam window functions.
6. LEAD() dan LAG() – Mengambil nilai sebelumnya atau berikutnya. LEAD() digunakan untuk mendapatkan nilai berikutnya dalam urutan baris, sementara LAG() digunakan untuk mendapatkan nilai sebelumnya.

6.3.8 Time Series di SQL

Time Series adalah jenis data yang dikumpulkan atau direkam dalam interval waktu tertentu, seperti harian, mingguan, bulanan, atau tahunan (Teate, 2021). Data ini digunakan untuk menganalisis tren, musiman, dan pola waktu untuk membuat prediksi atau keputusan bisnis.

Dalam SQL, kita dapat mengelola dan menganalisis data time series menggunakan berbagai teknik, termasuk agregasi, window functions, dan operasi time-based lainnya.

Operasi Dasar pada Time Series adalah mengambil data dalam rentang waktu tertentu, agregasi data per periode, menghitung perubahan (Lag dan Lead). Moving Average adalah salah satu metode analisis time series dan ARIMA (AutoRegressive Integrated Moving Average) adalah contoh metode prediksi time series.

6.4 Visualisasi Data

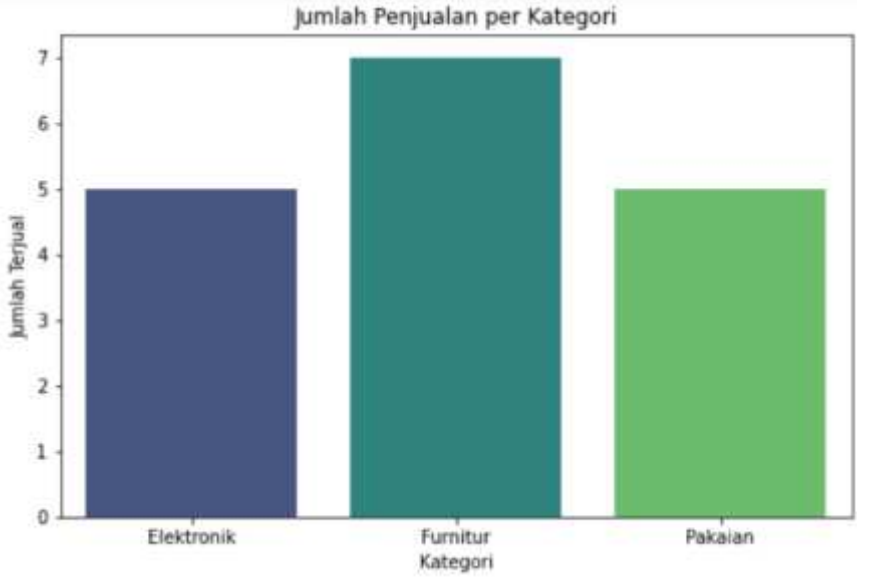
Visualisasi data adalah proses menampilkan data dalam bentuk grafis atau visual seperti grafik, diagram, peta, atau dashboard interaktif (Healy, 2018). Tujuannya adalah untuk membantu memahami pola, tren, dan hubungan dalam data dengan lebih cepat dan intuitif dibandingkan hanya melihat angka atau tabel.

Manfaat visualisasi data adalah meningkatkan pemahaman agar data yang kompleks lebih mudah dianalisis, memudahkan pengambilan keputusan karena informasi lebih cepat diinterpretasikan. Berikut adalah contoh visualisasi data dari database MySQL/MariaDB menggunakan library Pandas, Matplotlib dan Seaborn yang ditulis menggunakan bahasa pemrograman Python:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import mysql.connector

# Koneksi ke MySQL
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="dbtoko"
)

# Query data
query_kategori = "SELECT * FROM tbl_kategori"
query_barang = "SELECT * FROM tbl_barang"
query_pelanggan = "SELECT * FROM tbl_pelanggan"
query_penjualan = "SELECT * FROM tbl_penjualan"

kategori = pd.read_sql(query_kategori, db)
barang = pd.read_sql(query_barang, db)
pelanggan = pd.read_sql(query_pelanggan, db)
penjualan = pd.read_sql(query_penjualan, db)

# Gabungkan Data
penjualan = penjualan.merge(barang,
on='idbrg').merge(kategori, on='idkat').merge(pelanggan,
on='idpel')

# 1. Jumlah Penjualan per Kategori
penjualan_per_kategori =
penjualan.groupby('kategori')['jumlah'].sum()
plt.figure(figsize=(8, 5))
sns.barplot(x=penjualan_per_kategori.index,
y=penjualan_per_kategori.values, palette='viridis')
plt.title('Jumlah Penjualan per Kategori')
plt.xlabel('Kategori')
plt.ylabel('Jumlah Terjual')
plt.show()

# 2. Total Pendapatan per Kota Pelanggan
```

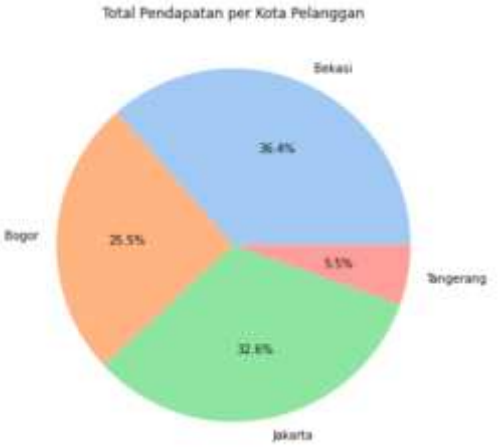
```
total_pendapatan_kota                                              =
penjualan.groupby('kota')['harga_total'].sum()
plt.figure(figsize=(7, 7))
plt.pie(total_pendapatan_kota,
labels=total_pendapatan_kota.index,        autopct='%1.1f%%',
colors=sns.color_palette('pastel'))
plt.title('Total Pendapatan per Kota Pelanggan')
plt.show()

# Tutup koneksi database
db.close()
```

Hasil dalam bentuk visualisasi data berdasarkan jumlah penjualan perkategori barang:



Gambar 6.8 Visualisasi penjualan barang per kategori

Hasil dalam bentuk visualisasi data berdasarkan total pendapatan pelanggan perkota:



Gambar 6.9 Visualisasi pendapatan pelanggan perkota

6.5 NoSQL

NoSQL (Not Only SQL) adalah sistem manajemen basis data yang tidak menggunakan model relasional (tidak berbasis tabel seperti SQL). NoSQL dirancang untuk menangani data dalam skala besar, dengan fleksibilitas yang lebih tinggi dibandingkan database relasional (McCreary & Kelly, 2013). Karakteristik NoSQL adalah sebagai berikut:

- Skalabilitas tinggi: Mampu menangani volume data yang besar dengan pendekatan horizontal scaling.
- Fleksibilitas skema: Tidak memiliki skema yang tetap, sehingga lebih mudah beradaptasi dengan perubahan data.
- Kinerja tinggi: Dapat menangani operasi baca/tulis dengan cepat dibandingkan database relasional.
- Dukungan untuk data terdistribusi: Cocok untuk sistem yang berjalan di banyak server sekaligus.

Adapun Jenis-Jenis dari NoSQL adalah:

- Key-Value Store: data disimpan dalam format pasangan kunci-nilai. Contoh: Redis, Riak, DynamoDB.

- Document Store: data disimpan dalam format dokumen seperti JSON atau BSON. Contoh: MongoDB, CouchDB.
- Column Family Store: data disimpan dalam bentuk kolom yang memungkinkan pencarian yang efisien. Contoh: Apache Cassandra, HBase.
- Graph Database: digunakan untuk menyimpan hubungan antar data dalam bentuk grafik. Contoh: Neo4j, ArangoDB.

NoSQL memiliki kelebihan yaitu mudah diskalakan secara horizontal, cocok untuk Big Data dan aplikasi real-time, skema fleksibel memungkinkan perubahan struktur data dengan mudah dan kinerja tinggi untuk aplikasi yang membutuhkan respons cepat. Sedangkan kekurangannya adalah tidak mendukung transaksi ACID secara penuh, kurangnya standar query seperti pada SQL.

Bab 7
Algoritma dan Teknik Machine Learning

7.1 Pengantar Machine Learning

7.1.1 Definisi Machine Learning

Machine Learning (ML) adalah cabang dari kecerdasan buatan (AI) yang memungkinkan komputer untuk belajar dari data tanpa diprogram secara eksplisit. Dengan kata lain, ML memberikan kemampuan kepada mesin untuk membuat keputusan atau memprediksi hasil berdasarkan pola yang diidentifikasi dalam data historis (Suri & Cabri, 2014)

7.1.2 Jenis-jenis Machine Learning

a. Supervised Learning (Pembelajaran Terawasi):
   - Menggunakan data berlabel untuk melatih model.
   - Tujuannya adalah untuk memetakan input ke output berdasarkan contoh yang telah diberikan.
   - Contoh algoritma: Regresi Linear, Regresi Logistik, Decision Tree, Support Vector Machine (SVM).

b. Unsupervised Learning (Pembelajaran Tak Terawasi):
   - Tidak menggunakan label pada data.
   - Fokusnya pada pengelompokan atau menemukan struktur dalam data.
   - Contoh algoritma: K-Means Clustering, Hierarchical Clustering, Principal Component Analysis (PCA).

c. Semi-Supervised Learning (Pembelajaran Semi-Terawasi):
   - Menggunakan kombinasi data berlabel dan tidak berlabel untuk pelatihan.
   - Berguna ketika label sulit atau mahal untuk diperoleh.

d. Reinforcement Learning (Pembelajaran Penguatan):
   - Model belajar melalui interaksi dengan lingkungan dan mendapatkan umpan balik berupa reward atau penalty.
   - Contoh aplikasi: robotika, sistem rekomendasi, permainan komputer.

7.1.3 Elemen Utama: Data, Model, dan Evaluasi

Dalam machine learning, data, model, dan evaluasi merupakan elemen utama yang menjadi fondasi pengembangan sistem yang cerdas. Data adalah bahan mentah yang dikumpulkan dari berbagai sumber, seperti sensor, log transaksi, atau media sosial, dan diolah menjadi format yang dapat digunakan untuk pelatihan model. Kualitas dan relevansi data sangat memengaruhi kinerja sistem. Model adalah representasi matematis atau algoritma yang dirancang untuk mengenali pola dari data, sehingga mampu membuat prediksi atau keputusan. Berbagai algoritma, seperti regresi, decision tree, dan neural network, dapat digunakan tergantung pada kompleksitas masalah. Setelah model dibangun, evaluasi dilakukan untuk mengukur efektivitasnya. Dengan menggunakan metrik seperti akurasi, precision, recall, dan F1-score, evaluasi membantu memastikan bahwa model bekerja sesuai harapan dan dapat diandalkan untuk digunakan dalam dunia nyata. Interaksi ketiga elemen ini menjadi kunci keberhasilan dalam membangun solusi berbasis machine learning.

1. Data
   - Data adalah fondasi dari setiap proses machine learning. Ini mencakup informasi mentah yang diolah untuk menemukan pola atau membuat prediksi.
   - Kualitas data sangat penting karena akan memengaruhi akurasi model.
2. Model
   - Model adalah algoritma matematis yang digunakan untuk mempelajari pola dari data.
   - Model ini dapat berupa regresi linear, pohon keputusan, neural network, atau metode lainnya yang sesuai dengan masalah yang sedang diselesaikan.
3. Evaluasi
   - Evaluasi adalah proses mengukur kinerja model menggunakan metrik tertentu seperti akurasi, precision, recall, dan F1-score.

- Evaluasi membantu menentukan apakah model memenuhi kebutuhan atau perlu penyempurnaan.

7.2 Algoritma Dasar Machine Learning

Algoritma dasar machine learning adalah fondasi utama dalam pengembangan sistem pembelajaran mesin yang mampu mempelajari pola dari data dan membuat prediksi atau keputusan. Algoritma ini dibagi ke dalam tiga kategori utama: supervised learning, unsupervised learning, dan reinforcement learning. Dalam supervised learning, algoritma seperti regresi linear, decision tree, dan support vector machine (SVM) digunakan untuk mempelajari hubungan antara input dan output berdasarkan data yang berlabel. Unsupervised learning, seperti clustering dengan K-Means atau reduksi dimensi menggunakan PCA, berfokus pada menemukan pola tersembunyi dalam data tanpa label. Sementara itu, reinforcement learning melibatkan pembelajaran melalui umpan balik berbasis reward dalam interaksi dengan lingkungan. Algoritma dasar ini, dengan keunggulannya masing-masing, menjadi dasar dalam berbagai aplikasi modern seperti prediksi, klasifikasi, dan analisis data kompleks (Fairuzabadi, Adytia, et al., 2024).

7.2.1 Regresi

Regresi adalah salah satu algoritma dasar dalam machine learning yang termasuk dalam kategori supervised learning. Algoritma ini digunakan untuk memodelkan hubungan antara variabel input (prediktor atau fitur) dan output (target). Secara khusus, regresi berfokus pada memprediksi nilai kontinu dari target berdasarkan input yang diberikan

Contoh jenis regresi adalah regresi linear, yang mencari garis lurus terbaik untuk meminimalkan kesalahan antara nilai prediksi dan nilai sebenarnya, dan regresi logistik, yang memprediksi probabilitas kejadian untuk klasifikasi biner menggunakan fungsi logistik. Selain itu, regresi polynomial dapat digunakan untuk menangani hubungan non-linear dengan menambahkan pangkat variabel input, sementara regresi Ridge dan Lasso membantu mengatasi overfitting melalui regularisasi. Regresi menjadi alat penting dalam berbagai aplikasi, seperti memprediksi harga rumah, menganalisis tren penjualan, atau memperkirakan nilai ekonomi berdasarkan data historis. Meskipun sederhana, algoritma ini memiliki fleksibilitas untuk memberikan wawasan signifikan dalam analisis data.

7.2.1.1 Regresi Linier

Regresi linier adalah algoritma machine learning yang paling sederhana dan sering digunakan untuk memodelkan hubungan antara variabel independen (X) dan variabel dependen (Y) dalam bentuk hubungan linear. Model ini bekerja dengan mencari garis lurus terbaik yang meminimalkan error, biasanya menggunakan metode kuadrat terkecil (least squares). Persamaan umumnya adalah $Y=b0+b1X+\epsilon$, di mana $b0$ adalah intersep (nilai Y ketika $X=0$), $b1$ adalah kemiringan garis (koefisien regresi yang menunjukkan seberapa besar perubahan Y untuk setiap unit perubahan X), dan ϵ adalah error atau residu. Regresi linier sangat cocok digunakan untuk data dengan hubungan yang hampir linier, seperti memprediksi harga rumah berdasarkan luas atau memperkirakan penjualan berdasarkan anggaran pemasaran. Namun, model ini memiliki keterbatasan, seperti ketidakmampuannya menangani hubungan non-linear dan sensitivitas terhadap outlier. Meski demikian, regresi linier tetap menjadi alat yang penting dalam analisis data karena kesederhanaannya dan interpretasi yang mudah (Montgomery et al., 2012).

7.2.1.2 Regresi Logistik

Regresi logistik adalah jenis regresi dalam machine learning yang digunakan untuk memodelkan probabilitas kejadian pada masalah klasifikasi biner, di mana output hanya memiliki dua kategori, seperti "ya" atau "tidak". Algoritma ini memprediksi nilai probabilitas menggunakan fungsi sigmoid, yang membatasi output dalam rentang 0 hingga 1. Model matematisnya adalah $P(Y = 1|X) = \frac{1}{1+e^{-(b0+b1X)}}$ di mana $b0$ adalah intersep, $b1$ adalah koefisien regresi, dan X adalah variabel input. Output berupa probabilitas tersebut biasanya dikonversi menjadi kelas dengan menetapkan ambang batas, seperti 0,5. Regresi logistik efektif untuk menangani hubungan non-linear antara variabel input dan probabilitas kejadian, menjadikannya populer dalam berbagai aplikasi seperti prediksi risiko, deteksi spam, dan analisis diagnosis medis. Selain itu, regresi logistik juga dapat diperluas ke kasus multiklas melalui pendekatan seperti softmax regression.

7.2.1.3 Regresi Polynomial

Regresi polynomial adalah bentuk lanjutan dari regresi linear yang digunakan untuk memodelkan hubungan non-linear antara variabel input (independen) dan output (dependen) dengan menambahkan pangkat variabel input sebagai fitur

tambahan. Modelnya ditulis sebagai $Y=b0+b1X+b2X2+...+bnXn$, di mana n adalah derajat polinomial yang menentukan kompleksitas hubungan. Dengan menambahkan pangkat variabel, regresi polynomial mampu menangkap pola non-linear yang tidak dapat diwakili oleh regresi linear sederhana. Metode ini sering digunakan dalam kasus di mana data menunjukkan kurva atau perubahan kompleks, seperti memodelkan pertumbuhan populasi atau hubungan antara waktu belajar dan hasil ujian. Meskipun fleksibel, regresi polynomial memiliki risiko overfitting, terutama jika derajat polinomial terlalu tinggi, sehingga diperlukan evaluasi yang cermat untuk menemukan model yang seimbang antara akurasi dan generalisasi.

7.2.1.4 Regresi Ridge dan Lasso

Regresi Ridge dan Lasso adalah teknik regresi yang dirancang untuk mengatasi masalah overfitting pada model machine learning, khususnya ketika data memiliki banyak fitur atau terdapat korelasi antar fitur. Keduanya menggunakan pendekatan regularisasi untuk mengontrol besarnya koefisien regresi agar model tidak terlalu kompleks. Regresi Ridge menambahkan penalti berupa kuadrat dari koefisien ($L2$-regularization) ke fungsi kehilangan, sehingga dapat mengecilkan nilai koefisien tetapi tidak menghilangkannya sepenuhnya. Di sisi lain, Regresi Lasso menggunakan penalti berupa nilai absolut koefisien ($L1$-regularization), yang memiliki kemampuan untuk mengecilkan beberapa koefisien menjadi nol, sehingga efektif dalam seleksi fitur. Dengan menambahkan regularisasi ini, Ridge dan Lasso tidak hanya meningkatkan generalisasi model tetapi juga membantu mencegah model dari ketergantungan berlebihan terhadap fitur yang kurang relevan, menjadikannya pilihan yang populer dalam analisis data dengan dimensi tinggi.

7.2.1.5 Proses Regresi

Proses regresi dalam machine learning melibatkan beberapa langkah utama untuk membangun model yang mampu memprediksi nilai target.

1. Pra-pemrosesan Data
   - Normalisasi atau standarisasi data untuk meningkatkan kinerja model.
   - Penanganan data hilang dan deteksi outlier.
2. Pelatihan Model

   Model dilatih menggunakan data latih untuk mencari parameter optimal (misalnya $b0$ dan $b1$).

3. Evaluasi Model

   Metrik evaluasi:

   - Mean Absolute Error (MAE): Rata-rata kesalahan absolut antara prediksi dan data aktual.
   - Mean Squared Error (MSE): Rata-rata kesalahan kuadrat antara prediksi dan data aktual.
   - R-squared (R^2): Menilai seberapa baik model menjelaskan variabilitas data target.

   e. Prediksi

   Menggunakan model untuk memprediksi nilai target pada data baru.

7.2.1.6 Kelebihan dan Kelemahan Regresi

Regresi merupakan salah satu algoritma dasar machine learning yang memiliki sejumlah kelebihan dan kelemahan. Kelebihannya terletak pada kesederhanaannya, kemudahan implementasi, serta interpretasi yang jelas, menjadikannya ideal untuk eksplorasi awal data dan memahami hubungan antara variabel. Regresi juga efektif untuk memprediksi hubungan linear antara variabel input dan output. Namun, regresi memiliki kelemahan, terutama dalam menghadapi hubungan yang kompleks atau non-linear, di mana model regresi sederhana menjadi kurang akurat. Selain itu, regresi sensitif terhadap outlier yang dapat mengganggu hasil prediksi, dan cenderung mengalami overfitting jika jumlah fitur jauh lebih besar dibandingkan jumlah data. Meskipun demikian, penggunaan teknik regularisasi seperti Ridge dan Lasso dapat membantu mengatasi beberapa keterbatasan ini, meningkatkan performa dan kemampuan generalisasi model.

7.2.2 Decision Tree

Decision Tree adalah salah satu algoritma machine learning yang termasuk dalam kategori supervised learning dan digunakan untuk tugas klasifikasi maupun regresi. Algoritma ini bekerja dengan memecah data secara berulang ke dalam kelompok-kelompok yang lebih kecil berdasarkan kondisi tertentu, membentuk struktur pohon yang terdiri dari node, cabang, dan daun (Leo Breiman Jerome H. Friedman & Stone, 1984).

7.2.2.1 Komponen Decision Tree

Komponen utama dalam decision tree terdiri dari root node, decision node, leaf node, dan cabang (branch), yang bersama-sama membentuk struktur pohon untuk memecahkan data dan membuat prediksi.

1. Root Node

   Node awal yang mewakili seluruh dataset dan memulai proses pemecahan.
2. Decision Node

   Node yang membagi data lebih lanjut berdasarkan aturan tertentu.
3. Leaf Node

   Node akhir yang merepresentasikan prediksi atau hasil klasifikasi.
4. Branch

   Jalur yang menghubungkan node, menunjukkan kondisi pemisahan.

Setiap komponen ini bekerja secara sinergis untuk memetakan hubungan antara variabel input dan output, memungkinkan decision tree menjadi model yang mudah dipahami dan efektif.

7.2.2.2 Cara Kerja Decision Tree

Decision tree adalah algoritma machine learning yang bekerja dengan memecah data menjadi subset yang lebih kecil berdasarkan aturan tertentu hingga mencapai hasil prediksi atau klasifikasi. Struktur pohon ini terdiri dari node-node yang mewakili pengambilan keputusan, dengan cabang-cabang yang menghubungkan setiap kondisi untuk membentuk jalur hingga mencapai leaf node sebagai hasil akhir. Proses kerja decision tree sangat intuitif karena menyerupai cara manusia membuat keputusan, yaitu dengan memeriksa kondisi secara bertahap. Dengan pendekatan ini, decision tree dapat digunakan untuk berbagai aplikasi seperti klasifikasi email, analisis risiko, dan prediksi harga, menjadikannya salah satu algoritma yang paling fleksibel dan mudah dipahami.

1. Pemilihan Fitur

   Proses awal dalam decision tree dimulai dengan memilih fitur terbaik untuk memisahkan data pada setiap langkah. Pemilihan ini dilakukan menggunakan metrik seperti Information Gain, Entropy, atau Gini Index, yang menghitung seberapa baik suatu fitur dapat mengurangi ketidakpastian atau impuritas dalam data. Fitur dengan nilai terbaik akan dipilih sebagai dasar pemisahan pada node tertentu, memastikan bahwa pemisahan data menghasilkan subset yang lebih homogen.

2. Pemisahan Data

   Setelah fitur terbaik dipilih, data dipecah menjadi subset berdasarkan aturan yang ditentukan oleh nilai fitur tersebut. Misalnya, untuk fitur numerik, aturan dapat berupa apakah nilai lebih besar atau lebih kecil dari ambang batas tertentu, sedangkan untuk fitur kategori, data dapat dipecah berdasarkan kategori uniknya. Proses ini berlanjut secara rekursif untuk setiap subset, menciptakan cabang-cabang baru dalam pohon.

3. Pembangunan Cabang

   Cabang-cabang decision tree berkembang saat data terus dipecah berdasarkan fitur-fitur berikutnya. Pemisahan berlanjut hingga mencapai kondisi penghentian, seperti ketika semua data dalam subset berasal dari kelas yang sama, jumlah data dalam subset terlalu kecil, atau kedalaman pohon mencapai batas maksimum. Setiap pemisahan dirancang untuk meminimalkan ketidakpastian dan membuat data dalam subset semakin homogen.

4. Prediksi

   Ketika pohon selesai dibangun, data baru dapat digunakan untuk prediksi. Data tersebut akan mengikuti jalur pada pohon, mulai dari root node dan melewati cabang-cabang sesuai dengan aturan pada decision node. Proses ini berlanjut hingga mencapai leaf node, di mana prediksi atau klasifikasi dilakukan berdasarkan nilai yang ada pada node tersebut.

5. Penyempurnaan (Opsional)

   Untuk meningkatkan performa dan mengurangi risiko overfitting, teknik seperti pruning dapat digunakan. Pruning memotong cabang-cabang yang tidak memberikan kontribusi signifikan terhadap akurasi model, sehingga pohon menjadi lebih sederhana dan lebih general. Dengan proses ini, decision tree dapat memberikan prediksi yang lebih andal dan sesuai dengan data baru.

7.2.2.3 Kelemahan dan Kelebihan Decision Tree

Salah satu kelebihan utama decision tree adalah kemudahannya untuk diinterpretasikan dan dipahami. Struktur pohon yang menyerupai diagram alur memudahkan manusia untuk mengikuti logika pengambilan keputusan, bahkan oleh mereka yang tidak memiliki latar belakang teknis. Selain itu, decision tree fleksibel dalam menangani data numerik maupun kategori tanpa memerlukan pra-pemrosesan yang rumit, seperti normalisasi atau standarisasi. Algoritma ini juga non-parametrik, sehingga tidak bergantung pada asumsi distribusi data

tertentu. Decision tree sangat efektif untuk dataset kecil hingga menengah karena mampu membagi data secara cepat dan efisien berdasarkan aturan sederhana.

Meskipun memiliki banyak kelebihan, decision tree juga memiliki beberapa kelemahan. Salah satu masalah utamanya adalah risiko overfitting, terutama jika pohon terlalu dalam atau kompleks, sehingga model menjadi sangat spesifik terhadap data latih dan kehilangan kemampuan generalisasi pada data baru. Selain itu, algoritma ini sensitif terhadap perubahan data; sedikit perubahan pada dataset dapat menyebabkan perubahan signifikan pada struktur pohon. Decision tree juga cenderung kurang optimal untuk menangani dataset yang sangat besar atau data dengan hubungan non-linear yang kompleks, di mana algoritma ensemble seperti Random Forest atau Gradient Boosting lebih unggul.

7.2.3 Support Vector Machine

Support Vector Machine (SVM) adalah algoritma machine learning yang termasuk dalam kategori supervised learning dan digunakan untuk tugas klasifikasi maupun regresi. SVM dirancang untuk menemukan hyperplane terbaik yang dapat memisahkan dataset ke dalam kelas-kelas yang berbeda dengan margin maksimum. Algoritma ini sangat populer karena kemampuannya yang efektif dalam menangani data dengan dimensi tinggi dan hubungan non-linear (Müller & Guido, 2016).

SVM bekerja dengan mencari hyperplane, yaitu garis (untuk data dua dimensi) atau bidang (untuk data tiga dimensi) yang memisahkan data ke dalam dua kelas dengan jarak terbesar dari data terdekat di masing-masing kelas. Data yang paling dekat dengan hyperplane disebut support vectors, dan data ini menentukan posisi hyperplane.

7.2.3.1 Linier dan Non Linier Support Vector Machine (SVM)

Linear SVM adalah Digunakan jika data dapat dipisahkan dengan garis lurus. Hyperplane dalam linear SVM merupakan garis lurus atau bidang yang membagi data menjadi dua kelompok berbeda.

Non-Linear SVM adalah Untuk dataset yang tidak dapat dipisahkan secara linear, SVM menggunakan teknik kernel trick. Fungsi kernel memetakan data ke dimensi yang lebih tinggi di mana data dapat dipisahkan secara linear. Kernel yang umum digunakan meliputi:

- Linear Kernel: Untuk data yang linier.

- Polynomial Kernel: Cocok untuk hubungan non-linear yang kompleks.
- Radial Basis Function (RBF) Kernel: Umum digunakan untuk hubungan non-linear dengan data berdimensi tinggi.

7.2.3.2 Kelebihan dan Kelemahan Support Machine Learning

Salah satu kelebihan utama SVM adalah kemampuannya yang sangat baik dalam menangani data berdimensi tinggi. Hal ini membuat SVM efektif digunakan pada dataset dengan banyak fitur, bahkan ketika jumlah sampel relatif kecil. Selain itu, SVM menggunakan prinsip margin maksimum, yang membantu meningkatkan kemampuan generalisasi model sehingga dapat bekerja dengan baik pada data baru. Fleksibilitas SVM juga menjadi keunggulan, karena algoritma ini dapat menggunakan berbagai jenis fungsi kernel untuk memetakan data non-linear ke ruang dimensi yang lebih tinggi, di mana data tersebut dapat dipisahkan secara linear. Keunggulan lainnya adalah kemampuannya untuk menghindari overfitting pada dataset dengan dimensi tinggi, terutama jika parameter regulasi diatur dengan baik.

Namun, SVM memiliki beberapa kelemahan yang perlu diperhatikan. Salah satu kelemahan utamanya adalah kompleksitas komputasi yang tinggi, terutama ketika bekerja dengan dataset besar. Proses pelatihan SVM, yang melibatkan penghitungan semua pasangan data, membutuhkan banyak waktu dan memori. Selain itu, pemilihan parameter seperti C (regularisasi) dan parameter kernel sangat berpengaruh terhadap performa model, sehingga tuning parameter menjadi langkah yang krusial namun memakan waktu. SVM juga memiliki kelemahan dalam hal interpretabilitas, karena model ini tidak memberikan aturan keputusan yang mudah dipahami seperti decision tree. Hal ini membuat SVM kurang cocok untuk aplikasi di mana transparansi model menjadi prioritas.

7.2.4 Clustering

Clustering adalah salah satu teknik dalam unsupervised learning yang bertujuan untuk mengelompokkan data berdasarkan kesamaan atau karakteristik tertentu tanpa memerlukan label pada data. Metode ini digunakan untuk menemukan struktur tersembunyi dalam dataset, di mana data dengan karakteristik serupa akan dikelompokkan ke dalam satu cluster. Clustering banyak diterapkan dalam eksplorasi data, segmentasi pelanggan, analisis pola, dan berbagai aplikasi lainnya (Nelli, 2015)

7.2.4.1 Jenis-jenis Clustering

Clustering terdiri dari beberapa metode utama yang digunakan sesuai dengan karakteristik data:

1. Partitional Clustering: Data dipecah menjadi sejumlah cluster tertentu, biasanya dengan menggunakan algoritma seperti K-Means. K-Means bekerja dengan menginisialisasi centroid secara acak dan memperbarui posisi centroid hingga cluster stabil.
2. Hierarchical Clustering: Mengelompokkan data dalam struktur hierarkis berupa dendrogram. Proses ini dilakukan secara agglomerative (menggabungkan data dari bawah ke atas) atau divisive (membagi data dari atas ke bawah).
3. Density-Based Clustering: Algoritma seperti DBSCAN mengidentifikasi cluster berdasarkan kepadatan data, memungkinkan pengelompokan data yang memiliki bentuk tidak teratur.
4. Model-Based Clustering: Metode seperti Gaussian Mixture Models (GMM) mengasumsikan bahwa data berasal dari distribusi tertentu, biasanya Gaussian, untuk membentuk cluster.

7.2.4.2 Proses Clustering

Proses clustering dimulai dengan mempersiapkan data melalui pra-pemrosesan, seperti normalisasi dan penghapusan outlier, guna meningkatkan akurasi pengelompokan. Algoritma clustering kemudian diterapkan pada dataset untuk membentuk cluster berdasarkan metrik kesamaan tertentu, seperti jarak Euclidean atau Cosine Similarity. Setelah cluster terbentuk, hasilnya dapat divisualisasikan menggunakan diagram seperti scatter plot untuk analisis lebih lanjut. Proses ini diakhiri dengan evaluasi menggunakan metrik seperti Silhouette Score atau Dunn Index, meskipun evaluasi clustering seringkali lebih subjektif karena tidak ada label data.

7.2.4.3 Kelebihan dan kelemahan Clustering

Clustering unggul dalam kemampuannya untuk menemukan pola atau struktur tersembunyi dalam data tanpa memerlukan label. Teknik ini fleksibel dan dapat diterapkan pada berbagai jenis data, baik numerik maupun kategori. Selain itu, clustering memungkinkan eksplorasi data yang efektif, memberikan wawasan awal yang penting untuk analisis lebih lanjut atau pengambilan keputusan.

Namun, clustering memiliki beberapa kelemahan. Algoritma seperti K-Means sensitif terhadap inisialisasi centroid, sehingga dapat memberikan hasil yang berbeda pada setiap eksekusi. Selain itu, clustering cenderung kurang efektif untuk dataset dengan ukuran besar atau bentuk distribusi yang kompleks, kecuali algoritma yang lebih canggih digunakan. Evaluasi hasil clustering juga menantang karena tidak adanya ground truth dalam unsupervised learning.

7.2.4.4 Aplikasi Clustering

Clustering banyak digunakan dalam berbagai domain. Dalam bisnis, clustering digunakan untuk segmentasi pelanggan, di mana pelanggan dikelompokkan berdasarkan preferensi atau perilaku belanja. Di bidang biologi, clustering membantu mengidentifikasi gen dengan fungsi serupa. Selain itu, clustering juga digunakan dalam analisis citra, seperti pengelompokan piksel untuk segmentasi gambar, serta dalam sistem rekomendasi dan analisis sentimen.

7.3 Deep Learning dan Neural Network

Deep learning adalah cabang dari machine learning yang menggunakan arsitektur berbasis neural networks untuk mempelajari pola kompleks dalam data. Teknik ini dirancang untuk meniru cara kerja otak manusia dengan membangun jaringan saraf buatan yang mampu melakukan tugas-tugas seperti klasifikasi, prediksi, dan pengenalan pola secara otomatis. Deep learning unggul dalam menangani dataset besar dengan dimensi tinggi, menjadikannya sangat efektif untuk aplikasi seperti pengenalan gambar, pemrosesan bahasa alami, dan analisis data berurutan (Goodfellow et al., 2016).

7.3.1 Konsep Neural Network

Neural networks adalah struktur komputasi yang terdiri dari sejumlah besar unit kecil yang disebut neuron, yang diorganisasikan dalam lapisan-lapisan. Setiap neuron menerima input, mengalirkannya melalui fungsi aktivasi, dan menghasilkan output untuk diteruskan ke lapisan berikutnya. Neural networks biasanya terdiri dari tiga jenis lapisan utama: input layer, yang menerima data mentah; hidden layer, yang memproses data dengan mengidentifikasi pola-pola kompleks; dan output layer, yang menghasilkan prediksi akhir. Proses pelatihan neural networks menggunakan algoritma backpropagation untuk menyesuaikan bobot dan bias setiap neuron berdasarkan kesalahan prediksi, sehingga model dapat belajar dari data (Fairuzabadi et al., 2025).

7.3.1.1 Convolutional Neural Netwok dan Recurrent Neural Network

Convolutional Neural Networks (CNN) dirancang khusus untuk memproses data dalam bentuk grid, seperti gambar. CNN menggunakan operasi konvolusi untuk mengekstrak fitur lokal dari gambar, seperti tepi, tekstur, atau objek tertentu. Lapisan-lapisan konvolusi ini diikuti oleh pooling layer untuk mengurangi dimensi data, yang membantu mengurangi kompleksitas komputasi tanpa kehilangan informasi penting. CNN banyak digunakan dalam aplikasi seperti deteksi objek, klasifikasi gambar, dan segmentasi gambar.

Sebaliknya, Recurrent Neural Networks (RNN) dirancang untuk menangani data berurutan, seperti teks, audio, atau data sensor. RNN memiliki arsitektur unik dengan loop dalam jaringannya, memungkinkan informasi dari waktu sebelumnya untuk memengaruhi prediksi waktu saat ini. Namun, RNN sering menghadapi masalah seperti vanishing gradient, yang membuatnya sulit untuk mempelajari hubungan jangka panjang. Untuk mengatasi hal ini, varian seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) dikembangkan, yang memungkinkan model untuk menangkap hubungan temporal yang lebih kompleks.

7.3.2 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) adalah arsitektur deep learning yang terdiri dari dua jaringan neural yang saling bersaing: generator dan discriminator. Generator bertugas menciptakan data palsu yang menyerupai data asli, sementara discriminator bertugas membedakan antara data asli dan data yang dihasilkan oleh generator. Proses pelatihan GAN bersifat adversarial, di mana generator berusaha menghasilkan data yang semakin realistis untuk "mengelabui" discriminator. GAN telah menjadi alat yang sangat kuat dalam menghasilkan data sintetis, seperti gambar realistis, peningkatan resolusi gambar (super-resolution), hingga menghasilkan karya seni atau video animasi.

7.4 Model dan Ensemble Learning

Optimasi model dan ensemble learning adalah dua pendekatan penting dalam machine learning yang bertujuan untuk meningkatkan akurasi dan kemampuan generalisasi model. Optimasi model dilakukan dengan menyempurnakan parameter dan struktur model, sementara ensemble learning menggabungkan beberapa model untuk menghasilkan prediksi yang lebih baik dibandingkan

dengan model tunggal. Kedua pendekatan ini berkontribusi signifikan dalam menangani tantangan seperti overfitting, underfitting, dan keandalan prediksi (Cao, 2018)

7.4.1 Hyperparameter Tuning dan Cross-Validation

Hyperparameter tuning adalah proses untuk mencari kombinasi parameter optimal yang mengontrol cara kerja algoritma machine learning, seperti jumlah pohon pada Random Forest atau tingkat regularisasi pada SVM. Parameter ini tidak dipelajari langsung dari data, tetapi ditentukan sebelum pelatihan model dimulai. Dua pendekatan populer untuk hyperparameter tuning adalah Grid Search dan Random Search. Grid Search mengevaluasi semua kombinasi parameter dalam ruang pencarian, sedangkan Random Search memilih kombinasi parameter secara acak, yang lebih efisien untuk ruang pencarian besar.

Untuk memastikan keakuratan model selama proses tuning, digunakan cross-validation, yang membagi dataset menjadi beberapa subset (folds). Model dilatih pada sebagian data dan diuji pada subset yang berbeda secara bergantian. Teknik ini membantu menghindari overfitting dengan memberikan gambaran yang lebih akurat tentang kinerja model pada data baru. Kombinasi hyperparameter yang memberikan hasil terbaik pada cross-validation kemudian dipilih untuk model akhir.

7.4.2 Teknik Ensemble: Bagging, Boosting dan Stacking

Ensemble learning adalah teknik yang menggabungkan beberapa model untuk meningkatkan performa prediksi. Tiga pendekatan ensemble utama adalah bagging, boosting, dan stacking.

1. Bagging (Bootstrap Aggregating)

   Bagging menciptakan beberapa model dari subset data yang diambil secara acak dengan pengembalian (bootstrapping). Setiap model dilatih secara independen, dan hasil akhirnya diperoleh dengan rata-rata (untuk regresi) atau voting (untuk klasifikasi). Contoh paling populer dari teknik ini adalah Random Forest, yang menggunakan beberapa pohon keputusan untuk meningkatkan stabilitas dan akurasi prediksi.

2. Boosting

   Boosting bekerja dengan melatih model secara berurutan, di mana setiap model baru berfokus pada memperbaiki kesalahan model sebelumnya. Model akhir adalah kombinasi dari semua model dengan bobot yang lebih

besar diberikan pada model yang lebih baik. Algoritma seperti AdaBoost, Gradient Boosting, dan XGBoost adalah contoh terkenal yang menggunakan pendekatan ini untuk meningkatkan akurasi, terutama pada data dengan pola kompleks.

3. Stacking

   Stacking menggabungkan beberapa model dari jenis yang berbeda dengan menggunakan model meta (meta-learner). Hasil prediksi dari model pertama digunakan sebagai input untuk model meta, yang bertugas mempelajari pola dari hasil tersebut dan membuat prediksi akhir. Teknik ini memungkinkan penggabungan keunggulan dari berbagai jenis model, sehingga menghasilkan performa yang lebih baik dibandingkan model individu.

Optimasi model dengan hyperparameter tuning dan cross-validation memastikan model bekerja secara optimal, sementara teknik ensemble seperti bagging, boosting, dan stacking menawarkan pendekatan cerdas untuk meningkatkan akurasi prediksi. Dengan memanfaatkan kedua pendekatan ini, model machine learning menjadi lebih andal dan mampu mengatasi tantangan yang muncul dalam berbagai jenis data.

7.5 Preprocessing Data dan Evaluasi Data

7.5.1 Preprocessing Data

Preprocessing data adalah langkah awal yang krusial dalam machine learning untuk memastikan bahwa data siap digunakan oleh model dan dapat menghasilkan hasil yang akurat. Salah satu teknik penting dalam preprocessing adalah normalisasi, yaitu mengubah skala fitur agar berada dalam rentang tertentu, seperti [0,1], untuk menghindari dominasi fitur dengan nilai besar terhadap model (Larose & Larose, 2014). Teknik ini sangat penting untuk algoritma seperti regresi atau K-Nearest Neighbors, yang sensitif terhadap skala fitur. Selain itu, encoding digunakan untuk menangani data kategori, seperti mengonversi teks menjadi nilai numerik yang dapat dipahami oleh model. Dua metode umum dalam encoding adalah label encoding, yang mengganti kategori dengan angka, dan one-hot encoding, yang menciptakan vektor biner untuk setiap kategori. Penanganan data hilang juga menjadi bagian penting dalam preprocessing, di mana nilai yang hilang dapat diisi dengan rata-rata (mean), median, atau nilai lainnya, tergantung pada konteks data.

7.5.2 Evaluasi Data

Setelah preprocessing selesai dan model machine learning diterapkan, langkah berikutnya adalah mengevaluasi performa model. Dalam klasifikasi, metrik evaluasi yang sering digunakan adalah precision, recall, dan F1-score. Precision mengukur seberapa akurat prediksi positif model, yaitu proporsi prediksi positif yang benar terhadap semua prediksi positif $\left(Precision = \frac{True\ Positive}{True\ Posistive+False\ Positive}\right)$. Recall mengukur kemampuan model untuk menemukan semua kasus positif sebenarnya dalam dataset $\left(Recall = \frac{True\ Positive}{True\ Positive+False\ Negative}\right)$. Sementara itu, F1-score adalah rata-rata harmonis dari precision dan recall, memberikan keseimbangan antara keduanya, terutama ketika terdapat ketidakseimbangan antara data positif dan negative $\left(F1 - Score = 2.\frac{Precision.Recall}{Precision+Recall}\right)$.

7.5.3 Pentingnya Preprocessing dan Evaluasi

Preprocessing data yang baik memastikan bahwa data dalam kondisi optimal untuk model belajar, sementara evaluasi yang tepat membantu menentukan apakah model bekerja dengan baik pada data baru. Normalisasi, encoding, dan penanganan data hilang membantu mengurangi bias dan memastikan keadilan dalam pelatihan model, sedangkan metrik seperti precision, recall, dan F1-score memberikan gambaran yang lebih detail tentang kinerja model dalam menangani data dengan distribusi yang tidak seimbang. Dengan langkah-langkah ini, proses machine learning menjadi lebih andal dan memberikan hasil yang akurat serta relevan.

7.6 Implementasi dan Studi Kasus Dalam Machine Learning

Machine learning telah menjadi alat penting dalam berbagai industri karena kemampuannya untuk mempelajari pola dari data dan membuat prediksi yang relevan. Implementasi machine learning tidak hanya berhenti pada pembangunan model, tetapi melibatkan serangkaian langkah yang dikenal sebagai workflow machine learning, mulai dari pengumpulan data hingga deployment model ke lingkungan produksi. Selain itu, aplikasi nyata dari machine learning dapat ditemukan di berbagai domain, seperti bisnis, kesehatan, teknologi, dan pendidikan, yang menunjukkan fleksibilitas dan dampaknya yang luas.

7.6.1 Workflow Machine Learning

Proses workflow machine learning terdiri dari beberapa tahap utama yang saling terhubung. Tahap pertama adalah pengumpulan data, di mana data dikumpulkan dari berbagai sumber, seperti basis data, API, atau sensor, dan diikuti oleh pra-pemrosesan data untuk membersihkan dan menyiapkan data agar cocok untuk analisis. Setelah itu, data dibagi menjadi data latih dan data uji, di mana model dilatih menggunakan data latih untuk mempelajari pola yang ada.

Tahap selanjutnya adalah evaluasi model, di mana kinerja model diuji menggunakan data uji dengan metrik tertentu, seperti akurasi, F1-score, atau AUC-ROC, untuk memastikan bahwa model dapat bekerja secara andal. Jika hasil evaluasi memadai, model kemudian dioptimalkan melalui teknik seperti hyperparameter tuning untuk meningkatkan performanya. Tahap akhir adalah deployment, di mana model diintegrasikan ke dalam sistem produksi, seperti aplikasi web atau API, sehingga dapat digunakan secara langsung oleh pengguna akhir. Siklus ini sering diulang secara iteratif untuk memperbarui model dengan data baru dan meningkatkan akurasi dari waktu ke waktu.

7.6.2 Contoh Implementasi di Berbagai Domain

1. Bisnis dan E-commerce

   Machine learning banyak digunakan dalam segmentasi pelanggan, sistem rekomendasi, dan analisis perilaku belanja. Misalnya, algoritma clustering membantu e-commerce seperti Amazon untuk mengelompokkan pelanggan berdasarkan preferensi belanja mereka, sehingga dapat memberikan rekomendasi produk yang lebih relevan.

2. Kesehatan

   Di bidang kesehatan, machine learning digunakan untuk analisis citra medis, seperti mendeteksi kanker melalui citra MRI, serta untuk memprediksi hasil klinis pasien berdasarkan data medis elektronik. Model berbasis machine learning juga digunakan dalam diagnosis penyakit dan pengembangan obat baru.

3. Teknologi dan Keamanan Siber

   Dalam dunia teknologi, machine learning digunakan untuk deteksi intrusi dan pencegahan serangan siber melalui analisis pola jaringan yang mencurigakan. Di perusahaan seperti Google dan Facebook, machine learning diterapkan untuk meningkatkan pengalaman pengguna melalui pengenalan wajah, prediksi teks, dan personalisasi iklan.

4. Pendidikan

   Machine learning membantu menciptakan sistem pembelajaran adaptif yang dapat menyesuaikan konten pembelajaran dengan kebutuhan individu siswa. Selain itu, analisis data pendidikan memungkinkan institusi untuk mengidentifikasi siswa yang memerlukan intervensi lebih awal.

5. Transportasi dan Logistik

   Dalam transportasi, machine learning mendukung pengembangan mobil otonom yang mampu memahami lingkungan di sekitarnya dan membuat keputusan real-time. Di bidang logistik, algoritma optimasi digunakan untuk merencanakan rute pengiriman yang efisien, mengurangi biaya operasional.

7.7 Tren Terbaru Dalam Machine Learning

7.7.1 AutoML dan Integrasi dengan IoT

AutoML (Automated Machine Learning) adalah tren terbaru yang bertujuan menyederhanakan proses pengembangan model machine learning dengan mengotomatiskan langkah-langkah seperti pra-pemrosesan data, pemilihan fitur, pemilihan algoritma, dan tuning hyperparameter. AutoML memungkinkan pengguna dengan sedikit atau tanpa pengalaman dalam machine learning untuk membangun model yang optimal secara efisien. Ketika digabungkan dengan Internet of Things (IoT), AutoML membuka peluang baru untuk analisis data real-time yang dihasilkan oleh perangkat IoT. Data besar dari perangkat IoT, seperti sensor dalam rumah pintar atau perangkat medis, dapat diproses menggunakan pipeline AutoML untuk menghasilkan insight secara otomatis. Tren ini mendukung pengambilan keputusan yang lebih cepat dan lebih cerdas dalam berbagai domain, seperti manajemen energi, pengawasan lingkungan, dan kesehatan.

7.7.2 Machine Learning di Edge Computing

Edge computing adalah pendekatan komputasi yang memproses data di dekat sumbernya, seperti perangkat IoT atau node jaringan, alih-alih mengirimkan data ke cloud untuk diproses. Integrasi machine learning dengan edge computing menjadi tren penting untuk menangani tantangan seperti latensi, privasi data, dan efisiensi bandwidth. Model machine learning yang di-deploy di perangkat edge, seperti microcontroller atau gateway IoT, memungkinkan analisis real-time tanpa perlu mengandalkan koneksi internet. Misalnya,

kendaraan otonom menggunakan machine learning di edge untuk menganalisis lingkungan mereka dalam hitungan milidetik. Tren ini juga relevan dalam aplikasi industri, seperti prediksi kegagalan mesin atau deteksi anomali, di mana waktu respons yang cepat sangat penting.

Bab 8
Pemodelan dan Evaluasi Data pada data science

8.1 Teknik Pemodelan Data Science

Data Science adalah bidang multidisiplin yang mengintegrasikan berbagai metode ilmiah, proses, algoritme, dan sistem untuk mengekstrak pengetahuan dan wawasan dari data terstruktur dan tidak terstruktur (Govindarajan, 2020; Mukherjee & Srinivasa Rao, 2022). Data science menggabungkan keahlian dari berbagai bidang seperti matematika, statistik, ilmu komputer, dan pengetahuan spesifik domain untuk menganalisis dan menginterpretasikan kumpulan data yang besar (Al-Haija, 2022; Leung, 2021; Lu, 2022). Tujuan utama dari data science adalah untuk mengubah data mentah menjadi informasi yang bermakna yang dapat mendukung proses pengambilan keputusan di berbagai industri, termasuk perawatan kesehatan, keuangan, dan teknologi (Ismail et al., 2024).

Pemodelan data adalah komponen penting dari data science, yang melibatkan pembuatan representasi abstrak dari entitas dunia nyata dan hubungan mereka dalam database (Smalter et al., 2008). Hal ini berfungsi sebagai fondasi untuk merancang sebuah sistem, dan sangat penting untuk memastikan keakuratan dan konsistensi data. Teknik pemodelan data telah berkembang seiring dengan kemajuan teknologi, namun prinsip-prinsip intinya tetap berakar pada metode tradisional (Jaakkola & Thalheim, 2020).

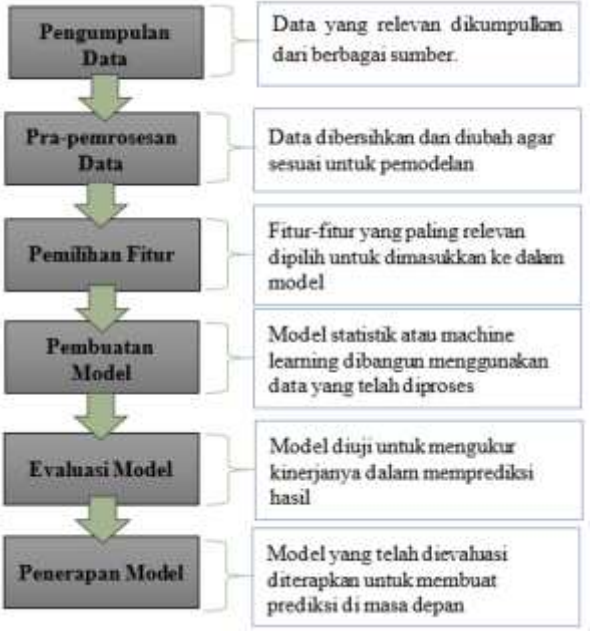
Tantangan dalam pemodelan data adalah memastikan keakuratan model matematika yang digunakan untuk mengestimasi nilai parameter. Hal ini melibatkan validasi turunan analitik dan membandingkannya dengan gradien numerik untuk memastikan ketepatannya (Rahayu et al., 2024). Selain itu, meningkatnya kompleksitas dan volume data, yang sering disebut sebagai big data, menghadirkan tantangan yang signifikan dalam hal manajemen, pemrosesan, dan analisis data (Ismail et al., 2024; Marino et al., 2018).

Dalam dunia data science, pemodelan dan evaluasi data adalah dua aspek krusial yang menentukan keberhasilan suatu proyek analitik. Pemodelan data berfokus pada pembuatan algoritma untuk mempelajari pola dalam data, sementara evaluasi model menilai seberapa baik model tersebut dapat

memprediksi hasil yang diinginkan. Proses ini juga melibatkan optimasi model untuk meningkatkan kinerja dan ketepatan prediksi. Pemodelan data dimulai dengan membuat representasi matematis atau statistik dari fenomena yang kita ingin analisis. Pemodelan data memungkinkan analis untuk mengidentifikasi hubungan, membuat prediksi, serta memahami, menginterpretasikan, dan memvisualisasikan data secara lebih strategis (Dimotikalis et al., 2021a, 2021b). Ada berbagai teknik yang digunakan dalam pemodelan data, diantaranya :

8.1.1 Predictive Modeling

Predictive Modeling melibatkan penggunaan teknik statistik untuk memprediksi hasil di masa depan berdasarkan data historis. Model ini dapat digunakan untuk berbagai tujuan, seperti memprediksi penjualan, mengidentifikasi risiko, atau memahami perilaku pelanggan. Predictive modeling melibatkan beberapa tahapan yang dapat dilihat pada gambar berikut:



Gambar 8.1: Tahapan Predictive Modeling

Ada berbagai teknik predictive modeling yang tersedia, antara lain:

1. Analisis Regresi: Digunakan untuk memprediksi variabel hasil kontinu berdasarkan satu atau lebih variabel prediktor (Al-Haija, 2022; Antonio Nuno and de Almeida, 2022). Analisis regresi linier digunakan untuk memprediksi variabel kontinu berdasarkan hubungan linier dengan variabel lain, sedangkan regresi logistik digunakan untuk memprediksi variabel kategorikal, misalnya, "ya" atau "tidak" berdasarkan variabel lain.
2. *Classification Algorithms:* Seperti pohon keputusan, klasifikasi Bayesian, dan jaringan saraf, yang digunakan untuk memprediksi hasil kategorikal. (Gupta et al., 2024)
3. Machine Learning : Teknik seperti mesin vektor pendukung dan algoritme genetik untuk meningkatkan akurasi model (Gupta et al., 2024; Smalter et al., 2008). Model ini terinspirasi oleh cara kerja otak manusia, digunakan untuk memecahkan masalah yang kompleks. Teknik-teknik ini digunakan untuk membuat model prediktif dan mengotomatiskan proses analisis data (Jaakkola & Thalheim, 2020; Kshatri et al., 2022; Mukherjee & Srinivasa Rao, 2022).

Predictive modeling menawarkan berbagai manfaat diantaranya:

1. Pengambilan Keputusan yang Lebih Baik: Prediksi yang akurat dapat membantu dalam pengambilan keputusan yang lebih tepat.
2. Peningkatan Efisiensi: Model dapat digunakan untuk mengotomatiskan tugas-tugas rutin dan meningkatkan efisiensi operasional.
3. Peningkatan Kepuasan Pelanggan: Dengan memahami perilaku pelanggan, perusahaan dapat memberikan layanan yang lebih personal dan meningkatkan kepuasan pelanggan.
4. Peningkatan Keuntungan: Dengan memprediksi penjualan dan mengidentifikasi peluang baru, perusahaan dapat meningkatkan keuntungan.

Meskipun menawarkan banyak manfaat, predictive modeling juga memiliki tantangan, antara lain:

1. Kualitas Data: Kualitas data yang buruk dapat menghasilkan model yang tidak akurat.
2. Kompleksitas Model: Model yang terlalu kompleks dapat sulit diinterpretasikan dan diimplementasikan.

3. Perubahan Lingkungan: Model perlu diperbarui secara berkala untuk menyesuaikan diri dengan perubahan lingkungan.

8.1.2 Descriptive Modeling

Descriptive modeling adalah teknik untuk meringkas dan menyajikan data historis yang bertujuan untuk mengidentifikasi pola, hubungan, dan tren yang tersembunyi di dalam data tersebut. Model deskriptif membantu memahami apa yang telah terjadi di masa lalu dan memberikan wawasan berharga untuk pengambilan keputusan di masa depan. Pemodelan deskriptif bertujuan untuk menggambarkan pola dan hubungan dalam data. Beberapa teknik umum yang digunakan dalam *descriptive modeling* antara lain

- Clustering: Mengelompokkan data ke dalam kelompok-kelompok berdasarkan kesamaan karakteristik. Metode statistik tradisional digabungkan dengan algoritme modern untuk meningkatkan interpretasi data dan akurasi model (Kshatri et al., 2022). Contohnya, pengelompokan pelanggan berdasarkan perilaku pembelian mereka.
- Segmentasi: Membagi data menjadi segmen-segmen yang lebih kecil berdasarkan kriteria tertentu. Contohnya, segmentasi pasar berdasarkan demografi dan preferensi pelanggan.
- Analisis Asosiasi: Mengidentifikasi hubungan antar variabel dalam data. Contohnya, menemukan produk-produk yang sering dibeli bersamaan.
- Analisis Deskriptif Sederhana: Metode ini menggunakan perhitungan statistik sederhana untuk menggambarkan data, seperti mean, median, modus, dan standar deviasi.
- Visualisasi Data: Merepresentasikan data dalam bentuk visual seperti grafik dan diagram untuk memudahkan pemahaman dan interpretasi.
- Cluster Analysis: Mengelompokkan titik data ke dalam klaster berdasarkan kemiripan (Al-Haija, 2022).
- Principal Component Analysis (PCA) : Mengurangi dimensi data dengan tetap mempertahankan sebagian besar varians. (Al-Haija, 2022)

Beberapa manfaat *Descriptive Modeling* diantaranya :

- Memahami insight data historis: Membantu memahami apa yang telah terjadi dan mengapa hal itu terjadi.

- Identifikasi peluang: Membantu mengidentifikasi peluang-peluang baru berdasarkan pola dan tren yang ditemukan dalam data.
- Pengambilan keputusan yang lebih baik: Memberikan wawasan yang lebih baik untuk pengambilan keputusan yang lebih tepat dan efektif.
- Peningkatan efisiensi: Membantu meningkatkan efisiensi operasional dengan memahami proses bisnis yang ada.

8.1.3 Data mining

Arah Masa Depan Masa depan pemodelan data dalam ilmu data diperkirakan akan sangat dipengaruhi oleh kemajuan dalam kecerdasan buatan (AI) dan pembelajaran mesin (ML). Teknologi ini tidak hanya mengotomatiskan tugas-tugas rutin tetapi juga meningkatkan kemampuan adaptasi dan kecerdasan model data. Selain itu, pengembangan kerangka kerja yang dapat diskalakan untuk analitik data besar akan sangat penting untuk mengelola volume dan kompleksitas data yang semakin meningkat (Jaakkola & Thalheim, 2020; Marino et al., 2018).

Pemodelan data adalah aspek fundamental dari ilmu data, yang memungkinkan transformasi data mentah menjadi wawasan yang dapat ditindaklanjuti. Terlepas dari tantangan yang ditimbulkan oleh big data, kemajuan dalam AI, ML, dan alat visualisasi data terus mendorong inovasi di bidang ini. Seiring dengan berkembangnya ilmu data, integrasi teknologi-teknologi ini akan menjadi sangat penting untuk mengembangkan model data yang lebih akurat dan efisien (Leung, 2021; Mukherjee & Srinivasa Rao, 2022).

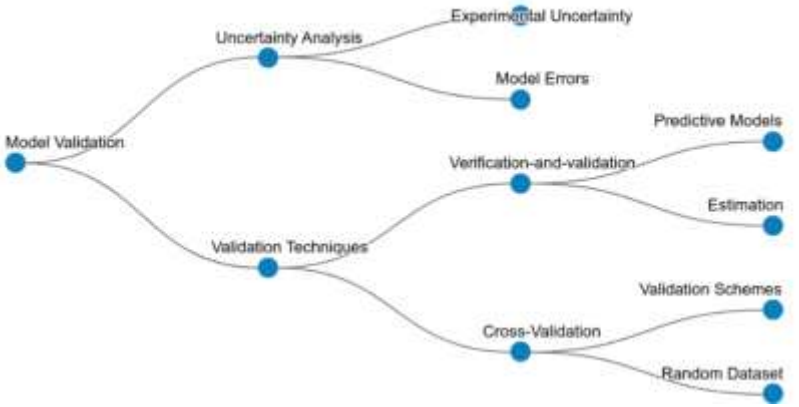
Data mining melibatkan penemuan pola dan pengetahuan dari kumpulan data yang besar dengan menggunakan metode seperti:

- Association Rule Learning: Mengidentifikasi hubungan yang menarik di antara variabel-variabel dalam basis data yang besar. (Antonio Nunoand de Almeida, 2022).
- Anomaly Detection : Mengidentifikasi titik data yang tidak biasa yang tidak sesuai dengan pola umum. (Al-Haija, 2022; Gupta et al., 2024)
- Regresi linier digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen (fitur) dan variabel dependen (target) dalam bentuk hubungan linear. Ini adalah metode yang sederhana dan sangat berguna dalam prediksi nilai kontinu, seperti memprediksi harga rumah berdasarkan fitur-fitur tertentu.

8.2 Validasi Model dan Cross-Validation

8.2.1 Validasi Model

Validasi model adalah proses menilai seberapa baik hasil analisis statistik akan digeneralisasi ke kumpulan data independen. Hal ini sangat penting untuk memastikan bahwa sebuah model berkinerja baik tidak hanya pada data pelatihan tetapi juga pada data yang tidak terlihat (Seraj et al., 2022). Tujuan utamanya adalah untuk mengukur akurasi dan keandalan model, yang dapat melibatkan identifikasi dan pengurangan ketidakpastian dan bias dalam model. Secara bagan dapat dilihat pada gambar berikut.



Gambar 8.2: Bagan Validasi model

Validasi model adalah langkah penting dalam ilmu data untuk memastikan bahwa prediksi model akurat dan dapat digeneralisasi ke data baru yang belum pernah ada sebelumnya. Validasi model melibatkan penilaian ketepatan model relatif terhadap data eksperimental dan dapat membantu mengukur ketidakpastian model atau meningkatkan model melalui kalibrasi. Proses ini diperumit oleh ketidakpastian dalam hasil simulasi dan eksperimen, yang dapat bersifat acak, karena kurangnya pengetahuan, atau bias (Voyles & Roy, 2014).

Dalam *Data Science*, membangun model yang akurat dan dapat diandalkan adalah tujuan utama. Validasi model adalah proses untuk mengevaluasi kinerja model dan memastikan bahwa model tersebut dapat digeneralisasi dengan baik pada data yang belum dilihat sebelumnya. Tujuannya adalah untuk menghindari *overfitting*, yaitu kondisi di mana model terlalu kompleks dan hanya bekerja

dengan baik pada data pelatihan, tetapi buruk pada data baru. Ada beberapa metode validasi model yang umum digunakan, antara lain:

1. *Hold-out Validation*: Metode ini membagi dataset menjadi dua bagian, yaitu data pelatihan dan data pengujian. Model dilatih pada data pelatihan, kemudian diuji pada data pengujian untuk mengukur kinerjanya.
2. *K-Fold Cross-Validation*: Metode ini membagi dataset menjadi *k* bagian yang sama besar. Model dilatih dan diuji sebanyak *k* kali, dengan setiap bagian digunakan sebagai data pengujian sekali dan sisanya sebagai data pelatihan. Hasil kinerja dari setiap pengujian kemudian dirata-ratakan untuk mendapatkan estimasi kinerja model yang lebih robust (Seraj et al., 2022; Verma et al., 2024).
3. *Stratified K-Fold Cross-Validation*: Metode ini mirip dengan *K-Fold Cross-Validation*, tetapi memastikan bahwa proporsi kelas dalam setiap bagian data pengujian sama dengan proporsi kelas dalam dataset asli. Metode ini sangat berguna jika dataset memiliki ketidakseimbangan kelas.
4. *Leave-One-Out Cross-Validation*: Metode ini adalah kasus khusus dari *K-Fold Cross-Validation* di mana *k* sama dengan jumlah total data. Setiap data poin digunakan sebagai data pengujian sekali dan sisanya sebagai data pelatihan (Larracy et al., 2021).

8.2.2 Validasi Silang (Cross-Validation)

Validasi silang (*Cross-Validation*) adalah teknik yang banyak digunakan untuk validasi model dalam ilmu data. Teknik ini melibatkan partisi data ke dalam subset, melatih model pada beberapa subset (himpunan pelatihan), dan memvalidasinya pada subset yang tersisa (himpunan validasi) (Seraj et al., 2022). Metode ini membantu dalam memperkirakan kesalahan prediksi yang sebenarnya dari model dan menyetel parameter model (Arthi et al., 2023).

Berikut merupakan jenis-jenis Cross Validation

1. Validasi Silang Lengkap *(Exhaustive Cross-Validation)*:
    a. *Leave-P-Out Cross-Validation* : Menggunakan semua cara yang memungkinkan untuk membagi sampel asli ke dalam set pelatihan dan validasi.
    b. *Leave-One-Out Cross-Validation*: Setiap pengamatan digunakan sekali sebagai set validasi sementara pengamatan yang tersisa membentuk set pelatihan

2. Non-Exhaustive Cross-Validation:
    a. Hold-Out Method : Membagi data ke dalam satu set pelatihan dan satu set validasi.
    b. K-Fold Cross-Validation: Membagi data menjadi 'k' subset. Model dilatih pada 'k-1' subset dan divalidasi pada subset yang tersisa. Proses ini diulang sebanyak 'k' kali, dengan setiap subset digunakan tepat satu kali sebagai data validasi (Seraj et al., 2022; Verma et al., 2024).
    c. Nested Cross-Validation Digunakan untuk penyetelan hiperparameter dan pemilihan model, memberikan estimasi yang tidak bias terhadap kinerja model (Larracy et al., 2021).

Cross-validation adalah teknik validasi model yang sangat penting karena memiliki beberapa keunggulan:

- Estimasi Kinerja yang Lebih Akurat: Cross-validation memberikan estimasi kinerja model yang lebih akurat dibandingkan dengan metode hold-out validation, karena menggunakan seluruh dataset untuk pelatihan dan pengujian. Memberikan estimasi yang lebih akurat dari kinerja model dibandingkan dengan train-test split sederhana, terutama ketika data terbatas (Arthi et al., 2023; Rafało, 2022; Seraj et al., 2022).
- Mengurangi Risiko Overfitting: Dengan menggunakan cross-validation, kita dapat mengidentifikasi apakah model mengalami overfitting atau tidak. Jika kinerja model pada data pengujian jauh lebih buruk daripada kinerja pada data pelatihan, maka kemungkinan model mengalami overfitting.
- Memilih Model Terbaik: Cross-validation memungkinkan kita untuk membandingkan kinerja beberapa model dan memilih model terbaik yang memiliki kinerja generalisasi yang terbaik (Shen et al., 2011). Cross validation juga membantu menyetel parameter model untuk mencapai performa terbaik (Gholamiangonabadi et al., 2020; Hylton et al., 2022).

8.3 Metrik Evaluasi Model

Dalam data science, membangun model hanyalah langkah awal. Selanjutnya yang tidak kalah penting adalah memastikan model tersebut bekerja dengan baik dan sesuai dengan tujuan yang diinginkan. Berdasarkan hal tersebut, kita membutuhkan metrik evaluasi model. Metrik evaluasi adalah alat penting dalam ilmu data untuk menilai kinerja model prediktif. Metrik ini memberikan ukuran kuantitatif tentang kinerja model, membantu kita memahami seberapa baik

model tersebut dalam memprediksi atau mengklasifikasikan data. Ada berbagai macam metrik evaluasi model, dan pemilihan metrik yang tepat tergantung pada jenis masalah yang dihadapi (klasifikasi atau regresi) dan tujuan dari model tersebut.

8.3.1 Metrik evaluasi untuk Klasifikasi

Matriks klasifikasi, atau confusion matrix, adalah alat penting dalam evaluasi kinerja model klasifikasi dalam data sains. Bentuknya seperti tabel yang memungkinkan kita untuk memvisualisasikan dan mengukur seberapa baik model klasifikasi bekerja dalam memprediksi kelas data. Matriks klasifikasi memberikan informasi penting tentang kinerja model klasifikasi. Dari matriks klasifikasi dapat dipergunakan untuk menghitung berbagai metrik evaluasi, diantaranya:

1. Akurasi: Mengukut proporsi prediksi yang benar dari total prediksi.
    - Kelebihan: Mudah diinterpretasikan.
    - Kekurangan: Tidak cocok untuk data yang tidak seimbang (jumlah data kelas berbeda jauh).
2. Presisi: Rasio prediksi positif yang benar terhadap total prediksi positif.
    - Kelebihan: Penting jika False Positive (prediksi positif padahal negatif) lebih merugikan.
    - Kekurangan: Tidak memperhitungkan False Negative (prediksi negatif padahal positif).
3. Penarikan kembali (Recall): Rasio prediksi data positif yang benar terhadap total positif yang sebenarnya, atau dengan kata lain proporsi data positif yang berhasil diprediksi dengan benar.
    - Kelebihan: Penting jika False Negative lebih merugikan.
    - Kekurangan: Tidak memperhitungkan False Positive.
4. Skor F1: Rata-rata harmonis dari presisi dan recall, yang memberikan keseimbangan di antara keduanya.
    - Kelebihan: Menyeimbangkan presisi dan recall.
    - Kekurangan: Tidak sejelas presisi atau recall secara terpisah

5. Area Under the ROC Curve (AUC-ROC): Mengevaluasi kemampuan model untuk membedakan antara kelas-kelas pada nilai ambang batas yang berbeda .
    - Kelebihan: Cocok untuk data yang tidak seimbang.
    - Kekurangan: Agak sulit diinterpretasikan (Anitha et al., 2024; C. Liu, 2024).

8.3.2 Metrik evaluasi untuk Regresi:

Dalam dunia data sains, regresi adalah salah satu teknik pemodelan yang paling umum digunakan. Tujuannya adalah untuk memprediksi nilai numerik berdasarkan data sebelumnya. Setelah membangun model regresi, langkah penting selanjutnya adalah mengevaluasi kinerjanya. Untuk melakukan ini, kita memerlukan metrik evaluasi yang tepat. Metrik ini akan membantu kita memahami seberapa baik model dalam memprediksi nilai sebenarnya. Metrik evaluasi yang umum digunakan untuk model regresi diantaranya:

1. MAE (Mean Absolute Error): Rata-rata selisih absolut antara prediksi dan nilai sebenarnya. Kelebihan: Mudah diinterpretasikan. Kekurangan: Sensitif terhadap outlier.
2. MSE (Mean Squared Error): Rata-rata kuadrat selisih antara prediksi dan nilai sebenarnya. Kelebihan: Lebih sensitif terhadap perbedaan besar. Kekurangan: Sulit diinterpretasikan karena dalam satuan kuadrat.
3. RMSE (Root Mean Squared Error): Akar kuadrat dari MSE. Kelebihan: Lebih mudah diinterpretasikan daripada MSE. Kekurangan: Tetap sensitif terhadap outlier.
4. R-squared (R^2): Menunjukkan proporsi varians dalam data yang dapat diprediksi dari variabel independent (Plevris et al., 2022; C. Wang & Wang, 2020). Kelebihan: Mudah diinterpretasikan, antara 0 dan 1. Kekurangan: Tidak menunjukkan apakah model overfitting atau tidak.

Pemilihan metrik yang tepat sangat penting karena akan memengaruhi bagaimana kita mengevaluasi dan membandingkan model. Metrik evaluasi model adalah alat penting dalam data science. Dengan memahami dan menggunakan metrik yang tepat, kita dapat memastikan bahwa model yang kita bangun bekerja dengan baik dan sesuai dengan tujuan yang diinginkan.

8.4 Optimasi Model

Membangun model yang akurat dan efisien adalah tujuan utama dalam data science. Model awal yang kita buat mungkin belum memberikan hasil yang optimal. Di sinilah pentingnya optimasi model. Optimasi model adalah proses Fine-tuning untuk meningkatkan kinerja model, baik dalam hal akurasi, kecepatan, maupun efisiensi penggunaan sumber daya. Mengoptimalkan model dalam ilmu data adalah proses penting yang melibatkan beberapa langkah dan teknik utama untuk meningkatkan kinerja dan akurasi model.

Tujuan utama dari optimasi model diantaranya :

1. Meningkatkan akurasi, model yang dioptimasi diharapkan dapat memberikan prediksi yang lebih akurat. Meningkatkan Kecepatan: Model yang dioptimasi dapat berjalan lebih cepat, sehingga menghasilkan prediksi dalam waktu yang lebih singkat.
2. Mengurangi Penggunaan Sumber Daya: Model yang dioptimasi dapat menggunakan lebih sedikit memori atau daya komputasi.
3. Mencegah Overfitting: Optimasi model juga bertujuan untuk mencegah overfitting, yaitu kondisi di mana model hanya bekerja dengan baik pada data pelatihan, tetapi buruk pada data baru.

Proses optimasi model biasanya melibatkan beberapa tahapan, antara lain:

1. Evaluasi Model Awal: Mengevaluasi kinerja model awal menggunakan metrik evaluasi yang sesuai.
2. Memilih Teknik Optimasi: Memilih teknik optimasi yang sesuai berdasarkan jenis model dan masalah yang dihadapi.
3. Menerapkan Teknik Optimasi: Menerapkan teknik optimasi yang dipilih pada model.
4. Mengevaluasi Model yang Dioptimasi: Mengevaluasi kinerja model yang telah dioptimasi dan membandingkannya dengan kinerja model awal.
5. Mengulangi Proses: Jika kinerja model belum memuaskan, ulangi proses optimasi dengan teknik yang berbeda atau pengaturan hyperparameter yang berbeda.

Ada berbagai macam teknik optimasi model yang dapat digunakan, antara lain:

1. Pemilihan Fitur (*Feature Selection*): Memilih fitur-fitur yang paling relevan untuk model. Fitur yang tidak relevan atau berlebihan dapat mengganggu

kinerja model. Memilih fitur yang paling relevan dan mengurangi dimensi data dapat secara signifikan meningkatkan kinerja model. Algoritme genetika sering digunakan untuk pemilihan fitur dan pengoptimalan parameter (Voyles & Roy, 2014).

2. Pengaturan Hyperparameter (*Hyperparameter Tuning*): Model machine learning memiliki parameter yang perlu diatur sebelum pelatihan. Pengaturan yang tepat dapat meningkatkan kinerja model secara signifikan. Penyetelan hiperparameter sangat penting untuk meningkatkan kinerja model. Teknik seperti Grid Search dan Random Search biasanya digunakan untuk menemukan hiperparameter terbaik (Anitha et al., 2024; Bezdan et al., 2024). Contoh aplikasi metode canggih seperti Algoritma Sinus Cosinus yang dikombinasikan dengan Pembelajaran Berbasis Oposisi telah menunjukkan peningkatan yang signifikan dalam kinerja model, terutama dalam tugas-tugas yang kompleks. Beberapa metode yang umum digunakan antara lain:
   a. Grid Search: Mencoba semua kombinasi hyperparameter yang mungkin.
   b. Random Search: Mencoba kombinasi hyperparameter secara acak.
   c. Bayesian Optimization: Menggunakan algoritma Bayesian untuk mencari hyperparameter yang optimal.
3. Regularisasi (Regularization): Teknik untuk mencegah overfitting dengan menambahkan penalti pada fungsi kerugian model. Beberapa jenis regularisasi yang umum digunakan adalah L1 dan L2.
4. Pruning: Mengurangi kompleksitas model dengan menghilangkan bagian-bagian yang kurang penting. Hal ini dapat meningkatkan kecepatan dan efisiensi model.
5. Ensemble Methods: Menggabungkan beberapa model untuk meningkatkan kinerja. Beberapa metode ensemble yang populer adalah Bagging, Boosting, dan Stacking.
6. Arsitektur Model: Memilih arsitektur model yang tepat untuk masalah yang dihadapi. Misalnya, untuk masalah pengolahan citra, Convolutional Neural Network (CNN) mungkin lebih cocok daripada model.

Optimasi model sebaiknya dilakukan setelah model awal selesai dibangun dan dievaluasi. Jika model menunjukkan kinerja yang kurang memuaskan, maka optimasi perlu dilakukan. Proses optimasi biasanya dilakukan secara iteratif, yaitu mencoba berbagai teknik optimasi dan mengevaluasi hasilnya hingga

ditemukan model yang optimal. Optimasi model adalah langkah penting dalam membangun model data science yang akurat, efisien, dan dapat diandalkan. Dengan menggunakan teknik optimasi yang tepat, kita dapat meningkatkan kinerja model secara signifikan dan mencegah overfitting.

Bab 9
Pengolahan Data Besar (*Big Data*)

9.1 Konsep dan Karakteristik Big Data

Big Data muncul sebagai hasil dari ledakan jumlah data yang dihasilkan dalam berbagai aspek kehidupan manusia. Dalam beberapa dekade terakhir, kemajuan teknologi informasi, proliferasi perangkat IoT, dan penggunaan media sosial secara masif telah menghasilkan data dalam jumlah yang belum pernah terjadi sebelumnya. Data ini dapat berupa transaksi digital, interaksi online, atau bahkan data sensor dari perangkat pintar. Sebagai respons, organisasi dan perusahaan mulai menyadari potensi data ini untuk mendukung pengambilan keputusan strategis, meningkatkan efisiensi, dan menciptakan inovasi baru.

Gambar 9.1: Ilustrasi Big Data

Pada awalnya, pengolahan data terbatas pada data terstruktur yang tersimpan dalam database relasional. Namun, kemunculan data semi-terstruktur dan tidak terstruktur seperti video, gambar, dan teks menuntut metode pengolahan yang lebih canggih. Teknologi Big Data, yang menggabungkan perangkat keras dan

perangkat lunak khusus, hadir untuk mengatasi tantangan ini, memungkinkan analisis data yang kompleks dan pengambilan wawasan yang lebih dalam.

9.1.1 Pengertian Big Data

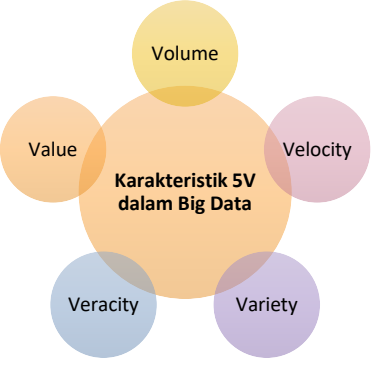
Big Data adalah istilah yang digunakan untuk menggambarkan kumpulan data yang memiliki volume sangat besar, kecepatan pengolahan yang tinggi, dan keanekaragaman tipe data yang luas sehingga tidak dapat dikelola atau dianalisis dengan metode konvensional. Data ini berasal dari berbagai sumber, termasuk media sosial, perangkat Internet of Things (IoT), transaksi digital, hingga sistem log perusahaan.

Menurut Marr (2017), Big Data tidak hanya berbicara tentang ukuran data, tetapi juga mencakup kemampuan untuk mengekstrak wawasan bermakna dari data tersebut. Konsep ini menjadi sangat relevan di era modern karena data dianggap sebagai salah satu aset paling berharga bagi organisasi. Data yang dikelola dengan baik dapat memberikan keunggulan kompetitif melalui pengambilan keputusan yang berbasis fakta dan prediksi yang akurat.

"Big Data is more than just large datasets; it's the ability to extract meaningful insights from vast amounts of structured and unstructured data" (Marr, 2017).

9.1.2 Karakteristik 5V dalam Big Data

Big Data sering kali didefinisikan melalui lima karakteristik utama yang dikenal dengan istilah 5V (Laney, 2001):



Gambar 9.2: Karakteristik 5V dalam Big Data

1. Volume

   Merujuk pada jumlah data yang sangat besar. Contohnya, perusahaan e-commerce seperti Amazon menghasilkan terabyte data setiap hari dari transaksi, ulasan pelanggan, dan data inventaris. Volume yang besar ini membutuhkan teknologi penyimpanan seperti Hadoop Distributed File System (HDFS) yang mampu menangani data skala petabyte dengan efisien.

2. Velocity

   Mengacu pada kecepatan pengumpulan dan pemrosesan data. Sebagai contoh, data sensor IoT pada kendaraan otonom perlu diproses secara real-time untuk menjamin keamanan berkendara. Platform seperti Apache Kafka digunakan untuk mengelola aliran data dalam waktu nyata.

3. Variety

   Menunjukkan beragam jenis data yang tersedia, mulai dari data terstruktur (seperti tabel database), semi-terstruktur (seperti JSON atau XML), hingga tidak terstruktur (seperti video, gambar, atau audio). Teknologi NoSQL seperti MongoDB dan Cassandra memungkinkan pengelolaan data dengan struktur yang fleksibel.

4. Veracity

   Berhubungan dengan tingkat akurasi dan keandalan data. Data yang dihasilkan dari media sosial, misalnya, mungkin mengandung bias atau informasi yang tidak valid. Dalam konteks ini, teknik data cleansing menjadi sangat penting untuk memastikan integritas data.

5. Value

   Mengacu pada nilai yang dapat dihasilkan dari data tersebut. Analisis Big Data dapat membantu organisasi menemukan pola tersembunyi, mengidentifikasi peluang baru, atau mengoptimalkan operasional. Misalnya, algoritma machine learning digunakan untuk mengungkap korelasi yang tidak terlihat dalam data besar.

"The concept of 5Vs provides a framework for understanding the opportunities and challenges that Big Data presents" (Laney, 2001).

9.1.3 Peran Big Data dalam Bisnis dan Organisasi

Big Data telah menjadi elemen penting dalam mendukung pengambilan keputusan strategis di berbagai sektor. Berikut adalah beberapa contoh implementasi Big Data:

1. Sektor Kesehatan

   Analisis data pasien yang besar dapat membantu rumah sakit memberikan perawatan yang lebih personal. Sebagai contoh, algoritma berbasis Big Data digunakan untuk memprediksi risiko penyakit tertentu berdasarkan riwayat kesehatan pasien (Raghupathi & Raghupathi, 2014). Teknologi seperti analisis genomik juga memungkinkan pengobatan yang lebih presisi.

2. Sektor Pendidikan

   Dalam bidang pendidikan, data interaksi siswa pada platform pembelajaran online dapat dianalisis untuk memantau kinerja siswa dan memprediksi hasil akademik mereka (Long & Siemens, 2011). Implementasi Learning Analytics memberikan wawasan yang dapat membantu institusi dalam merancang kurikulum yang lebih adaptif.

3. Sektor Industri

   Di sektor manufaktur, analisis pola permintaan dan penawaran melalui data besar memungkinkan optimalisasi rantai pasok. Contohnya, perusahaan logistik menggunakan data untuk merencanakan rute pengiriman yang paling efisien (Davenport & Dyché, 2014). Selain itu, penggunaan Internet of Things (IoT) memungkinkan prediktif maintenance untuk meningkatkan efisiensi operasional.

9.1.4 Manfaat dan Tantangan Umum Big Data

Big Data menawarkan berbagai manfaat yang mendukung perkembangan organisasi modern. Berikut adalah beberapa di antaranya:

1. Pengambilan Keputusan yang Lebih Baik

   Dengan analisis data yang canggih, organisasi dapat membuat keputusan yang lebih informasional dan berbasis fakta.

2. Identifikasi Peluang Baru

   Big Data memungkinkan organisasi menemukan peluang bisnis baru melalui analisis pola dan tren yang sebelumnya tidak terlihat.

3. Personalisasi Layanan

   Data pelanggan yang dianalisis secara mendalam dapat digunakan untuk memberikan layanan yang lebih personal, meningkatkan pengalaman pelanggan.

4. Efisiensi Operasional

   Dalam bidang manufaktur, misalnya, prediktif maintenance dapat diterapkan untuk mengurangi waktu henti mesin dan meningkatkan produktivitas.

Tantangan utama yang dihadapi dalam pengelolaan Big Data mencakup beberapa aspek kritis yang memerlukan perhatian serius. Dengan data yang terus berkembang dalam volume, kecepatan, dan keragamannya, organisasi menghadapi kebutuhan untuk memastikan keamanan, efisiensi, dan kualitas pengelolaan data yang optimal.

1. Keamanan dan Privasi

   Pengelolaan data besar sering kali menghadapi risiko kebocoran informasi, terutama jika melibatkan data sensitif seperti data pelanggan. Tantangan ini semakin kompleks dengan adanya regulasi privasi yang ketat, seperti GDPR di Uni Eropa, yang menuntut perusahaan untuk menerapkan langkah-langkah pengamanan yang kuat, termasuk enkripsi data dan sistem otentikasi multi-faktor. Solusi seperti teknologi blockchain juga mulai digunakan untuk memastikan integritas data.

2. Skalabilitas Teknologi

   Infrastruktur TI yang memadai diperlukan untuk menangani volume data yang besar, dan ini sering kali memerlukan investasi yang signifikan. Cloud computing menjadi solusi populer karena fleksibilitasnya dalam menyediakan sumber daya secara dinamis sesuai kebutuhan. Platform seperti Amazon Web Services (AWS) dan Microsoft Azure memberikan kemampuan untuk memproses data besar dengan efisiensi tinggi.

3. Kualitas Data

   Data yang tidak konsisten atau tidak valid dapat mengurangi efektivitas analisis. Oleh karena itu, proses data preprocessing, termasuk teknik data cleansing dan normalisasi, sangat penting untuk memastikan kualitas data. Implementasi pipeline data otomatis juga dapat membantu dalam menjaga konsistensi dan akurasi data yang masuk ke sistem.

4. Kekurangan Tenaga Ahli

   Kurangnya sumber daya manusia dengan keahlian analisis Big Data menjadi hambatan di banyak organisasi. Untuk mengatasi masalah ini, perusahaan mulai berinvestasi dalam program pelatihan dan sertifikasi bagi karyawan mereka. Selain itu, kolaborasi dengan universitas dan lembaga pendidikan dapat membantu menciptakan generasi baru analis data yang siap menghadapi tantangan di bidang ini.

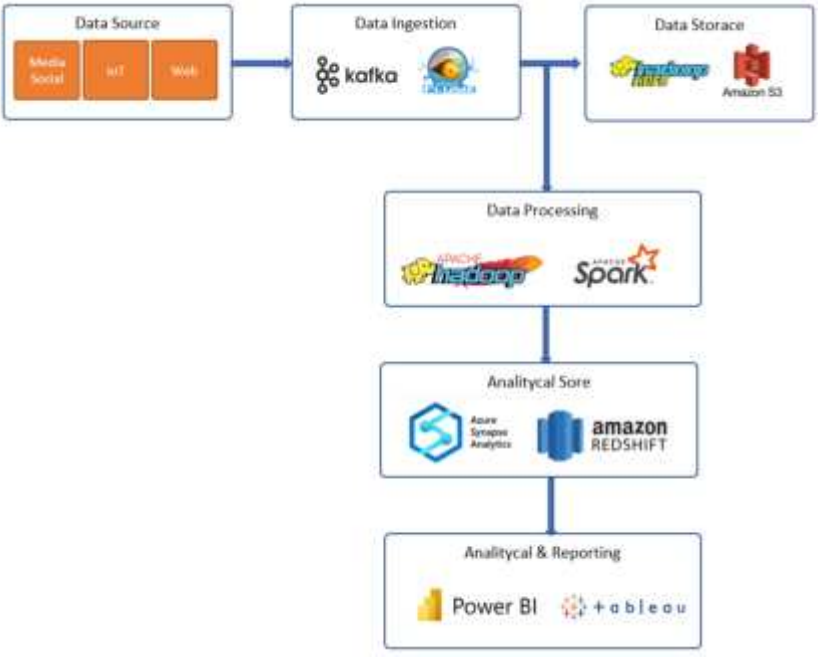
"While Big Data offers transformative opportunities, it also requires organizations to address significant technical and ethical challenges" (Kshetri, 2014).

9.2 Arsitektur dan Teknologi Big Data

Arsitektur Big Data dirancang untuk menangani volume, kecepatan, dan keragaman data yang besar secara efisien. Struktur ini terdiri dari beberapa lapisan utama, termasuk data ingestion untuk mengumpulkan data dari berbagai sumber seperti media sosial dan IoT, data processing untuk mengolah data mentah menjadi informasi yang bermakna menggunakan teknologi seperti Hadoop dan Apache Spark, serta data storage yang memungkinkan penyimpanan data dalam skala besar menggunakan sistem seperti HDFS atau Amazon S3. Dengan dukungan teknologi canggih seperti database NoSQL, arsitektur ini mampu mendukung analitik lanjutan, termasuk machine learning, guna menghasilkan wawasan strategis untuk pengambilan keputusan.

9.2.1 Arsitektur Big Data

Arsitektur Big Data dirancang untuk menangani volume, kecepatan, dan keragaman data yang besar. Sistem ini memungkinkan organisasi mengelola data secara efisien dan terorganisir, serta mendukung analitik modern yang dapat memberikan wawasan mendalam bagi pengambilan keputusan strategis (White, 2015; Zaharia et al., 2015). Dalam arsitektur ini, data melewati beberapa lapisan utama:



Gambar 9.3: Arsitektur Big Da

1. Lapisan Data Ingestion

   Lapisan ini bertugas mengumpulkan data dari berbagai sumber, seperti media sosial, sensor IoT, aplikasi web, dan transaksi digital. Teknologi seperti Apache Kafka digunakan untuk menangani pemrosesan data real-time, sementara Apache Flume dirancang untuk mengelola aliran data log. Sistem ingestion ini mendukung protokol seperti HTTP, FTP, dan MQTT untuk memastikan data dari sumber yang beragam dapat diintegrasikan dengan lancar.

2. Lapisan Data Processing

   Pada lapisan ini, data mentah diolah menjadi informasi yang dapat digunakan melalui dua pendekatan utama: pemrosesan batch dengan Hadoop MapReduce dan pemrosesan real-time dengan Apache Spark Streaming. Untuk mendukung efisiensi, arsitektur modern sering kali menggabungkan kedua pendekatan ini (*hybrid processing*). Langkah-langkah seperti transformasi data, agregasi, dan pembersihan diterapkan

untuk memastikan data berkualitas tinggi. Selain itu, teknologi machine learning seperti MLlib dari Spark memungkinkan analitik lanjutan untuk prediksi dan klasifikasi.

3. Lapisan Data Storage

   Data hasil pemrosesan membutuhkan penyimpanan yang skalabel dan andal. HDFS (Hadoop Distributed File System) sering digunakan untuk menangani data dalam jumlah besar dengan kemampuan terdistribusi. Solusi berbasis cloud seperti Amazon S3 memberikan fleksibilitas tambahan dan memungkinkan integrasi dengan alat analitik lainnya, seperti AWS Glue untuk orkestrasi data.

Arsitektur Big Data memungkinkan organisasi untuk memanfaatkan data dalam skala besar secara efisien, mendukung analitik yang lebih mendalam dan pengambilan keputusan yang lebih baik.

9.2.2 Data Lakes vs Data Warehouses

Dalam arsitektur Big Data, penyimpanan data adalah komponen kunci. Dua pendekatan utama yang sering digunakan adalah Data Lakes dan Data Warehouses. Meskipun keduanya berfungsi untuk menyimpan data, mereka memiliki perbedaan mendasar yang dirangkum dalam tabel berikut:

Tabel 9.1: Data Lakes vs Data Warehouses

Aspek	Data Lakes	Data Warehouses
Bentuk Data	Data mentah, baik terstruktur, semi-terstruktur, maupun tidak terstruktur	Data yang telah diproses dan terstruktur
Fleksibilitas	Tinggi, cocok untuk eksplorasi data dan analitik tingkat lanjut seperti machine learning	Terbatas pada analitik bisnis berbasis query SQL
Performa Query	Bergantung pada alat tambahan (misalnya, Spark untuk query data)	Tinggi, dirancang untuk query yang cepat
Platform Populer	AWS Lake Formation, Azure Data Lake	Amazon Redshift, Snowflake

Catatan:

Data Lakes lebih cocok untuk data eksplorasi, sementara Data Warehouses optimal untuk analitik operasional yang terstruktur

9.2.3 Teknologi Big Data Utama

Teknologi-teknologi ini memungkinkan organisasi untuk memproses data dalam skala besar dengan kecepatan dan efisiensi tinggi, mendukung kebutuhan analitik modern yang semakin kompleks.

Hadoop

Framework open-source ini menjadi fondasi banyak sistem Big Data. Dengan HDFS, Hadoop menyediakan penyimpanan terdistribusi, sedangkan MapReduce memungkinkan pemrosesan data secara paralel. Hadoop sering digunakan untuk pemrosesan batch dan memiliki ekosistem yang kaya, termasuk Hive dan Pig (White, 2015).Framework open-source ini menjadi fondasi banyak sistem Big Data. Dengan HDFS, Hadoop menyediakan penyimpanan terdistribusi, sedangkan MapReduce memungkinkan pemrosesan data secara paralel. Hadoop sering digunakan untuk pemrosesan batch dan memiliki ekosistem yang kaya, termasuk Hive dan Pig.

Apache Spark

Spark menawarkan pemrosesan data in-memory yang jauh lebih cepat dibanding Hadoop. Spark mendukung berbagai beban kerja, termasuk batch processing, stream processing, dan analitik berbasis machine learning melalui pustaka Mllib (Zaharia et al., 2015).Spark menawarkan pemrosesan data in-memory yang jauh lebih cepat dibanding Hadoop. Spark mendukung berbagai beban kerja, termasuk batch processing, stream processing, dan analitik berbasis machine learning melalui pustaka MLlib.

NoSQL Databases

MongoDB adalah sistem database NoSQL yang dirancang untuk mengelola data semi-terstruktur dalam format dokumen JSON. Sistem ini menawarkan fleksibilitas tinggi untuk aplikasi yang membutuhkan skalabilitas horizontal dan kemampuan menyimpan data yang bervariasi dalam strukturnya (Strauch, 2012). Cassandra, di sisi lain, adalah database terdistribusi yang dirancang untuk menangani data dalam volume besar dengan ketersediaan tinggi. Sistem ini ideal untuk aplikasi yang membutuhkan latensi rendah dan kemampuan menulis serta

membaca data secara cepat, seperti platform media sosial atau e-commerce (Guller, 2015).

9.2.4 Ekosistem Big Data

Ekosistem Big Data memainkan peran penting dalam mendukung berbagai kebutuhan analitik dan operasional. Alat-alat seperti Hive, Pig, HBase, dan Kafka dirancang untuk menyederhanakan proses pengelolaan, pemrosesan, dan analitik data dalam skala besar. Dengan integrasi yang kuat, ekosistem ini memungkinkan organisasi untuk mendapatkan wawasan yang lebih mendalam dan mendukung pengambilan keputusan strategis (White, 2015; Zaharia et al., 2015).

Tabel 9.2: Alat dan Fungsi Utama dalam Ekosistem Big Data

Alat	Deskripsi	Kelebihan	Kasus Penggunaan
Hive 	Hive adalah alat berbasis SQL yang memungkinkan pengguna menjalankan query terhadap data yang disimpan di Hadoop. Cocok untuk analitik data batch dengan sintaks yang mudah dipahami oleh pengguna SQL.	• Antarmuka yang familiar untuk pengguna SQL. • Mendukung skala besar untuk query analitik pada data yang terdistribusi. • Mendukung partisi dan bucket pada dataset besar, meningkatkan performa query.	• Analitik data transaksi pada e-commerce. • Query historis pada data pelanggan.
Pig 	Pig adalah framework scripting yang dirancang untuk menyederhanakan proses pengolahan data besar menggunakan bahasa tingkat tinggi bernama Pig Latin.	• Sederhana dan mudah digunakan untuk proses ETL dibandingkan dengan MapReduce. • Mendukung extensibility dengan fungsi definisi pengguna (User-Defined Functions, UDF).	• Membersihkan dan menggabungkan data mentah dari berbagai sumber. • Agregasi data sensor IoT.
Hbase 	HBase adalah database NoSQL berbasis kolom yang diimplementasikan di atas HDFS, ideal untuk data dengan volume besar dan	• Latensi rendah untuk baca/tulis data besar. • Skalabilitas tinggi untuk data yang memerlukan akses kolumnar. • Mendukung integrasi langsung dengan Hadoop.	• Aplikasi real-time seperti rekomendasi film. • Sistem manajemen pelanggan

Alat	Deskripsi	Kelebihan	Kasus Penggunaan
	kebutuhan akses waktu nyata.		pada bank/telekomunikasi.
Kafka 	Kafka adalah platform messaging terdistribusi yang dirancang untuk menangani aliran data real-time dengan throughput tinggi.	• Mendukung throughput tinggi untuk pengiriman pesan real-time. • Skalabilitas horizontal untuk menangani data besar dengan menambah node. • Mendukung penyimpanan pesan sementara untuk menghindari kehilangan data.	• Streaming log aplikasi untuk analisis • Pengiriman data real-time dari IoT ke sistem

9.3 Tantangan dan Solusi Big Data

Big Data juga membuka peluang untuk meningkatkan inovasi dalam berbagai sektor, termasuk kesehatan, pendidikan, dan manufaktur. Misalnya, analisis data dalam skala besar memungkinkan prediksi yang lebih akurat dan pengambilan keputusan yang lebih baik, memberikan keunggulan kompetitif bagi organisasi yang mampu mengelola data secara efektif (Erl et al., 2016).

9.3.1 Tantangan Big Data

Big Data menghadirkan peluang besar dalam berbagai sektor industri, namun implementasinya tidak tanpa tantangan. Berikut adalah beberapa tantangan utama yang sering dihadapi oleh organisasi dalam mengelola Big Data:

1. Skalabilitas

   Volume data yang terus meningkat secara eksponensial memerlukan infrastruktur yang mampu berkembang seiring waktu. Data dari berbagai sumber, seperti media sosial, perangkat IoT, dan log aplikasi, menciptakan kebutuhan akan sistem yang mampu menangani data dalam skala petabyte hingga exabyte.

   Contoh Kasus: Sistem manajemen basis data tradisional (RDBMS) sering kali tidak mampu memenuhi tuntutan analisis data yang sangat besar, sehingga memerlukan solusi yang lebih modern dan fleksibel.

2. Keamanan Data

   Data dalam skala besar sering kali mencakup informasi sensitif seperti data pelanggan, transaksi finansial, dan catatan medis. Ancaman keamanan

seperti serangan siber, ransomware, dan ancaman dari dalam organisasi memerlukan perhatian khusus.

Contoh Kasus: Insiden pelanggaran data seperti skandal Facebook-Cambridge Analytica menunjukkan pentingnya langkah-langkah keamanan dalam pengelolaan data besar.

3. Privasi

   Peraturan seperti GDPR (General Data Protection Regulation) di Uni Eropa dan UU Perlindungan Data Pribadi di Indonesia mengharuskan organisasi memastikan bahwa data pribadi dilindungi dengan ketat. Penggunaan data tanpa persetujuan eksplisit dapat menimbulkan sanksi hukum yang berat.

   Contoh Kasus: Analisis data pelanggan untuk tujuan pemasaran tanpa izin eksplisit dapat menciptakan masalah hukum dan etika.

4. Integrasi Data

   Data Big Data berasal dari berbagai sumber dan sering kali dalam format yang berbeda-beda. Mengintegrasikan data ini menjadi sebuah sistem analitik yang kohesif merupakan tantangan teknis yang signifikan.

Contoh Kasus: Platform e-commerce yang mencoba menyatukan data dari toko fisik, aplikasi mobile, dan media sosial sering menghadapi kendala dalam konsistensi dan kompatibilitas data.

9.3.2 Solusi dan Pendekatan Modern

Untuk mengatasi berbagai tantangan yang ada, berikut adalah solusi dan pendekatan modern yang dapat diadopsi:

1. Teknologi Containerization

   Teknologi seperti Docker dan Kubernetes memungkinkan penyebaran aplikasi yang lebih cepat, skalabel, dan efisien. Dengan pendekatan ini, organisasi dapat menciptakan lingkungan kerja yang konsisten dan optimal untuk pengolahan Big Data.

   Manfaat Utama: Kemampuan untuk menskalakan aplikasi Big Data secara dinamis, sesuai kebutuhan.

2. Cloud Computing

   Layanan cloud seperti Amazon Web Services (AWS), Microsoft Azure, dan Google Cloud Platform (GCP) menawarkan infrastruktur fleksibel yang dapat disesuaikan dengan kebutuhan data yang besar.

Contoh Implementasi: AWS Redshift untuk data warehousing dan Google BigQuery untuk analitik real-time adalah solusi populer untuk pengolahan Big Data.

3. Solusi Keamanan Data

   Penerapan teknologi keamanan seperti enkripsi end-to-end, blockchain, dan zero trust architecture memberikan perlindungan yang lebih baik terhadap data sensitif.

   Strategi Tambahan: Penggunaan SIEM (Security Information and Event Management) untuk mendeteksi ancaman keamanan secara proaktif.

9.4 Tren Masa Depan dalam Big Data

Big Data terus berkembang seiring dengan kemajuan teknologi. Berikut adalah beberapa tren masa depan yang menjanjikan:

1. AI-Driven Big Data

   Kecerdasan buatan (AI) kini menjadi bagian integral dari analitik Big Data. AI membantu meningkatkan efisiensi dalam pengolahan data, baik dalam analisis prediktif, pengenalan pola, maupun analisis sentimen.

   Contoh Tren: Penggunaan Natural Language Processing (NLP) untuk menganalisis data teks secara otomatis.

2. Quantum Computing

   Quantum computing menjanjikan terobosan besar dalam pengolahan data. Teknologi ini dapat mempercepat proses analitik yang kompleks, seperti optimisasi logistik atau simulasi data berskala besar.

   Contoh Kasus: Penelitian Quantum Machine Learning (QML) yang memungkinkan pengolahan dataset besar dengan efisiensi tinggi.

3. Edge Computing

   Dengan maraknya perangkat IoT, edge computing menjadi solusi yang semakin relevan. Data diproses di tepi jaringan, mengurangi latensi dan beban pada pusat data.

   Contoh Implementasi: Pengolahan data langsung di perangkat pintar seperti kamera pengawas atau sensor IoT.

Bab 10
Teknik Prediksi dan Analitik Data

10.1 Pendahuluan

Teknik prediksi dan analitik merupakan inti dari praktik Data Science, digunakan untuk menghasilkan wawasan berbasis data yang dapat mendukung pengambilan keputusan strategis. Dalam konteks Data Science, prediksi berfokus pada penggunaan algoritma untuk memperkirakan nilai masa depan berdasarkan pola dalam data historis. Teknik analitik mencakup analisis deskriptif, diagnostik, prediktif, dan preskriptif, masing-masing memberikan wawasan pada tahap yang berbeda dalam siklus data.

Gambar 10.1: Ilustrasi Pemanfaatan Teknik Prediksi dan Analitik Data

10.1.1 Teknik Prediksi

Prediksi sering kali mengandalkan algoritma machine learning, seperti regresi linier, decision trees, dan neural networks. Algoritma ini memungkinkan analisis data yang kompleks dan pemodelan hubungan non-linear antara variabel. Misalnya, regresi linier sederhana digunakan untuk memperkirakan nilai kuantitatif, sedangkan neural networks lebih cocok untuk menangani data berdimensi tinggi seperti gambar dan teks (James et al., 2013). Menurut (Géron, 2019), pengembangan model prediktif membutuhkan data yang berkualitas, yang mencakup pengolahan awal data (*pre-processing*), pemilihan fitur (*feature selection*), dan validasi model. Géron menekankan pentingnya teknik validasi silang (*cross-validation*) dalam memastikan generalisasi model terhadap data baru.

10.1.2 Teknik Analitik

Teknik analitik mencakup eksplorasi data untuk mengidentifikasi pola dan hubungan antara variabel. Analisis deskriptif digunakan untuk meringkas data, sedangkan analisis diagnostik membantu mengidentifikasi penyebab di balik pola tertentu. Analisis prediktif memanfaatkan algoritma statistik dan machine learning untuk memprediksi hasil masa depan, sementara analisis preskriptif memberikan rekomendasi tindakan berdasarkan hasil prediksi. Sebagai contoh, (Witten et al., 2017) menekankan penggunaan algoritma seperti k-nearest neighbors dan support vector machines untuk analitik prediktif, terutama dalam kasus klasifikasi data. Selain itu, kombinasi teknik analitik dengan pemrosesan bahasa alami (*Natural Language Processing*) dan analitik big data membuka peluang baru dalam berbagai domain, termasuk e-commerce dan kesehatan (Nguyen et al., 2020).

10.1.3 Implementasi dan Tantangan

Implementasi teknik prediksi dan analitik menghadapi tantangan terkait kualitas data, bias algoritma, dan interpretabilitas model. Adopsi algoritma yang transparan seperti decision trees dapat membantu meningkatkan pemahaman tentang hasil analisis (Chen et al., 2021). Selain itu, integrasi teknik seperti explainable AI (XAI) membantu mengurangi risiko bias dan memastikan penggunaan yang etis. Pengembangan teknik prediksi dan analitik yang efektif membutuhkan pemahaman mendalam tentang data, algoritma, dan domain aplikasi. Dengan demikian, data scientist harus memanfaatkan pendekatan

multidisiplin untuk mengatasi berbagai kompleksitas yang muncul dalam siklus pengolahan data.

10.2 Tahapan Dalam Data Science

Data Science adalah disiplin ilmu yang menggabungkan statistik, komputasi, dan pemahaman domain untuk mengekstraksi wawasan dari data. Proses dalam Data Science terdiri dari beberapa tahapan yang saling terkait, yang dimulai dari pemahaman masalah hingga penyampaian hasil analisis kepada pemangku kepentingan. Tahapan-tahapan ini dirancang untuk memastikan hasil analisis yang valid, dapat diandalkan, dan relevan.

10.2.1 Pemahaman Masalah

Tahap pertama dalam proyek Data Science adalah memahami masalah yang ingin diselesaikan. Memahami konteks bisnis atau domain adalah langkah penting yang memandu keseluruhan proyek (Provost & Fawcett, 2013). Data scientist harus bekerja sama dengan pemangku kepentingan untuk merumuskan pertanyaan penelitian atau masalah yang spesifik.

10.2.2 Pengumpulan Data

Data yang relevan dikumpulkan dari berbagai sumber seperti database, API, atau web scraping. Kualitas data sangat menentukan keberhasilan tahap berikutnya (Géron, 2019). Proses pengumpulan data mencakup identifikasi sumber data yang andal dan memastikan data tersebut sesuai dengan kebutuhan analisis.

10.2.3 Eksplorasi dan Pembersihan Data

Tahapan eksplorasi dan pembersihan data mencakup identifikasi nilai-nilai yang hilang, outlier, dan inkonsistensi dalam dataset. Tahap ini sangat penting karena data yang buruk dapat menghasilkan model yang tidak akurat. Teknik visualisasi, seperti histogram dan scatter plots, sering digunakan untuk memahami distribusi data (Witten et al., 2017),.

10.2.4 Pemodelan

Tahap pemodelan melibatkan penerapan algoritma machine learning atau teknik statistik untuk membuat prediksi atau mengidentifikasi pola dalam data.

Pemilihan model bergantung pada jenis masalah, apakah itu klasifikasi, regresi, atau clustering. Validasi silang penting untuk memastikan model tidak mengalami overfitting (Géron, 2019).

10.2.5 Evaluasi Model

Setelah model dilatih, kinerjanya dievaluasi menggunakan metrik seperti akurasi, precision, recall, atau mean squared error. Menurut Kotsiantis (2020), evaluasi yang baik mencakup analisis kesalahan untuk mengidentifikasi kelemahan model dan peluang perbaikan.

10.2.6 Penerapan dan Komunikasi Hasil

Tahap akhir adalah penerapan model dan komunikasi hasil analisis kepada pemangku kepentingan. Ini mencakup visualisasi data dan penulisan laporan yang mudah dipahami. Storytelling berbasis data menjadi alat yang efektif untuk menjembatani kesenjangan antara analisis teknis dan keputusan bisnis (Nguyen et al., 2020).

10.2.7 Tantangan dalam Proses

Setiap tahapan memiliki tantangan tersendiri, seperti kurangnya data berkualitas, kesalahan algoritma, dan bias interpretasi. Oleh karena itu, pendekatan iteratif sering kali diperlukan untuk memperbaiki hasil secara berkelanjutan.

10.3 Pengumpulan dan Penyiapan Data

Pengumpulan dan penyiapan data adalah langkah fundamental dalam proses Data Science. Langkah ini mencakup berbagai aktivitas yang bertujuan untuk mendapatkan data berkualitas tinggi yang dapat digunakan dalam analisis atau pengembangan model prediktif. Tanpa data yang bersih dan relevan, hasil analisis dapat menjadi tidak akurat dan tidak dapat diandalkan.

10.3.1 Pengumpulan Data

Pengumpulan data melibatkan proses memperoleh informasi dari berbagai sumber, baik internal maupun eksternal. Menurut (Géron, 2019), sumber data dapat berupa basis data relasional, API, file CSV, atau data web yang diekstraksi melalui web scraping. Dalam beberapa kasus, data dapat berasal dari sensor,

survei, atau transaksi pengguna. Sumber data harus dipilih dengan hati-hati untuk memastikan kelengkapan dan relevansinya. Provost dan Fawcett (2013) menekankan pentingnya memahami kebutuhan bisnis untuk menentukan jenis data yang harus dikumpulkan. Misalnya, dalam e-commerce, data transaksi dan perilaku pelanggan adalah elemen utama untuk analisis.

10.3.2 Penyiapan Data

Setelah data dikumpulkan, tahap penyiapan data dilakukan untuk memastikan kualitas dan kegunaannya. Penyiapan data melibatkan langkah-langkah berikut (Witten et al., 2017):

1) Pembersihan Data: Mengidentifikasi dan menangani data yang hilang, outlier, atau duplikasi.
2) Transformasi Data: Mengubah format data, seperti mengonversi data kategori menjadi format numerik.
3) Integrasi Data: Menggabungkan data dari berbagai sumber ke dalam satu dataset terpadu.
4) Pemilihan Fitur: Memilih atribut yang relevan dengan tujuan analisis untuk mengurangi dimensi data.

10.3.3 Tantangan dalam Pengumpulan dan Penyiapan Data

Proses pengumpulan dan penyiapan data menghadapi berbagai tantangan, seperti kualitas data yang buruk, format data yang tidak konsisten, dan masalah privasi. Data besar sering kali mengandung noise dan redundansi, yang memerlukan algoritma dan teknik canggih untuk mengatasinya (Nguyen et al., 2020). Automasi dalam penyiapan data, seperti penggunaan teknik Machine Learning untuk pembersihan data, dapat meningkatkan efisiensi proses. Namun, interpretabilitas tetap menjadi aspek penting, terutama dalam aplikasi kritis seperti kesehatan (Z. Wang et al., 2021).

10.3.4 Peran Data dalam Keberhasilan Analisis

Data yang berkualitas tinggi adalah dasar dari model yang andal. Pengumpulan dan penyiapan data yang cermat dapat mengurangi risiko bias model dan meningkatkan akurasi prediksi. Oleh karena itu, tahapan ini tidak boleh dianggap remeh dalam siklus Data Science (Kotsiantis, 2020).

10.4 Eksplorasi dan Visualisasi Data

Eksplorasi dan visualisasi data adalah tahap penting dalam proses Data Science yang bertujuan untuk memahami struktur, pola, dan hubungan dalam data. Tahap ini membantu data scientist mengidentifikasi anomali, outlier, dan tren yang relevan, yang kemudian menjadi dasar untuk analisis lebih lanjut atau pembuatan model prediktif. Eksplorasi dan visualisasi data adalah fondasi penting untuk analisis data yang lebih dalam. Dengan alat yang tepat dan penerapan prinsip desain yang baik, tahap ini dapat memberikan wawasan yang signifikan dan mendukung komunikasi hasil yang efektif kepada berbagai pemangku kepentingan.

10.4.1 Eksplorasi Data

Eksplorasi data merupakan langkah awal dalam memahami dataset. Menurut James et al. (2013), proses ini mencakup analisis statistik deskriptif seperti mean, median, varians, dan distribusi data. Langkah ini bertujuan untuk memperoleh wawasan awal tentang karakteristik data. Pentingnya eksplorasi data terletak pada kemampuan untuk menemukan pola tersembunyi atau ketidakkonsistenan yang dapat memengaruhi hasil analisis. Géron (2019) menekankan bahwa eksplorasi data memungkinkan identifikasi nilai yang hilang, duplikasi, atau skala yang tidak seimbang di antara variabel. Selain itu, teknik seperti analisis korelasi sering digunakan untuk mengevaluasi hubungan antarvariabel.

10.4.2 Visualisasi Data

Visualisasi data adalah representasi grafis dari informasi untuk mempermudah pemahaman dan komunikasi. Menurut Witten et al. (2017), visualisasi membantu menggambarkan pola yang sulit diidentifikasi melalui analisis statistik saja. Grafik seperti scatter plots, bar charts, dan heatmaps sangat umum digunakan dalam tahap ini. Visualisasi data tidak hanya mendukung eksplorasi, tetapi juga memfasilitasi komunikasi hasil analisis kepada pemangku kepentingan. Nguyen et al. (2020) mencatat bahwa storytelling berbasis data yang menggunakan visualisasi efektif dapat menjembatani kesenjangan antara hasil analitik yang kompleks dan kebutuhan bisnis.

10.4.3 Teknologi dan Alat

Berbagai alat dan teknologi mendukung eksplorasi dan visualisasi data. Python dengan pustaka seperti Matplotlib, Seaborn, dan Plotly, serta R dengan ggplot2 adalah beberapa yang paling populer. Menurut Wang et al. (2021), alat-alat ini memungkinkan pembuatan visualisasi yang interaktif dan informatif, yang sangat bermanfaat dalam presentasi data besar (big data).

10.4.4 Tantangan dan Solusi

Proses eksplorasi dan visualisasi data menghadapi tantangan seperti bias interpretasi, data yang besar dan tidak terstruktur, serta kesulitan dalam merancang grafik yang efektif. Chen et al. (2021) menyarankan bahwa penggunaan prinsip-prinsip visualisasi yang baik, seperti menjaga kesederhanaan dan relevansi, dapat membantu mengatasi masalah ini. Selain itu, automasi dalam visualisasi data melalui dashboard interaktif memungkinkan pemantauan data secara real-time. Kotsiantis (2020) mencatat bahwa ini menjadi tren penting dalam industri untuk meningkatkan efisiensi dan akurasi pengambilan keputusan berbasis data.

10.5 Teknik, Evaluasi dan Tantangan Pemodelan Prediktif

Pemodelan prediktif adalah esensi dari Data Science, digunakan untuk memperkirakan nilai atau kategori berdasarkan data historis. Proses ini melibatkan pemilihan algoritma yang sesuai, pelatihan model, dan evaluasi kinerja untuk memastikan prediksi yang akurat.

10.5.1 Teknik Pemodelan Prediktif

Teknik pemodelan prediktif mencakup algoritma berbasis statistik dan machine learning. Menurut Géron (2019), regresi linier adalah teknik dasar untuk prediksi kuantitatif, sementara algoritma seperti random forest dan gradient boosting lebih cocok untuk masalah kompleks dengan data besar. James et al. (2013) menjelaskan bahwa klasifikasi, regresi, dan clustering adalah tiga kategori utama dalam pemodelan prediktif. Contohnya, support vector machines (SVM) digunakan untuk klasifikasi data dengan dimensi tinggi, sedangkan k-means cocok untuk clustering. Teknik seperti neural networks dan deep learning telah membuka peluang baru dalam pemrosesan data yang tidak terstruktur seperti gambar dan teks. Witten et al. (2017) menambahkan bahwa pemodelan prediktif juga melibatkan fitur engineering, yaitu proses membuat

variabel baru atau memilih fitur yang relevan untuk meningkatkan performa model.

10.5.2 Evaluasi Pemodelan

Evaluasi model prediktif bertujuan untuk memastikan akurasi dan generalisasi model. Metrik evaluasi seperti akurasi, precision, recall, dan F1-score sering digunakan dalam klasifikasi, sementara *Mean Squared Error* (MSE) dan Root Mean Squared Error (RMSE) adalah standar dalam regresi (Nguyen et al., 2020). Proses validasi silang (*cross-validation*) juga penting untuk menghindari *overfitting*, yaitu ketika model terlalu cocok dengan data latih sehingga kinerjanya buruk pada data baru. Wang et al. (2021) menekankan pentingnya membagi dataset menjadi data latih, validasi, dan uji untuk mengevaluasi kinerja model secara menyeluruh.

10.5.3 Tantangan dalam Pemodelan Prediktif

Tantangan dalam pemodelan prediktif mencakup data yang tidak seimbang, interpretabilitas model, dan bias algoritma. Chen et al. (2021) menyarankan penggunaan explainable AI (XAI) untuk meningkatkan transparansi dan interpretabilitas model, terutama dalam aplikasi kritis seperti kesehatan. Keseluruhan, keberhasilan pemodelan prediktif bergantung pada pemilihan algoritma yang tepat, kualitas data, dan proses evaluasi yang cermat untuk memastikan hasil yang dapat dipercaya dan relevan.

10.6 Model Prediktif - Regresi Linear

Regresi linear adalah salah satu teknik pemodelan prediktif yang paling dasar dan banyak digunakan dalam Data Science. Model ini memprediksi nilai variabel dependen (respons) berdasarkan hubungan linear dengan satu atau lebih variabel independen (prediktor). Sederhana, transparan, dan mudah diinterpretasikan, regresi linear sering menjadi titik awal dalam analisis prediktif.

10.6.1 Teknik Pemodelan

Model regresi linear dapat dibagi menjadi dua jenis: regresi linier sederhana (dengan satu prediktor) dan regresi linier berganda (dengan lebih dari satu prediktor). James et al. (2013) menjelaskan bahwa teknik ini mengasumsikan hubungan linear antara variabel, normalitas residual, dan homogenitas varians.

Proses fitting model melibatkan minimisasi sum of squared residuals (SSR) melalui metode least squares. Selain itu, Géron (2019) menyebutkan bahwa regularisasi seperti Ridge dan Lasso dapat digunakan untuk mengurangi risiko overfitting dalam regresi linier berganda, terutama ketika terdapat multikolinearitas di antara prediktor.

10.6.2 Evaluasi Model

Evaluasi model regresi linier dilakukan dengan mengukur akurasi prediksi dan validitas asumsi model. Metrik evaluasi utama meliputi *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *Coefficient of Determination* (R^2). Witten et al. (2017) menekankan pentingnya analisis residual untuk memvalidasi asumsi linearitas dan normalitas dalam model. Menurut Nguyen et al. (2020), validasi silang (cross-validation) adalah langkah penting untuk memastikan model dapat digeneralisasi ke data baru. Teknik ini membagi dataset menjadi data latih dan uji untuk mengevaluasi kinerja model secara obyektif.

10.6.3 Kelebihan dan Keterbatasan

Regresi linier unggul dalam interpretabilitas dan efisiensi komputasi. Namun, model ini memiliki keterbatasan dalam menangkap hubungan non-linear. Wang et al. (2021) menyarankan penggunaan teknik lain, seperti regresi polinomial atau algoritma machine learning, jika pola data tidak sesuai dengan asumsi linearitas .

10.7 Model Prediktif - Klasifikasi

Klasifikasi adalah salah satu teknik pemodelan prediktif yang bertujuan untuk mengelompokkan data ke dalam kategori atau kelas tertentu. Teknik ini sering digunakan dalam berbagai aplikasi, seperti deteksi penipuan, diagnosis medis, dan analisis sentimen.

10.7.1 Teknik Pemodelan

Model klasifikasi bekerja dengan mempelajari pola dari data latih untuk memprediksi kelas data baru. Menurut James et al. (2013), beberapa algoritma yang umum digunakan dalam klasifikasi adalah decision trees, logistic regression, k-nearest neighbors (k-NN), dan support vector machines (SVM). Logistic regression adalah pilihan yang baik untuk kasus dengan hubungan

linier antara variabel independen dan probabilitas kelas, sedangkan SVM lebih unggul dalam menangani data berdimensi tinggi. Géron (2019) menyebutkan bahwa algoritma ensemble seperti Random Forest dan Gradient Boosting telah menjadi populer karena kemampuan mereka mengurangi overfitting dan meningkatkan akurasi. Teknik ini menggabungkan beberapa model untuk menghasilkan prediksi yang lebih kuat. Witten et al. (2017) menekankan pentingnya preprocessing data dalam klasifikasi, termasuk normalisasi dan pemilihan fitur, untuk meningkatkan kinerja model.

10.7.2 Evaluasi Model

Evaluasi model klasifikasi memerlukan metrik khusus untuk menilai akurasi prediksi. Metrik umum meliputi akurasi, precision, recall, dan F1-score. Menurut Nguyen et al. (2020), confusion matrix adalah alat penting untuk menganalisis kinerja model, terutama ketika data tidak seimbang. Proses validasi silang (*cross-validation*) sering digunakan untuk memastikan generalisasi model. Wang et al. (2021) menekankan pentingnya teknik ini dalam mengurangi risiko overfitting dan memastikan performa model yang konsisten pada data baru.

10.7.3 Tantangan dan Solusi

Tantangan dalam klasifikasi mencakup data yang tidak seimbang, noise, dan kompleksitas model. Chen et al. (2021) menyarankan penggunaan teknik seperti oversampling atau penalized algorithms untuk menangani data tidak seimbang. Selain itu, interpretabilitas model menjadi penting dalam aplikasi kritis, seperti kesehatan, untuk memastikan hasil dapat dipercaya.

10.8 Model Prediktif - Clustering

Clustering adalah teknik pemodelan prediktif tanpa pengawasan yang digunakan untuk mengelompokkan data ke dalam grup berdasarkan kesamaan antar data. Teknik ini sering diterapkan dalam eksplorasi data, segmentasi pelanggan, dan deteksi pola.

10.8.1 Teknik Pemodelan

Clustering memanfaatkan algoritma untuk menemukan kelompok alami (cluster) dalam data tanpa memerlukan label. Menurut Witten et al. (2017), algoritma populer untuk clustering mencakup k-means, hierarchical clustering,

dan DBSCAN. K-means adalah algoritma yang sederhana dan efisien, menggunakan pendekatan centroid untuk meminimalkan jarak antara data dalam satu cluster. Di sisi lain, hierarchical clustering menghasilkan dendrogram untuk merepresentasikan hierarki antar data. Géron (2019) menekankan bahwa pemilihan algoritma clustering tergantung pada struktur data dan tujuan analisis. Misalnya, DBSCAN sangat efektif untuk mendeteksi cluster dengan bentuk tidak teratur dan menangani outlier, yang menjadi kelemahan k-means. James et al. (2013) mencatat bahwa pemilihan jumlah cluster (k) adalah tantangan utama dalam k-means. Teknik seperti *elbow method* dan *silhouette analysis* sering digunakan untuk menentukan jumlah cluster yang optimal.

10.8.2 Evaluasi Model

Tidak seperti klasifikasi, evaluasi model clustering lebih kompleks karena tidak adanya label data. Menurut Nguyen et al. (2020), metrik evaluasi seperti silhouette score, Davies-Bouldin Index, dan Dunn Index digunakan untuk mengukur kualitas cluster berdasarkan kepadatan dan pemisahan antar cluster. Wang et al. (2021) menyarankan validasi visual melalui scatter plot atau heatmap untuk memahami struktur cluster, terutama pada data berdimensi rendah. Selain itu, validasi eksternal dapat dilakukan jika tersedia informasi label tambahan untuk membandingkan hasil clustering.

10.8.3 Tantangan dan Solusi

Clustering menghadapi tantangan seperti sensitivitas terhadap outlier, skala data, dan ketidakseimbangan cluster. Chen et al. (2021) merekomendasikan normalisasi data sebelum clustering untuk mengurangi pengaruh skala, serta algoritma robust seperti DBSCAN untuk menangani outlier.

10.9 Model Prediktif - Prediksi Deret Waktu

Prediksi deret waktu adalah metode pemodelan prediktif yang berfokus pada data yang terorganisasi berdasarkan waktu. Teknik ini sering digunakan dalam aplikasi seperti peramalan ekonomi, analisis stok pasar, dan prediksi permintaan energi.

10.9.1 Teknik Pemodelan

Model prediksi deret waktu menggunakan pola dalam data historis untuk memperkirakan nilai masa depan. Menurut James et al. (2013), model yang umum digunakan meliputi model linier seperti Autoregressive Integrated Moving Average (ARIMA) dan model non-linier seperti Long Short-Term Memory (LSTM) dalam deep learning. Géron (2019) mencatat bahwa ARIMA merupakan pilihan yang efektif untuk data yang stasioner, sementara LSTM lebih unggul dalam menangkap pola kompleks dalam data tidak stasioner dan berdimensi tinggi. Transformasi data seperti dekomposisi deret waktu menjadi tren, musiman, dan residual sering dilakukan untuk meningkatkan performa model. Witten et al. (2017) menekankan pentingnya preprocessing dalam prediksi deret waktu, seperti normalisasi data dan penanganan outlier. Selain itu, pengujian seperti Augmented Dickey-Fuller (ADF) digunakan untuk menguji stasionaritas data.

10.9.2 Evaluasi Model

Evaluasi model prediksi deret waktu memerlukan metrik yang dapat menangkap kesalahan prediksi. Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan Mean Absolute Percentage Error (MAPE) adalah metrik yang umum digunakan (Nguyen et al., 2020).

Wang et al. (2021) menekankan pentingnya pembagian data menjadi train-test split berdasarkan urutan waktu untuk menjaga kronologi data. Teknik seperti walk-forward validation membantu mengevaluasi performa model pada berbagai jendela waktu, memastikan generalisasi yang baik.

10.9.3 Tantangan dan Solusi

Tantangan utama dalam prediksi deret waktu mencakup volatilitas data, pola musiman, dan perubahan tren. Chen et al. (2021) merekomendasikan penggunaan model hybrid yang menggabungkan algoritma linier dan non-linier untuk menangani kompleksitas data.

10.10 Model Prediktif - Pengenalan Pola

Pengenalan pola adalah teknik pemodelan prediktif yang bertujuan untuk mengidentifikasi struktur atau pola dalam data dan mengklasifikasikannya ke dalam kategori tertentu. Metode ini banyak digunakan dalam berbagai bidang seperti pengenalan wajah, analisis teks, dan deteksi anomali.

10.10.1 Teknik Pemodelan

Menurut James et al. (2013), pengenalan pola melibatkan kombinasi fitur data dengan algoritma pembelajaran mesin untuk menghasilkan model yang dapat mengenali pola baru. Algoritma yang umum digunakan termasuk k-nearest neighbors (k-NN), support vector machines (SVM), dan neural networks. Géron (2019) menekankan bahwa pemilihan fitur sangat penting dalam pengenalan pola. Teknik seperti principal component analysis (PCA) sering digunakan untuk mereduksi dimensi data tanpa kehilangan informasi yang signifikan. Selain itu, deep learning dengan convolutional neural networks (CNN) telah menjadi metode unggulan dalam pengenalan pola berbasis gambar. Witten et al. (2017) menambahkan bahwa preprocessing data, seperti normalisasi dan augmentasi data, dapat meningkatkan akurasi model dengan membuat data lebih representatif untuk pengenalan pola.

10.10.2 Evaluasi Model

Evaluasi model dilakukan dengan metrik seperti akurasi, precision, recall, F1-score, dan area under the curve (AUC) (Nguyen et al., 2020). Validasi silang (cross-validation) juga digunakan untuk memastikan model tidak overfitting. Wang et al. (2021) menyarankan penggunaan dataset uji yang terpisah untuk mengevaluasi generalisasi model dalam pengenalan pola. Selain itu, analisis kesalahan dapat membantu mengidentifikasi kelemahan model dan memberikan wawasan untuk peningkatan.

10.10.3 Tantangan dan Solusi

Tantangan utama dalam pengenalan pola adalah ketidakseimbangan data, noise, dan interpretabilitas model. Chen et al. (2021) merekomendasikan teknik explainable AI (XAI) untuk membuat model pengenalan pola lebih transparan, khususnya dalam aplikasi kritis seperti kesehatan dan keamanan.

10.11 Validasi Silang dan Overfitting

Validasi silang (*cross-validation*) adalah metode penting dalam Data Science untuk mengevaluasi kinerja model dan mencegah overfitting. Overfitting adalah kondisi di mana model terlalu cocok dengan data latih sehingga kehilangan kemampuan untuk melakukan generalisasi pada data baru. Validasi silang adalah komponen penting dalam siklus Data Science untuk memastikan performa model yang optimal dan menghindari overfitting. Dengan kombinasi

teknik validasi dan pengaturan model yang baik, data scientist dapat menghasilkan solusi yang lebih andal dan dapat diandalkan pada data nyata.

10.11.1 Validasi Silang

Validasi silang adalah teknik untuk membagi data menjadi beberapa subset guna menguji performa model pada data yang tidak digunakan dalam pelatihan. Menurut Géron (2019), metode ini memberikan estimasi yang lebih andal mengenai kinerja model dibandingkan hanya menggunakan pembagian tunggal seperti data latih dan uji. Metode yang paling umum digunakan adalah *k-fold cross-validation*, di mana dataset dibagi menjadi kk bagian, dan model dilatih serta diuji sebanyak kk kali dengan bagian data yang berbeda. James et al. (2013) menyarankan bahwa nilai kk yang umum digunakan adalah 5 atau 10, karena memberikan keseimbangan antara bias dan varians evaluasi. Witten et al. (2017) juga menjelaskan teknik *leave-one-out cross-validation* (LOOCV), di mana setiap pengamatan digunakan sebagai data uji satu per satu. Meskipun memberikan estimasi kinerja yang akurat, LOOCV bisa menjadi sangat mahal secara komputasi.

10.11.2 Overfitting dan Solusi

Overfitting terjadi ketika model menangkap noise atau detail yang tidak relevan dalam data latih, sehingga performanya menurun pada data baru. Nguyen et al. (2020) menjelaskan bahwa overfitting sering terjadi pada model yang kompleks, seperti deep learning, atau ketika data terlalu sedikit. Beberapa teknik untuk mencegah overfitting meliputi:

1) Validasi Silang: Sebagai langkah evaluasi yang memastikan model tidak hanya cocok pada data latih.
2) Regularisasi: Géron (2019) menyarankan penggunaan metode seperti Lasso atau Ridge, yang menambahkan penalti pada koefisien besar dalam regresi linier.
3) Early Stopping: Dalam pembelajaran mendalam, pelatihan dihentikan ketika performa pada data validasi mulai menurun.
4) Augmentasi Data: Wang et al. (2021) mencatat bahwa augmentasi data membantu meningkatkan generalisasi model dengan menciptakan variasi dalam data latih.

10.11.3 Urgensi Validasi Silang

Validasi silang membantu mengidentifikasi model yang terlalu sederhana (*underfitting*) atau terlalu kompleks (*overfitting*). Chen et al. (2021) menekankan pentingnya teknik ini dalam memilih model yang optimal dan memastikan bahwa hasil prediksi relevan pada data baru. Selain itu, validasi silang juga dapat digunakan untuk membandingkan berbagai algoritma guna menemukan yang terbaik untuk masalah tertentu.

10.12 Analisis dan Implementasi Model Prediktif

Model prediktif adalah alat penting dalam Data Science yang digunakan untuk memprediksi hasil berdasarkan data historis. Proses analisis dan implementasi model prediktif melibatkan langkah-langkah yang terstruktur untuk memastikan keandalan dan efektivitas model dalam memecahkan masalah nyata. Analisis dan implementasi model prediktif adalah proses yang kompleks namun penting dalam Data Science. Keberhasilan tahap ini tergantung pada pemilihan algoritma yang tepat, evaluasi model yang cermat, serta integrasi yang efisien ke dalam sistem produksi. Dengan pendekatan iteratif, model dapat dioptimalkan untuk memberikan hasil yang akurat dan relevan sesuai dengan kebutuhan bisnis.

10.12.1 Analisis Model Prediktif

Analisis model prediktif dimulai dengan memilih algoritma yang sesuai berdasarkan tujuan dan sifat data. Menurut Géron (2019), algoritma seperti regresi linier digunakan untuk prediksi kuantitatif, sedangkan random forest dan *Support Vector Machines* (SVM) cocok untuk klasifikasi. Dalam beberapa kasus, model deep learning seperti neural networks diterapkan untuk data kompleks seperti gambar atau teks. Proses analisis melibatkan:

1) Pemilihan Fitur: Memilih variabel yang relevan untuk memastikan model menangkap pola yang signifikan.
2) Evaluasi Model: Validasi silang (*cross-validation*) digunakan untuk mengukur performa model dan menghindari overfitting. James et al. (2013) mencatat bahwa metrik seperti akurasi, precision, recall, dan mean squared error digunakan untuk mengevaluasi kinerja model sesuai jenis prediksi.

3) Selain itu, analisis residual membantu mengidentifikasi pola kesalahan model yang mungkin memerlukan penyesuaian lebih lanjut, seperti menambahkan fitur atau mengubah algoritma.

10.12.2 Implementasi Model Prediktif

Implementasi model melibatkan penerapan model yang telah dilatih pada sistem produksi. Menurut Nguyen et al. (2020), langkah-langkah implementasi meliputi deployment model dalam bentuk API, integrasi dengan sistem manajemen data, dan monitoring kinerja secara real-time. Untuk memastikan model tetap relevan, diperlukan strategi pembaruan model secara berkala, terutama dalam lingkungan yang dinamis seperti e-commerce atau sistem rekomendasi. Wang et al. (2021) menekankan pentingnya pemantauan metrik kinerja, seperti akurasi prediksi pada data baru, untuk mengidentifikasi degradasi performa akibat perubahan data (*data drift*).

10.12.3 Tantangan dan Solusi

Tantangan utama dalam analisis dan implementasi model prediktif mencakup bias algoritma, interpretabilitas hasil, dan integrasi ke dalam sistem yang ada. Menurut Chen et al. (2021), explainable AI (XAI) menjadi solusi penting dalam memastikan model dapat dipahami oleh pemangku kepentingan non-teknis. Selain itu, pemrosesan data yang efisien dengan memanfaatkan big data frameworks seperti Apache Spark atau TensorFlow membantu menangani volume data yang besar dalam implementasi model di skala industri (Kotsiantis, 2020).

10.13 Penggunaan Teknik Prediksi dan Analitik Data

10.13.1 Model Studi Kasus Pertama

Sampling adalah proses pengambilan subset data dari populasi yang lebih besar untuk analisis. Dalam Data Science, teknik ini memungkinkan pengolahan data yang lebih efisien dan memberikan hasil yang dapat digeneralisasi ke populasi asli. R dan Python adalah dua platform populer yang menyediakan pustaka dan fungsi bawaan untuk sampling dan analisis data prediktif. Berikut adalah contoh implementasi sampling untuk teknik prediksi dan analitik data pada customer pengguna provider GSM 4G menggunakan platform aplikasi R dan Python.

1. Pendekatan dengan R

Deskripsi:

Bila memiliki dataset pelanggan provider GSM 4G dengan informasi seperti penggunaan data bulanan, lama berlangganan, dan keluhan pelanggan. Tujuannya adalah menganalisis churn pelanggan dengan prediksi berbasis data historis.

Coding R untuk Sampling dan Prediksi:

```
# Memuat pustaka yang dibutuhkan
library(caret)
library(dplyr)

# Contoh dataset pelanggan (simulasi)
set.seed(123)
data <- data.frame(
  CustomerID = 1:1000,
  UsageData = rnorm(1000, mean = 50, sd = 15),  # Penggunaan data
(GB)
  Tenure = sample(1:60, 1000, replace = TRUE),  # Lama berlangganan
(bulan)
  Complaints = sample(0:1, 1000, replace = TRUE),  # Keluhan pelanggan
(0=tidak, 1=ya)
  Churn = sample(0:1, 1000, replace = TRUE)  # Apakah pelanggan churn
)

# Membagi data menjadi set pelatihan dan pengujian
set.seed(123)
trainIndex <- createDataPartition(data$Churn, p = 0.8, list = FALSE)
train <- data[trainIndex, ]
test <- data[-trainIndex, ]

# Model prediksi (contoh: Logistic Regression)
model <- train(Churn ~ UsageData + Tenure + Complaints,
               data = train,
               method = "glm",
               family = "binomial")
summary(model)

# Prediksi pada data uji
predictions <- predict(model, newdata = test)
confusionMatrix(predictions, as.factor(test$Churn))
```

2. Pendekatan dengan Python

Deskripsi Kasus:
Skenario serupa diterapkan di Python, dengan menggunakan pustaka seperti Pandas untuk manipulasi data, Scikit-learn untuk pembagian data, dan model Logistic Regression untuk prediksi churn pelanggan.

Coding Python untuk Sampling dan Prediksi:

```
# Memuat pustaka yang dibutuhkan
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Contoh dataset pelanggan (simulasi)
np.random.seed(123)
data = pd.DataFrame({
    'CustomerID': np.arange(1, 1001),
    'UsageData': np.random.normal(50, 15, 1000),  # Penggunaan data
(GB)
    'Tenure': np.random.randint(1, 61, 1000),  # Lama berlangganan
(bulan)
    'Complaints': np.random.choice([0, 1], size=1000),  # Keluhan
pelanggan (0=tidak, 1=ya)
    'Churn': np.random.choice([0, 1], size=1000)  # Apakah pelanggan
churn
})

# Membagi data menjadi set pelatihan dan pengujian
X = data[['UsageData', 'Tenure', 'Complaints']]
y = data['Churn']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=123, stratify=y)

# Model prediksi (contoh: Logistic Regression)
model = LogisticRegression()
model.fit(X_train, y_train)

# Prediksi pada data uji
y_pred = model.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Penjelasannya:

1) Sampling Data: dalam kedua platform, data dibagi menjadi data pelatihan dan pengujian menggunakan metode sampling acak terstratifikasi. Teknik ini memastikan proporsi kelas target (churn) terjaga.
2) Model Prediksi: Logistic Regression digunakan sebagai model dasar untuk memprediksi churn. Model ini cocok untuk klasifikasi biner dan memberikan interpretasi koefisien yang jelas.
3) Evaluasi Model: Matriks kebingungan (*confusion matrix*) dan metrik evaluasi seperti akurasi, precision, recall, dan F1-score digunakan untuk menilai kinerja model pada data uji.

Pendekatan dengan R dan Python memungkinkan analisis yang fleksibel dan prediktif pada data pelanggan provider GSM 4G. R memberikan keunggulan dalam visualisasi analisis statistik, sementara Python menawarkan efisiensi dan skalabilitas dalam pengolahan data besar.

10.13.2 Model Studi Kasus Kedua

Gunakan metode kuadrat terkecil untuk melakukan regresi linier berganda dan temukan garis yang paling sesuai untuk sekumpulan data berpasangan. Buat persamaan, grafik, dan interval prediksi.

Import libraries and located datasets:

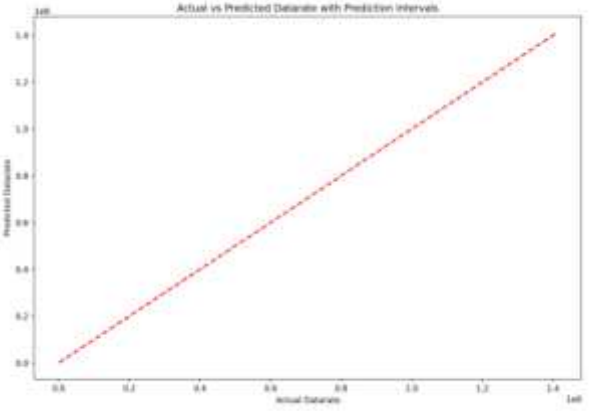
```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.sandbox.regression.predstd import wls_prediction_std
# Load the dataset
data = pd.read_csv("QoS dataset.csv")

# Display the first few rows of the dataset to understand its structure
print(data.head())
```

Model Regresi Linear dan Prediksi Plotted

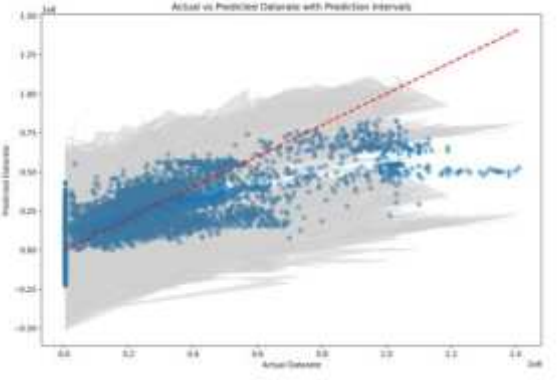
```
# Prepare the data Using datarate as dependent variable and other features as
independent variables
X = data[['speed_kmh', 'PCell_RSRP_1', 'PCell_RSRQ_1', 'PCell_RSSI_1',
'PCell_SNR_1']]
y = data['datarate']

# Add constant to predictor variables
```

```
X = sm.add_constant(X)

# Fit linear regression model
model = sm.OLS(y, X).fit()

# Calculate prediction intervals
prstd, iv_l, iv_u = wls_prediction_std(model)

# Generate predictions for plotting
y_pred = model.predict(X)

# Create the plot
plt.figure(figsize=(12, 8))
plt.scatter(y, y_pred, alpha=0.5)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--', lw=2)
plt.xlabel('Actual Datarate')
plt.ylabel('Predicted Datarate')
plt.title('Actual vs Predicted Datarate with Prediction Intervals')

# Add prediction intervals
plt.fill_between(y, iv_l, iv_u, color='gray', alpha=0.2)
plt.show()

# Print the model summary and equation coefficients
print("\
Model Summary:")
print(model.summary().tables[1])

# Create equation string
coef = model.params
equation = "Datarate = {:.2f}".format(coef[0])
for i, col in enumerate(X.columns[1:], 1):
    equation += " + ({:.2f} × {})".format(coef[i], col)

print("\
Regression Equation:")
print(equation)
```

Cuplikan koding diatas, menyiapkan data untuk model regresi linier, menyesuaikan model, menghitung interval prediksi, dan memvisualisasikan laju data aktual versus prediksi beserta persamaan regresi.



melakukan analisis regresi linier berganda pada dataset QoS. Berikut hasilnya: plot sebaran yang menunjukkan nilai datarate aktual vs prediksi dengan interval prediksi dapat diperoleh;



Persamaan regresi

```
Datarate = 4290574,68 + (-69119,65 × speed_kmh) + (2754792,53 × PCell_RSRP_1) + (-8523671,16 × PCell_RSRQ_1) + (-2415210,47 × PCell_RSSI_1) + (1506705,94 × PCell_SNR_1)
```

Koefisien dan statistik model terperinci:

	coef	std err	t	P>|t|	[0.025	0.975]
const	4.291e+06	5.1e+06	0.841	0.400	-5.71e+06	1.43e+07
speed_kmh	-6.912e+04	8046.338	-8.590	0.000	-8.49e+04	-5.33e+04
PCell_RSRP_1	2.755e+06	2.29e+05	12.015	0.000	2.31e+06	3.2e+06
PCell_RSRQ_1	-8.524e+06	2.36e+05	-36.137	0.000	-8.99e+06	-8.06e+06
PCell_RSSI_1	-2.415e+06	2.27e+05	-10.642	0.000	-2.86e+06	-1.97e+06
PCell_SNR_1	1.507e+06	3.13e+04	48.178	0.000	1.45e+06	1.57e+06

Hasil analisis:

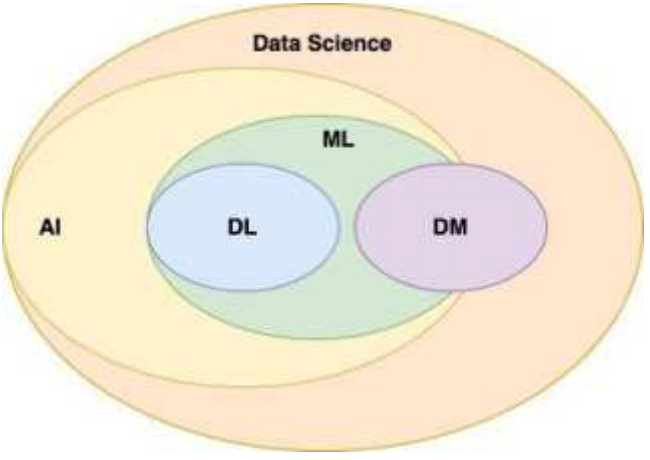
1) Semua prediktor signifikan secara statistik (nilai-p < 0,05)
2) Hubungan positif terkuat adalah dengan PCell_RSRP_1 dan PCell_SNR_1
3) Terdapat hubungan negatif dengan speed_kmh, PCell_RSRQ_1, dan PCell_RSSI_1
4) Area yang diarsir abu-abu dalam plot mewakili interval prediksi 95%
5) Garis putus-putus merah mewakili prediksi sempurna (di mana nilai aktual sama dengan nilai prediksi)

Model menunjukkan bahwa metrik kualitas sinyal (RSRP, RSRQ, RSSI, dan SNR) memiliki efek substansial pada kecepatan data, sementara kecepatan memiliki dampak negatif yang relatif lebih kecil.

Bab 11
Penerapan Deep Learning dalam Data Science

11.1 Pendahuluan Deep Learning

Deep Learning (DL) merupakan salah satu cabang dari Machine Learning yang menggunakan jaringan saraf tiruan (Artificial Neural Network) dengan banyak lapisan dan kemampuan untuk mengekstraksi pola dan fitur dari data secara otomatis (Suyanto et al., 2019) Click or tap here to enter text.. Deep Learning disebut Deep Neural Network karena jaringan ini memiliki jumlah lapisan yang tidak terbatas (Nourbakhsh & Habibi, 2023). Teknologi ini telah membawa revolusi dalam berbagai bidang, mulai dari pengenalan gambar, pemrosesan bahasa alami, hingga kendaraan otonom. Dalam data science, deep learning banyak digunakan dalam berbagai bidang, termasuk pemrosesan gambar, teks, suara, dan data numerik. Untuk menunjukkan posisi Deep Learning dalam Data Science dengan lebih jelas, dapat ditunjukkan pada Gambar 11.1 berikut:



Gambar 11.1: Posisi Deep Learning dalam Data Science

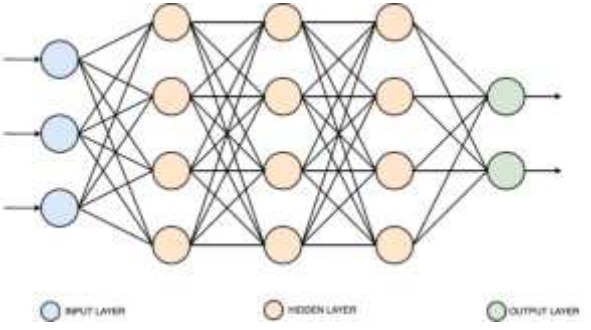
(Dr. Eng. Said Mirza Pahlevi, 2023)

Berbeda dengan algoritma Machine Learning tradisional yang sering memerlukan rekayasa fitur secara manual, Deep Learning mampu mempelajari fitur-fitur yang relevan secara langsung dari data. Hal ini dimungkinkan karena Deep Learning memiliki arsitektur berlapis yang secara hierarkis dapat menangkap representasi dari data input, dari pola sederhana hingga kompleks.

Deep Learning membutuhkan daya komputasi yang lebih besar dibandingkan dengan Machine Learning. Deep Learning melibatkan perhitungan matematika yang sangat kompleks, terutama dalam operasi matriks dan vektor yang digunakan dalam forward propagation, backpropagation, dan optimasi model. Graphics Processing Unit (GPU) memungkinkan pelatihan model lebih cepat dibandingkan CPU karena memiliki ribuan core yang dapat menangani operasi matematika secara paralel. GPU adalah unit pemrosesan khusus yang dirancang untuk menangani operasi komputasi yang bersifat paralel, terutama dalam rendering grafis. Namun, dalam perkembangannya, GPU juga banyak digunakan untuk berbagai tugas komputasi intensif di luar grafis, termasuk Deep Learning, simulasi ilmiah, dan analisis data besar (Big Data). Dalam proses pelatihan model berbasis Deep Learning, penggunaan GPU dapat mempercepat komputasi dan meningkatkan efisiensi pemrosesan data.

11.2 Arsitektur Deep Learning

Deep Learning menggunakan struktur jaringan yang disebut dengan Deep Neural Network (DNN) (Dr. Eng. Said Mirza Pahlevi, 2023). Dalam jaringan ini, informasi mengalir melalui lapisan input, lapisan tersembunyi (hidden layers), dan lapisan output seperti tampak pada Gambar 11.2 berikut:



Gambar 11.2 Jaringan Deep Learning

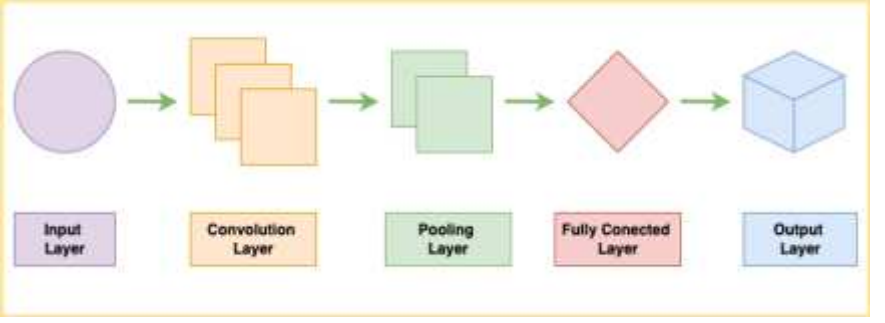
Seiring berkembangnya teknologi, berbagai arsitektur jaringan telah dikembangkan untuk menangani jenis data yang berbeda seperti gambar, teks, dan data sekuensial. Beberapa arsitektur Deep Learning yang paling umum digunakan meliputi Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU), Transformer, dan Generative Adversarial Networks (GAN). Setiap arsitektur memiliki karakteristik unik yang membuatnya cocok untuk berbagai tugas dalam Data Science.

11.2.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah jenis jaringan saraf yang sangat efektif dalam menangani data berbasis gambar dan video. CNN dirancang untuk mengenali pola dalam data spasial dengan menggunakan operasi konvolusi.

CNN dapat menyederhanakan kepadatan piksel dalam jumlah besar, sehingga sangat mengurangi jumlah parameter yang harus digunakan. CNN merepresentasikan input sebagai matriks yang digunakan untuk menghasilkan lapisan konvolusional pertama (Olorunnimbe & Viktor, 2023).

CNN terinspirasi oleh proses-proses biologi dimana pola konektivitas antar neuron menyerupai organisasi visual cortex pada binatang (Suyanto et al., 2019). CNN terdiri dari beberapa lapisan utama, yaitu lapisan konvolusi, aktivasi, pooling, dan fully connected. Arsitektur CNN tampak seperti pada Gambar 11.3 berikut:



Gambar 11.3 Arsitektur CNN

Lapisan konvolusi berfungsi sebagai ekstraktor fitur dengan menerapkan filter kecil yang bergerak melintasi data input untuk menangkap fitur lokal. Hasil konvolusi ini menghasilkan feature map yang merepresentasikan pola spesifik, seperti tepi atau tekstur pada gambar. Setelah konvolusi, lapisan aktivasi seperti

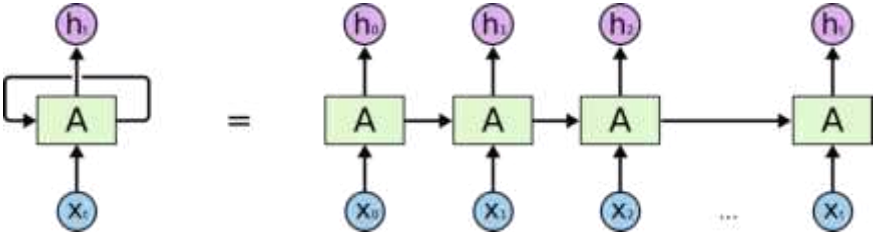
ReLU (Rectified Linear Unit) diterapkan untuk memperkenalkan non-linearitas ke dalam model, memungkinkan CNN mempelajari pola yang lebih kompleks. Selanjutnya, lapisan pooling digunakan untuk mengurangi ukuran feature map, mempertahankan informasi penting sekaligus mengurangi jumlah parameter dan kompleksitas komputasi.

Lapisan fully connected pada bagian akhir menghubungkan semua neuron untuk menggabungkan fitur-fitur yang telah diekstraksi dan melakukan prediksi akhir, seperti klasifikasi atau regresi. CNN memiliki beberapa kelebihan, yaitu efisiensi parameter yang tinggi, kemampuan mengenali pola di berbagai posisi atau orientasi, dan ekstraksi fitur otomatis. Namun, CNN juga memiliki kelemahan, seperti kebutuhan data yang besar dan komputasi yang intensif, serta kurang optimal dalam menangkap dependensi jangka panjang pada data time series. Meskipun begitu, CNN tetap menjadi pilihan utama dalam berbagai aplikasi seperti pengolahan citra, analisis sinyal, pemrosesan bahasa alami, dan prediksi time series karena kemampuannya dalam menangkap pola penting pada data yang memiliki struktur spasial dan temporal.

11.2.2 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) dirancang untuk menangani data sekuensial seperti teks, ucapan, dan data berbasis waktu. Berbeda CNN yang memproses data secara independen, RNN memiliki loop internal yang memungkinkan informasi sebelumnya digunakan untuk mempengaruhi prediksi saat ini. RNN memiliki mekanisme umpan balik yang memungkinkan informasi mengalir kembali ke dalam jaringan. Dengan demikian, RNN dapat memanfaatkan urutan data untuk membentuk memori, yang membantu dalam memahami hubungan antara data saat ini dan sebelumnya dalam suatu urutan (Beniwal et al., 2024).

RNN menggunakan koneksi loop untuk menyimpan informasi yang dihasilkan oleh suatu neuron yang akan digunakan untuk proses selanjutnya. Sehingga untuk proses selanjutnya selain data saat itu, informasi loop tersebut juga digunakan. Untuk lebih jelasnya, perhatikan pada Gambar 1.4 berikut:



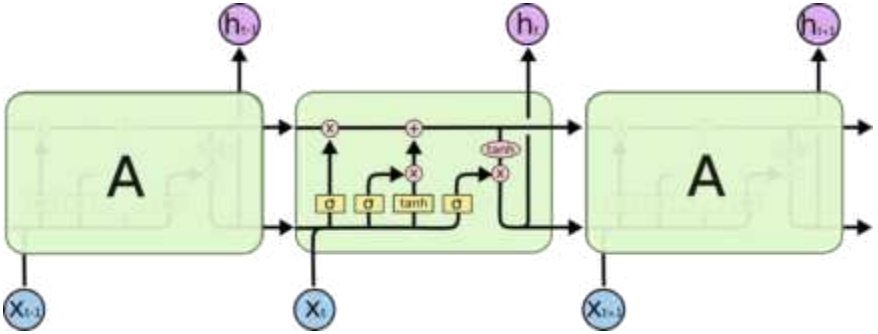
Gambar 11.4 Arsitektur RNN

(Sumber: https://colah.github.io/posts/2015-08-Understanding-LSTMs)

11.2.3 Long Short-Term Memory (LSTM)

LSTM pertama kali diusulkan pada tahun 1997 oleh Sepp Hochreiter dan Jurgen Schmidhuber, dimana LSTM merupakan pengembangan dari jenis RNN yang khusus dirancang untuk mengatasi masalah vanishing gradient pada RNN (Dr. Suyanto, Kurniawan Nur Ramadhani and Satria Mandala, 2019). Masalah vanishing gradient sering terjadi pada RNN saat menangani data dengan dependensi jangka panjang.

LSTM sangat cocok untuk tugas-tugas prediksi deret waktu, karena kemampuannya dalam mengingat informasi jangka panjang dan jangka pendek secara bersamaan. LSTM mampu menyimpan informasi dalam jangka waktu yang panjang, secara selektif mengabaikan data yang tidak diperlukan, serta menyesuaikan isi berdasarkan masukan terbaru (Zhang et al., 2024). Arsitektur LSTM tampak seperti pada Gambar 1.5 berikut:



Gambar 11.5 Arsitektur LSTM

(Sumber: https://colah.github.io/posts/2015-08-Understanding-LSTMs)

Struktur algoritma LSTM terdiri dari beberapa cell. State dari cell (Xt) dan hidden state (ht) akan diteruskan ke cell berikutnya. LSTM memiliki tiga jenis gate, yaitu: input gate, foreget gate, dan output gate (Singh et al., 2023). LSTM memiliki memori jangka panjang dalam bentuk bobot dan bobot berubah selama pelatihan. LSTM juga memiliki memori jangka pendek dalam bentuk aktivasi sementara, yang berpindah dari setiap node ke node yang berurutan. Model LSTM memperkenalkan jenis penyimpanan perantara melalui sel memori. Sebuah sel memori adalah unit komposit yang terdiri atas node sederhana dalam pola konektivitas tertentu dengan melibatkan node multiplikasi.

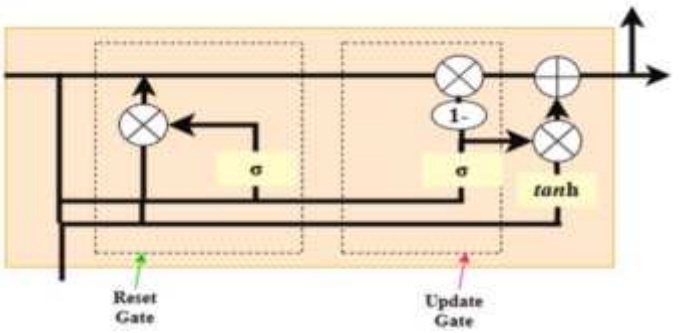
Input gate dalam LSTM adalah salah satu dari tiga gerbang utama yang mengatur aliran informasi ke dalam sel LSTM. Fungsi utama dari input gate adalah untuk mengontrol bagaimana informasi baru diperbarui atau disimpan dalam state sel. Input gate bertanggung jawab untuk menentukan seberapa banyak informasi baru dari input dan hidden state sebelumnya yang akan diperbarui dalam cell state. Input gate dalam LSTM sangat penting karena mengatur proses pembaruan informasi baru dalam cell state. Dengan mengontrol aliran informasi ini, input gate membantu LSTM untuk menyimpan informasi yang relevan dan membuang yang tidak relevan, yang sangat berguna dalam menangani urutan data yang panjang dan kompleks.

Fungsi utama dari forget gate adalah untuk menentukan seberapa banyak informasi dari cell state sebelumnya yang akan dilupakan atau dipertahankan pada cell state saat ini. Forget gate bertugas untuk mengontrol penghapusan informasi yang tidak lagi relevan dari cell state berdasarkan input saat ini dan hidden state sebelumnya. Forget gate dalam LSTM sangat penting karena ia mengatur penghapusan informasi yang tidak lagi relevan dari cell state. Dengan kemampuan ini, forget gate membantu LSTM untuk menangani urutan data yang panjang dan memastikan bahwa informasi penting tetap ada sementara informasi yang tidak relevan dihapus.

Fungsi utama dari output gate adalah untuk menentukan bagian mana dari cell state yang akan digunakan untuk menghasilkan hidden state yang baru. Hidden state ini kemudian digunakan sebagai output LSTM pada langkah waktu tersebut dan juga diteruskan ke langkah waktu berikutnya. Output gate dalam LSTM sangat penting karena ia mengatur informasi yang diambil dari cell state dan diproyeksikan keluar sebagai hidden state. Dengan kemampuan ini, output gate membantu LSTM untuk menentukan informasi relevan yang akan diteruskan ke langkah waktu berikutnya atau digunakan sebagai output saat ini.

11.2.4 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) adalah salah satu arsitektur RNN yang diperkenalkan oleh Kyunghyun Cho pada tahun 2014 sebagai penyempurnaan dari RNN tradisional dan alternatif yang lebih sederhana LSTM (Beniwal et al., 2024). GRU dirancang untuk menangani masalah utama yang sering dihadapi oleh RNN, yaitu vanishing gradient, yang menyebabkan kesulitan dalam mempelajari hubungan jangka panjang pada data sekuensial. Arsitektur GRU tampak seperti pada Gambar 1.6 berikut:



Gambar 11.6 Arsitektur GRU (Beniwal, Singh and Kumar, 2024)

Dalam model GRU, aliran informasi diatur melalui dua gerbang utama: reset gate dan update gate (Beniwal et al., 2024). Reset gate berfungsi untuk mengatur seberapa besar pengaruh dari informasi masa lalu yang akan dipertahankan atau dilupakan saat memproses informasi baru. Jika reset gate ini rendah, model akan mengabaikan sebagian besar informasi dari langkah sebelumnya. Hal ini penting dalam menangani situasi di mana informasi masa lalu tidak lagi relevan dengan kondisi terkini. Sebaliknya, ketika reset gate bernilai tinggi, informasi dari masa lalu tetap berpengaruh besar terhadap keluaran jaringan.

Selain reset gate, GRU juga memiliki update gate, yang bertanggung jawab untuk mengatur bagaimana hidden state (representasi internal jaringan) diperbarui. Update gate memutuskan seberapa banyak hidden state dari langkah sebelumnya yang akan dipertahankan dan digabungkan dengan hidden state baru. Jika update gate bernilai tinggi, jaringan akan lebih mempertahankan ingatan lama, seolah-olah tidak terjadi banyak perubahan dalam sistem. Namun, jika update gate rendah, hidden state akan lebih banyak diperbarui oleh input baru, yang memungkinkan jaringan mempelajari pola baru dalam data.

11.2.5 Transformer

Mekanisme utama dalam Transformer adalah self-attention, yang memungkinkan model untuk fokus pada bagian penting dalam urutan input tanpa harus memprosesnya secara berurutan. Ini menjadikan Transformer jauh lebih efisien dibandingkan dengan RNN. Tidak seperti jaringan berulang, transformer tidak mengalami masalah hilangnya gradien dan dapat mengakses informasi dari titik mana pun di masa lalu tanpa terpengaruh oleh jarak antar kata.

Arsitektur transformer terdiri dari dua komponen utama, yaitu encoder dan decoder. Bagian encoder berisi serangkaian lapisan yang mengkodekan data masukan berdasarkan pola tertentu, sementara decoder bertugas mendekode hasil encoding untuk menghasilkan keluaran yang diinginkan. Komponen kunci dalam encoder adalah mekanisme self-attention multi-head, yang memungkinkan model menangkap ketergantungan jangka panjang maupun pendek. Dengan perhatian yang berfokus pada berbagai aspek pola temporal, transformer dapat mengekstrak lebih banyak informasi fitur secara efektif (S. Wang, 2023).

11.2.6 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GAN) adalah arsitektur deep learning yang dirancang untuk menghasilkan data sintetis yang menyerupai data asli. GAN terbukti cukup adaptif dalam menghasilkan gambar baru berdasarkan kumpulan gambar yang digunakan selama pelatihan (Suyanto et al., 2019).

Model ini terdiri dari dua jaringan saraf: generator yang menghasilkan data palsu dan discriminator yang bertugas membedakan antara data asli dan palsu. Komponen Utama dalam GAN:

1. Generator (G)
    - Generator bertugas untuk membuat data baru yang menyerupai data asli.
    - Input Generator berupa noise acak (z) yang diambil dari distribusi probabilitas tertentu (misalnya distribusi normal).
    - Generator menggunakan jaringan saraf untuk mengubah noise menjadi data yang mirip dengan data asli.
    - Tujuan Generator adalah membuat data yang dapat menipu Discriminator.

2. Discriminator (D)
    - Discriminator bertugas untuk membedakan antara data asli dan data yang dihasilkan oleh Generator.
    - Input Discriminator bisa berasal dari: data asli dari dataset, data palsu dari Generator.
    - Discriminator dilatih untuk mengklasifikasikan inputnya sebagai "asli" atau "palsu".
    - Tujuan Discriminator adalah memperbaiki kemampuannya dalam mendeteksi data palsu.

Dengan proses pembelajaran berbasis persaingan antara kedua jaringan ini, GAN dapat digunakan dalam berbagai aplikasi, termasuk pembuatan gambar realistis, deepfake, dan augmentasi data.

11.3 Implementasi Deep Learning dalam Data Science

Deep Learning telah menjadi salah satu teknologi utama dalam Data Science, memungkinkan analisis data yang lebih dalam dan akurat dibandingkan dengan metode tradisional. Implementasi Deep Learning dalam Data Science mencakup pengolahan data besar (Big Data), pemrosesan gambar, analisis teks, dan prediksi berbasis pola. Dengan perkembangan komputasi seperti GPU dan TPU, model Deep Learning kini lebih cepat dan lebih efisien dalam menangani berbagai tantangan analisis data.

Untuk mengimplementasikan Deep Learning dalam Data Science, beberapa framework dan library telah dikembangkan untuk memudahkan pengolahan data dan pelatihan model:

1. TensorFlow
    - Dikembangkan oleh Google, sangat populer dalam riset dan produksi.
    - Mendukung CPU, GPU, dan TPU untuk pelatihan model skala besar.
2. PyTorch
    - Dikembangkan oleh Facebook, memiliki fleksibilitas tinggi dan mudah digunakan.
    - Banyak digunakan dalam penelitian akademik maupun industri karena eksekusi dinamisnya, yang memungkinkan pengembang untuk mendebug dan bereksperimen dengan model lebih cepat dibandingkan framework lain.

3. Keras
    - API tinggi berbasis TensorFlow untuk kemudahan pengembangan model Deep Learning.
    - Pengguna dapat membangun model deep learning dengan beberapa baris kode sederhana tanpa harus memahami detail implementasi matematis yang kompleks.

Implementasi Deep Learning membutuhkan perangkat keras yang mampu menangani komputasi intensif, terutama untuk melatih model skala besar.

1. Central Processing Unit (CPU)
    - Digunakan dalam tahap preprocessing data dan inferensi model ringan.
    - Kurang efisien untuk pelatihan model Deep Learning skala besar.
2. Graphics Processing Unit (GPU)
    - NVIDIA CUDA GPU adalah standar dalam pelatihan model Deep Learning.
    - Mempercepat operasi matriks dan tensor, penting dalam backpropagation.
3. Tensor Processing Unit (TPU)
    - Dikembangkan oleh Google khusus untuk mempercepat pelatihan model Deep Learning.
    - Digunakan dalam Google Colab dan Google Cloud AI.
4. Edge AI dan TinyML
    - Implementasi Deep Learning di perangkat kecil seperti Raspberry Pi, Jetson Nano, dan microcontrollers.
    - Digunakan dalam IoT, pengenalan suara, dan analitik real-time.

Beberapa dataset umum yang digunakan dalam penelitian dan implementasi Deep Learning dalam Data Science:

1. Dataset untuk Computer Vision
    - MNIST
      Klasifikasi angka tulisan tangan.
    - CIFAR-10 / CIFAR-100
      Klasifikasi gambar objek umum.
    - ImageNet
      Dataset skala besar untuk pengenalan objek.
    - COCO (Common Objects in Context)
      Digunakan dalam deteksi dan segmentasi objek.

2. Dataset untuk Natural Language Processing (NLP)
    - IMDB Dataset
      Analisis sentimen.
    - Wikipedia Corpus
      Digunakan untuk pre-training model NLP seperti BERT dan GPT.
    - SNLI (Stanford Natural Language Inference)
      Digunakan untuk memahami hubungan antar teks.
3. Dataset untuk Prediksi Keuangan
    - Yahoo Finance, Alpha Vantage, Quandl
      Dataset harga saham dan pasar finansial.
    - Kaggle Cryptocurrency Dataset
      Digunakan untuk analisis dan prediksi harga Bitcoin dan mata uang kripto lainnya.
4. Dataset untuk Big Data dan IoT
    - Google BigQuery Public Datasets
      Dataset besar dari berbagai domain.
    - Sensor IoT Dataset (UCI Repository)
      Digunakan untuk analisis data sensor dalam smart city dan industri 4.0.

Contoh implementasi Deep Learning dalam berbagai bidang Data Science:

1. Computer Vision
   Deep Learning digunakan untuk memproses dan memahami gambar dan video.
   Contoh implementasi:
    - Pengenalan wajah (Face Recognition)
    - Deteksi kanker dari citra medis
    - Pengenalan objek dalam kendaraan otonom
2. Natural Language Processing (NLP)
   Deep Learning digunakan dalam pemrosesan teks dan bahasa alami.
   Contoh implementasi:
    - Penerjemahan bahasa otomatis.
    - Chatbot AI.
    - Analisis sentimen media sosial dan ulasan produk.
3. Keuangan dan Perdagangan (Financial Trading)
   Deep Learning digunakan untuk prediksi pasar saham dan analisis risiko.

Contoh implementasi:

- Prediksi harga saham menggunakan LSTM dan Transformer.
- Deteksi penipuan dalam transaksi keuangan menggunakan GAN.
- Analisis sentimen berita terhadap pergerakan harga saham.

4. Kesehatan dan Medis
   Deep Learning membantu diagnosis dan pengobatan berbasis AI.
   Contoh implementasi:
   - Deteksi penyakit dari gambar medis (MRI, CT Scan).
   - Analisis genomik untuk deteksi dini kanker.
   - Pembuatan obat baru menggunakan GAN dan Transformer.
5. Big Data dan Internet of Things (IoT)
   Deep Learning digunakan untuk mengolah data besar dari berbagai sensor dan perangkat IoT.
   Contoh implementasi:
   - Sistem smart city untuk pengaturan lalu lintas berbasis AI.
   - Prediksi konsumsi energi dalam industri menggunakan model Time-Series.
   - Analisis data sensor IoT untuk perawatan mesin prediktif.

Deep Learning telah mengubah cara Data Science bekerja dengan memungkinkan analisis lebih akurat dan otomatisasi yang lebih canggih. Dengan perkembangan teknologi dan ketersediaan sumber daya, implementasi Deep Learning dalam Data Science akan terus berkembang dan membawa inovasi di berbagai industri.

Bab 12
Alat dan Platform Untuk Data Science

12.1 Pengantar Alat dan Platform Data Science

Alat dan platform dalam data science memainkan peran yang sangat penting dalam mendukung seluruh siklus kerja data science, mulai dari pengumpulan data, pembersihan, eksplorasi, analisis, hingga implementasi model. Perkembangan teknologi selama dekade terakhir telah menghasilkan berbagai alat yang dirancang untuk memenuhi kebutuhan data scientist dengan tingkat efisiensi dan fleksibilitas yang tinggi.

Gambar 12.1:Alat dan Platform Untuk Data Science

Platform data modern tidak hanya menyediakan infrastruktur yang kuat tetapi juga menawarkan integrasi dengan layanan komputasi awan untuk mendukung skalabilitas dan kolaborasi (Tranquillin et al., 2023). Platform seperti ini memungkinkan tim data science untuk berkolaborasi secara lebih efektif dalam

proyek-proyek besar yang membutuhkan sumber daya komputasi yang signifikan.

Alat-alat data science dapat dikategorikan ke dalam beberapa kelompok berdasarkan fungsinya:

1. Pengumpulan dan Penyimpanan Data

   Platform seperti Apache Hadoop dan Apache Spark dirancang untuk menangani big data secara efisien. Hadoop menggunakan arsitektur penyimpanan terdistribusi, sedangkan Spark menawarkan kemampuan pemrosesan data dalam memori yang lebih cepat. Alat-alat ini sangat ideal untuk data yang tidak terstruktur atau semi-terstruktur, yang umum ditemukan dalam proyek data science (Muniasamy et al., 2024).

2. Pembersihan dan Transformasi Data

   Proses pembersihan dan transformasi data sering kali menggunakan alat seperti Pandas dan NumPy di Python. Pustaka ini menyediakan fungsi untuk manipulasi data, pengisian nilai yang hilang, dan normalisasi data, yang merupakan langkah krusial sebelum analisis data dilakukan (Idrissi, 2023).

3. Analisis dan Pemodelan Data

   Analisis data melibatkan penggunaan algoritma machine learning yang diimplementasikan melalui pustaka seperti Scikit-learn, TensorFlow, dan PyTorch. Pustaka ini mendukung pengembangan model berbasis jaringan saraf tiruan (neural networks) dan pemrosesan graf data, yang menjadi semakin populer dalam aplikasi berbasis AI (Song, 2024).

4. Visualisasi Data

   Alat seperti Tableau dan Matplotlib digunakan untuk visualisasi data yang efektif. Visualisasi yang baik membantu data scientist dalam mengidentifikasi pola dan tren di dalam data, serta menyampaikan hasil analisis kepada pemangku kepentingan dengan cara yang mudah dimengerti (Dinov, 2023).

5. Deployment dan Monitoring Model

   Setelah model dikembangkan, langkah berikutnya adalah mengimplementasikannya ke lingkungan produksi menggunakan platform seperti AWS SageMaker dan Google AI Platform. Alat ini penting untuk memantau kinerja model secara berkelanjutan dan memperbarui model seiring waktu jika diperlukan (Tranquillin et al., 2023).

Berikut adalah tabel yang merangkum fungsi dan alat yang digunakan dalam berbagai tahap proses data science:

Tabel 12.1: Fungsi dan Tools Data Science

No	Fungsi	Tools
1	Pengumpulan dan Penyimpanan Data	Apache Hadoop, Apache Spark (Muniasamy et al., 2024)
2	Pembersihan dan Transformasi Data	Pandas, NumPy (Idrissi, 2023)
3	Analisis dan Pemodelan Data	Scikit-learn, TensorFlow, PyTorch (Song, 2024)
4	Visualisasi Data	Tableau, Matplotlib (Dinov, 2023)
5	Deployment dan Monitoring Model	AWS SageMaker, Google AI Platform (Tranquillin et al., 2023)

Kemajuan pesat dalam teknologi data science, alat dan platform modern menawarkan solusi yang lebih canggih dan terintegrasi untuk setiap tahap dalam siklus kerja data science. Pemilihan alat yang tepat harus mempertimbangkan kebutuhan proyek, skalabilitas, dan kemudahan integrasi dengan sistem yang ada. Studi mendalam tentang alat ini, seperti yang dijelaskan dalam referensi terbaru, membantu memastikan bahwa data scientist memiliki pengetahuan dan keterampilan yang diperlukan untuk memanfaatkan teknologi terbaru secara efektif.

12.2 Platform Cloud untuk Data Science

latform cloud telah merevolusi cara kerja data science dengan menyediakan infrastruktur yang fleksibel dan skalabel untuk menangani berbagai proses data. Layanan cloud memungkinkan data scientist untuk mengakses sumber daya komputasi yang luas tanpa harus berinvestasi pada perangkat keras yang mahal. Platform ini juga mendukung kolaborasi secara real-time dan menyediakan solusi untuk pemrosesan data besar, analisis, dan implementasi model machine learning.

Layanan cloud dapat dikategorikan ke dalam beberapa jenis:

1. Infrastructure as a Service (IaaS)
   Menyediakan infrastruktur dasar seperti komputasi, penyimpanan, dan jaringan. Contoh: Amazon EC2 dan Google Compute Engine (Tranquillin et al., 2023).
2. Platform as a Service (PaaS)
   Menyediakan lingkungan pengembangan yang telah terintegrasi dengan alat-alat analitik dan layanan data science. Contoh: Google AI Platform dan Azure Machine Learning (Song, 2024).
3. Software as a Service (SaaS)
   Memberikan layanan perangkat lunak berbasis cloud yang dapat diakses secara langsung. Contoh: Google Sheets dan Tableau Online (Muniasamy et al., 2024).

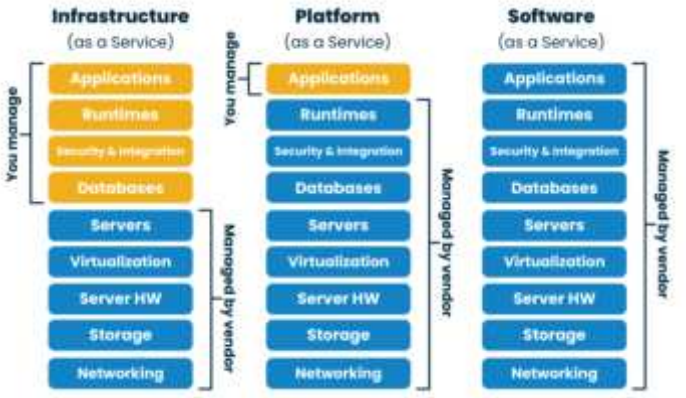
Cloud computing terbagi menjadi tiga model layanan utama yang menawarkan solusi untuk berbagai kebutuhan teknologi informasi: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), dan SaaS (Software as a Service). Setiap model memiliki keunggulan yang disesuaikan dengan kebutuhan pengguna, mulai dari penyediaan infrastruktur, pengembangan aplikasi, hingga layanan perangkat lunak siap pakai.

Tabel 12.2: Perbandingan Model Layanan Cloud Computing

Indikator	**IaaS (Infrastructure as a Service)**	**PaaS (Platform as a Service)**	**SaaS (Software as a Service)**
Kegunaan	Digunakan oleh arsitek jaringan.	Digunakan oleh developer.	Digunakan oleh end user.
Akses	Virtual machine dan virtual storage.	Lingkungan pengembangan aplikasi.	Akses langsung ke aplikasi siap pakai.
Model	Menyediakan sumber daya komputasi yang divisualisasikan melalui internet.	Memberikan alat untuk pengembangan aplikasi.	Menyediakan layanan komputasi awan melalui perangkat lunak host.
Pemahaman Teknis	Membutuhkan pengetahuan teknis tinggi.	Memerlukan pemahaman pengaturan dasar.	Tidak memerlukan pengetahuan teknis.

Indikator	IaaS (Infrastructure as a Service)	PaaS (Platform as a Service)	SaaS (Software as a Service)
Popularitas	Populer di kalangan pengembang dan peneliti.	Populer di kalangan pengembang aplikasi dan skrip.	Populer di kalangan konsumen dan perusahaan.
Layanan Cloud	Amazon Web Services, Sun, vCloud Express.	Facebook & Google search engine.	MS Office web, Facebook, Google apps.
Layanan Perusahaan	AWS virtual private cloud.	Microsoft Azure.	IBM cloud analysis.
Layanan Outsourced	Salesforce.	force.com, Gigaspaces.	AWS, Terremark.

Ketiga model layanan ini menawarkan keunggulan yang berbeda tergantung kebutuhan pengguna. IaaS memberikan kontrol penuh atas infrastruktur dan cocok untuk pengembang yang membutuhkan fleksibilitas tinggi. PaaS menyediakan lingkungan yang mendukung pengembangan dan pengujian aplikasi tanpa memerlukan pengelolaan infrastruktur. SaaS dirancang untuk pengguna akhir yang membutuhkan aplikasi siap pakai dengan kemudahan akses tanpa perlu pemeliharaan teknis.



Gambar 12.2: Perbedaan IaaS, PaaS dan SaaS
(Sumber: saptatunas.com)

Keunggulan utama platform cloud meliputi:

1. Skalabilitas: Kemampuan untuk menambah atau mengurangi kapasitas komputasi sesuai dengan kebutuhan (Tranquillin et al., 2023).
2. Aksesibilitas: Kemudahan akses dari berbagai lokasi dengan koneksi internet (Song, 2024).
3. Efisiensi Biaya: Penghematan biaya operasional dengan model pembayaran berdasarkan penggunaan (Dinov, 2023).
4. Keamanan Data: Fitur keamanan tingkat tinggi yang mencakup enkripsi data dan kontrol akses (Idrissi, 2023).

Studi kasus menunjukkan bahwa perusahaan yang mengadopsi platform cloud untuk proyek data science dapat mempercepat proses pengembangan model dan meningkatkan ketepatan hasil analisis. Platform seperti AWS SageMaker dan Google Cloud AI Platform telah menjadi standar industri untuk membangun, melatih, dan menerapkan model machine learning di lingkungan produksi (Tranquillin et al., 2023).

12.3 Tools Data Science

Alat dalam data science dapat diklasifikasikan menjadi dua kategori utama: open source dan komersial. Masing-masing memiliki keunggulan dan kelemahan yang perlu dipertimbangkan berdasarkan kebutuhan proyek dan sumber daya yang tersedia.

12.3.1 Python

Python dikenal karena sintaksnya yang sederhana dan mudah dibaca, menjadikannya pilihan populer di kalangan pemula maupun profesional. Dalam konteks data science, Python menawarkan fleksibilitas tinggi dan dukungan ekosistem pustaka yang luas.

Tabel 12.3: Alat-alat Pada Python

Kategori	Alat	Fungsi	Fitur Utama
Pengolahan dan Analisis Data	Pandas	Manipulasi dan analisis data terstruktur.	• Membaca dan menulis format seperti CSV, Excel, SQL, dan JSON. • Filtering, grouping, dan penggabungan dataset. • Pengolahan data waktu.

Kategori	Alat	Fungsi	Fitur Utama
	NumPy	Komputasi numerik dan operasi matriks tingkat tinggi.	• Dukungan array multidimensi. • Operasi matematika seperti linear algebra dan transformasi Fourier. • Integrasi dengan TensorFlow dan Scikit-learn.
Machine Learning dan AI	Scikit-learn	Algoritma machine learning klasik seperti regresi, klasifikasi, dan klastering.	• Algoritma siap pakai dengan API yang mudah digunakan. • Pipeline machine learning. • Evaluasi model dan validasi silang.
	TensorFlow dan Keras	Pengembangan model deep learning yang kompleks dengan dukungan GPU dan komputasi terdistribusi.	• Melatih jaringan saraf tiruan (neural networks). • Transfer learning untuk data kecil. • API TensorFlow Serving untuk implementasi produksi.
Visualisasi Data	Matplotlib	Pembuatan grafik 2D yang mendetail dan fleksibel.	Membuat grafik statis dan dinamis dengan kontrol tinggi.
	Seaborn	Visualisasi statistik yang elegan dan informatif.	Grafis statistik tingkat tinggi dengan integrasi ke Pandas.
	Plotly	Visualisasi data interaktif dan integrasi dengan dashboard.	Grafik interaktif yang mendukung aplikasi berbasis web dengan Dash.
Automasi dan Pipeline	Apache Airflow	Orkestrasi pipeline data yang kompleks dan proses machine learning.	Penjadwalan dan monitoring alur kerja data secara otomatis.
	Luigi	Automasi dan pipeline data yang fleksibel.	Dukungan untuk pipeline modular dan dependensi antar tugas.
	MLflow	Manajemen eksperimen dan deployment model machine learning.	Pelacakan eksperimen, model registry, dan deployment otomatis.

12.3.2 R

Bahasa pemrograman R memiliki keunggulan dalam analisis statistik dan visualisasi data. Dengan ekosistem pustaka yang kaya, R mendukung berbagai tugas data science, mulai dari pengolahan data hingga pengembangan model machine learning dan integrasi dengan pipeline otomatis.

Gabungan antara alat seperti ggplot2 untuk visualisasi yang mendalam, caret untuk machine learning, dan Shiny untuk dashboard interaktif menjadikan R pilihan utama untuk aplikasi yang memerlukan analisis statistik yang kuat dan komunikasi hasil yang efektif. Berikut adalah tabel yang merangkum fitur utama alat-alat data science berbasis R:

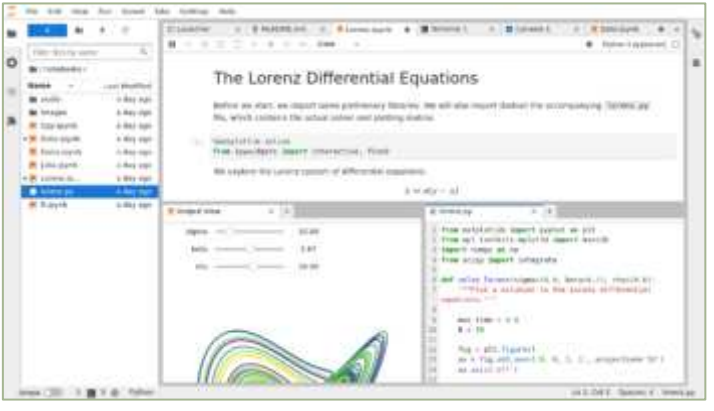
Tabel 12.4: Fitur utama alat-alat data science berbasis R

Kategori	Alat	Fungsi	Fitur Utama
Pengolahan dan Analisis Data	dplyr	Manipulasi dan transformasi data.	• Filtering, penggabungan, dan pengelompokan data. • Operasi data pipeline yang efisien dan mudah dibaca.
	tidyr	Merapikan dan membersihkan data.	• Mengubah data dari format lebar ke panjang (pivoting). • Mengatasi data yang hilang atau tidak terstruktur.
	data.table	Pengolahan data besar dengan kinerja tinggi.	• Operasi tabel data yang sangat cepat untuk kumpulan data besar. • Mendukung penggabungan, filtering, dan transformasi dalam satu sintaks singkat.
Statistik dan Machine Learning	caret	Pengembangan dan evaluasi model machine learning.	• Mendukung berbagai algoritma machine learning. • Pengaturan parameter dan validasi silang yang mudah.
	mlr	Pipeline machine learning dan optimasi model.	• Dukungan untuk ensemble learning dan optimasi hyperparameter. • Alat untuk validasi dan pelaporan hasil model.
	randomForest	Model prediktif berbasis pohon keputusan.	• Mendukung klasifikasi dan regresi dengan algoritma random forest. • Penanganan data berdimensi tinggi.
	xgboost	Gradient boosting untuk prediksi akurat.	• Optimal untuk data besar dan model kompleks. • Dukungan untuk tuning hyperparameter dan ensemble.
Visualisasi Data	ggplot2	Visualisasi data yang fleksibel dan berkualitas tinggi.	• Grafis berbasis lapisan yang mudah dikustomisasi.

Kategori	Alat	Fungsi	Fitur Utama
			• Dukungan untuk berbagai tipe grafik (scatter plot, bar chart, heatmap, dll.).
	plotly	Visualisasi data interaktif.	Grafik dinamis dan interaktif yang dapat diintegrasikan dengan dashboard.
	Shiny	Pengembangan aplikasi web interaktif berbasis data.	Alat untuk membangun dashboard dan visualisasi data yang dapat diakses secara online.
Automasi dan Pipeline	drake	Automasi pipeline analisis data.	Alat untuk mengelola ketergantungan tugas dan memastikan reproduktibilitas.
	plumber	Mengintegrasikan model analitik sebagai API web.	Membuat RESTful API dari kode R dengan mudah.
	workflowr	Mengelola proyek data science yang reproducible.	Dukungan untuk kontrol versi dan dokumentasi otomatis proyek.

12.3.3 Jupyter Notebook

Jupyter Notebook merupakan alat pengembangan interaktif yang dirancang untuk mendukung proses analisis data, pengembangan model machine learning, dan dokumentasi hasil penelitian secara terintegrasi. Alat ini memainkan peran penting dalam ekosistem data science modern karena kemampuannya untuk menggabungkan kode, visualisasi, dan penjelasan deskriptif dalam satu dokumen yang mudah dibaca dan dibagikan (Grus, 2019).



Gambar 12.3: Tampilan Jupyter Notebook

Fungsi utama Jupyter Notebook, sebagai berikut:

1. Pengembangan Interaktif dan Eksperimen

   Jupyter Notebook memungkinkan eksekusi kode secara bertahap melalui pendekatan berbasis sel (cells). Fitur ini mendukung proses pengujian iteratif, di mana pengguna dapat menjalankan potongan kode, memeriksa hasil, dan menyempurnakan logika tanpa perlu menjalankan seluruh skrip dari awal (Kluyver et al., 2016). Selain itu, Jupyter Notebook juga mendukung visualisasi langsung di dalam notebook, memungkinkan data scientist untuk menganalisis dan memvisualisasikan data dengan lebih efektif menggunakan pustaka seperti Matplotlib dan Seaborn (McKinney, 2017).

2. Prototipe Model Machine Learning

   Jupyter Notebook sering digunakan sebagai alat prototipe untuk eksperimen machine learning. Pustaka seperti Scikit-learn dan TensorFlow dapat dengan mudah diintegrasikan untuk membangun dan menguji model prediktif. Model ini kemudian dapat dioptimalkan dan dievaluasi secara langsung di dalam notebook menggunakan metrik evaluasi seperti akurasi, precision, dan recall (Chollet, 2018).

3. Dokumentasi dan Kolaborasi

   Salah satu keunggulan utama Jupyter Notebook adalah kemampuannya dalam menggabungkan kode dengan teks deskriptif yang ditulis dalam format Markdown. Hal ini memungkinkan penulisan catatan terstruktur yang menjelaskan proses analisis atau eksperimen yang sedang dijalankan (Sharma, 2021). Dokumen yang dihasilkan juga dapat diekspor ke berbagai format, termasuk PDF, HTML, dan LaTeX, untuk keperluan presentasi atau publikasi ilmiah. Selain itu, Jupyter Notebook mendukung kolaborasi melalui integrasi dengan platform berbasis cloud seperti Google Colab dan Microsoft Azure Notebooks (Carneiro et al., 2020).

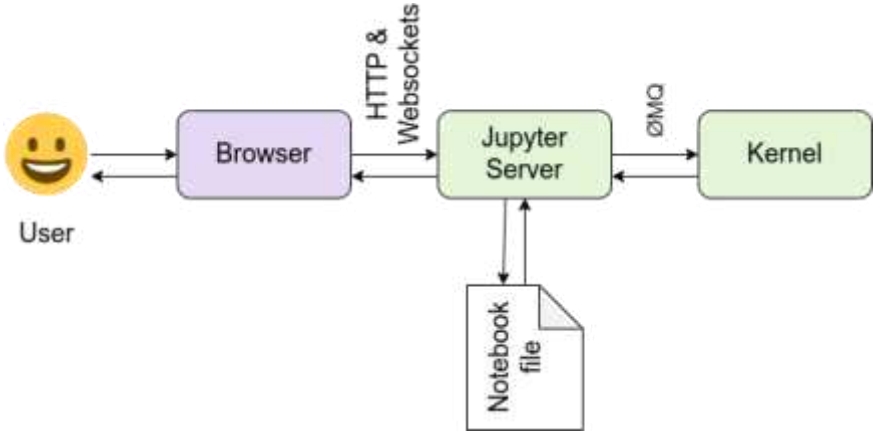
4. Integrasi dengan Pustaka Data Science

   Jupyter Notebook menyediakan dukungan yang kuat untuk pustaka analisis data seperti Pandas dan NumPy. Dengan Pandas, pengguna dapat dengan mudah mengelola dan memanipulasi dataset yang besar menggunakan struktur data berbasis tabel (McKinney, 2017). Sementara itu, NumPy menyediakan alat untuk komputasi numerik yang efisien dan manipulasi array multidimensi (Van Der Walt et al., 2011).

5. Ekosistem yang Luas dan Fleksibel

   Dengan komunitas pengguna yang besar dan dukungan ekosistem ekstensi, Jupyter Notebook telah berkembang menjadi alat yang sangat fleksibel. Ekstensi seperti JupyterLab memperluas fungsionalitas notebook dengan fitur tambahan seperti manajemen file, integrasi terminal, dan dukungan multi-tab (Granger & Pérez, 2020). Selain itu, alat ini mendukung lebih dari 40 bahasa pemrograman, termasuk Python, R, dan Julia (Kluyver et al., 2016).

Jupyter Notebook memiliki arsitektur yang terdiri dari beberapa komponen utama yang bekerja secara terintegrasi untuk memberikan pengalaman pengembangan yang interaktif dan fleksibel. Setiap komponen memiliki peran khusus dalam mengelola data, menjalankan kode, dan menampilkan hasil analisis. Tabel berikut merangkum fungsi dan peran masing-masing komponen dalam ekosistem Jupyter Notebook.



Gambar 12.4: Arsitektur Jupyter Notebook
(Sumber: docs.jupyter.org)

Tabel 12.5:Komponen Utama dalam Jupyter Notebook

Komponen	Deskripsi
User (Pengguna)	Mengakses Jupyter Notebook melalui antarmuka berbasis web menggunakan browser untuk menulis, menjalankan, dan mendokumentasikan kode.
Browser	Bertindak sebagai antarmuka pengguna (UI) yang memungkinkan interaksi dengan notebook. Mengirim dan menerima perintah melalui protokol HTTP atau WebSocket.

Komponen	Deskripsi
Jupyter Server	Mengelola sesi pengguna, menangani permintaan dari browser, dan berfungsi sebagai jembatan antara antarmuka pengguna dan kernel eksekusi kode.
Kernel	Proses backend yang menjalankan kode. Mendukung berbagai bahasa pemrograman seperti Python, R, dan Julia. Menggunakan protokol ZeroMQ (ØMQ) untuk komunikasi dengan server.
Notebook File (.ipynb)	Menyimpan kode, output, visualisasi, dan teks markdown dalam format JSON. Dapat diekspor ke format PDF, HTML, dan LaTeX untuk berbagi hasil.

12.3.4 Apache Spark

Apache Spark adalah framework komputasi terdistribusi yang dirancang untuk menangani pengolahan data dalam skala besar dengan efisiensi tinggi. Dikembangkan oleh Apache Software Foundation, Spark telah menjadi komponen utama dalam ekosistem big data modern. Kemampuannya untuk memproses data secara paralel dan mendukung analisis real-time menjadikannya alat yang sangat andal untuk data scientist dan engineer dalam mengelola dan menganalisis data besar (Zaharia et al., 2016).

Apache Spark dikenal karena kemampuannya dalam memproses data dengan kecepatan tinggi dan fleksibilitas yang luar biasa. Berikut adalah beberapa karakteristik utama yang membuatnya unggul:

1. Komputasi dalam Memori (*In-Memory Computing*)

   Salah satu keunggulan utama Spark adalah kemampuannya untuk memproses data langsung di dalam memori. Pendekatan ini mengurangi waktu yang diperlukan untuk membaca dan menulis data ke disk, sehingga mempercepat eksekusi hingga 100 kali lebih cepat dibandingkan Hadoop MapReduce (Karau & Warren, 2017).

2. Model Pemrograman yang Fleksibel

   Spark mendukung berbagai bahasa pemrograman populer seperti Python, Scala, Java, dan R. Ini memberikan kemudahan bagi pengguna dengan latar belakang yang berbeda untuk memanfaatkan kekuatan Spark tanpa harus mempelajari bahasa baru secara mendalam (Chambers & Zaharia, 2018).

3. Skalabilitas dan Distribusi

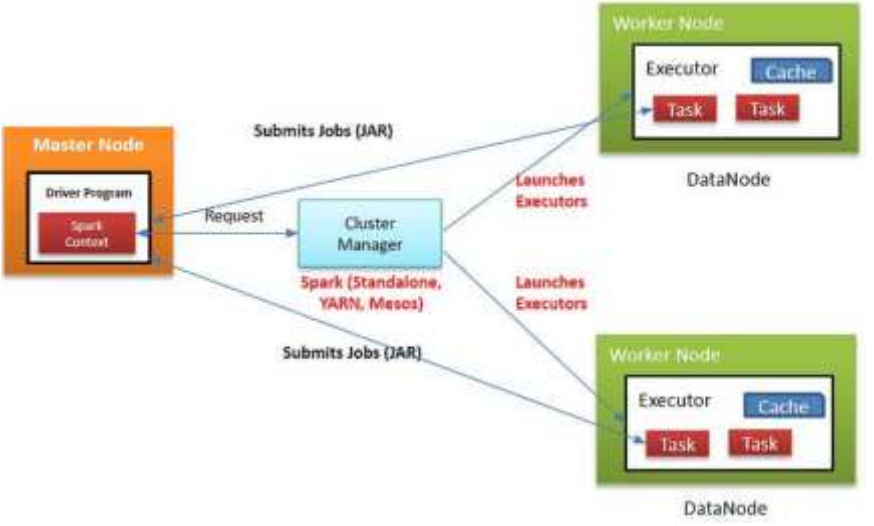
   Dirancang untuk menangani data dalam skala petabyte, Spark memungkinkan komputasi terdistribusi yang efisien di berbagai node dalam

cluster. Dengan integrasi ke Hadoop Distributed File System (HDFS) dan sistem penyimpanan cloud seperti Amazon S3, Spark menawarkan skalabilitas tinggi yang dibutuhkan untuk analisis data skala besar.

Apache Spark menyediakan berbagai library bawaan yang mendukung kebutuhan analitik yang beragam:

1. Spark SQL: Memungkinkan pengguna menjalankan kueri SQL untuk mengakses data terstruktur dan semi-terstruktur.
2. MLlib: Library untuk machine learning, termasuk algoritma untuk klasifikasi, regresi, dan klasterisasi.
3. GraphX: Alat untuk analisis graf yang digunakan dalam jaringan sosial dan pemetaan hubungan.
4. Spark Streaming: Mendukung pemrosesan data real-time dari sumber seperti Apache Kafka dan Amazon Kinesis (Zaharia et al., 2016).

Arsitektur Apache Spark dirancang untuk mendukung komputasi paralel yang terdistribusi. Komponen utama dalam arsitektur ini meliputi:



Gambar 12.5: Arsitektur Apache Spark

(Sumber: https://npntraining.medium.com/)

1. Driver Program

   Driver bertanggung jawab untuk mengatur eksekusi aplikasi Spark dan membuat SparkContext, yang menghubungkan program ke cluster manager.
2. Cluster Manager

   Mengatur sumber daya komputasi yang tersedia di dalam cluster. Spark mendukung berbagai jenis cluster manager, termasuk YARN, Apache Mesos, dan mode standalone.
3. Executor

   Executor adalah proses yang berjalan di setiap node dalam cluster untuk menjalankan tugas yang ditugaskan oleh driver program. Executor juga menyimpan data sementara di memori selama pemrosesan.
4. Directed Acyclic Graph (DAG) Scheduler

   Scheduler ini merancang alur kerja berbasis graf untuk memastikan bahwa tugas-tugas dapat dijalankan secara paralel dan optimal. DAG membantu mengidentifikasi dependensi antar tugas sehingga alur eksekusi dapat dimaksimalkan.

Apache Spark digunakan secara luas dalam berbagai skenario analitik, termasuk:

1. Analisis Data Besar (*Big Data Analytics*)

   Dengan integrasi ke HDFS dan Amazon S3, Spark memproses data besar dalam waktu yang lebih singkat dibandingkan framework lainnya.
2. Machine Learning

   Library MLlib menyediakan algoritma machine learning yang siap digunakan untuk klasifikasi, regresi, dan pengelompokan data. Contohnya, Spark digunakan dalam sistem rekomendasi dan deteksi penipuan di sektor keuangan.
3. Analisis Data Real-Time

   Melalui Spark Streaming, data yang diterima secara real-time dari sumber seperti Kafka dapat diproses dan dianalisis secara langsung untuk menghasilkan insight yang cepat dan relevan.
4. Analisis Graf

   Library GraphX memungkinkan pengguna untuk melakukan analisis jaringan kompleks, seperti pemetaan hubungan dalam jaringan sosial atau deteksi anomali dalam sistem komunikasi.

- TensorFlow dan PyTorch
  Framework deep learning yang mendukung pengembangan model machine learning canggih.
- KNIME
  Alat pengolahan data berbasis GUI yang menyediakan fitur integrasi alur kerja dengan berbagai alat lainnya.
- RapidMiner: Meskipun memiliki versi komersial, RapidMiner menyediakan versi open source untuk eksperimen analitik yang kuat.

12.3.5 MATLAB

MATLAB, atau singkatan dari MATrix LABoratory, adalah sebuah lingkungan pemrograman yang dirancang khusus untuk pemrosesan data numerik dan simulasi teknis. Dikembangkan oleh MathWorks, MATLAB telah menjadi standar industri dalam berbagai bidang, termasuk teknik, fisika, sains data, dan keuangan (Pratap, 2010). Dengan kemampuannya untuk memproses data berbasis matriks, membuat visualisasi canggih, dan mendukung skrip yang intuitif, MATLAB memberikan solusi yang komprehensif bagi pengguna yang membutuhkan analisis data teknis yang mendalam (Hanselman & Littlefield, 2011).

Gambar 12.6: Contoh Tampilan UI Matlab

(Sumber: mathworks.com)

MATLAB memiliki berbagai fitur unggulan yang menjadikannya alat pilihan dalam berbagai aplikasi analisis numerik dan simulasi teknis (Higham & Higham, 2016):

1. Pemrosesan Matriks yang Kuat

   MATLAB dirancang untuk bekerja dengan matriks secara efisien. Struktur datanya secara default berbasis matriks, sehingga mempermudah operasi matematis seperti aljabar linier, transformasi Fourier, dan analisis statistik. Hal ini menjadikannya alat yang sangat andal untuk bidang teknik dan analisis numerik.

2. Simulasi Sistem Dinamis dengan Simulink

   MATLAB dilengkapi dengan Simulink, sebuah platform pemodelan berbasis grafis yang memungkinkan pengguna untuk mensimulasikan sistem dinamis seperti kontrol mesin, desain robotika, dan simulasi mekanis. Dengan Simulink, pengguna dapat memodelkan sistem secara visual menggunakan blok diagram, tanpa perlu menulis kode kompleks.

3. Visualisasi Data yang Kaya

   MATLAB menyediakan alat visualisasi yang canggih untuk membantu pengguna memahami dan menganalisis data. Fitur ini mencakup:

   - Pembuatan grafik 2D dan 3D, seperti scatter plot, surface plot, dan heatmap.
   - Visualisasi interaktif untuk menyesuaikan grafik secara dinamis.
   - Kemampuan ekspor grafik ke berbagai format, termasuk PDF dan PNG, untuk kebutuhan pelaporan.

4. Library dan Toolbox yang Luas

   MATLAB menyediakan berbagai toolbox khusus untuk kebutuhan spesifik, seperti:

   - Optimization Toolbox: Untuk optimasi linear, non-linear, dan multiobjektif.
   - Statistics and Machine Learning Toolbox: Untuk analisis data prediktif dan pemodelan statistik.
   - Image Processing Toolbox: Untuk analisis citra dan pengolahan sinyal.
   - Deep Learning Toolbox: Untuk membangun dan melatih jaringan saraf tiruan.

5. Kemampuan Skrip yang Intuitif

   MATLAB menggunakan bahasa pemrograman yang sederhana dan mudah dipahami, sehingga cocok bagi pemula maupun profesional. Pengguna dapat menulis skrip untuk mengotomasi proses analisis data, memproses batch data, atau menjalankan eksperimen simulasi yang kompleks.

6. Integrasi dengan Bahasa Lain

   MATLAB mendukung integrasi dengan bahasa pemrograman lain seperti Python, C/C++, dan Java. Fitur ini memungkinkan pengguna untuk menggabungkan kemampuan MATLAB dengan alat lain yang sudah digunakan di lingkungan kerja mereka.

MATLAB adalah salah satu alat utama dalam berbagai disiplin ilmu dan industri yang membutuhkan analisis data numerik, simulasi sistem, serta pengembangan model pembelajaran mesin. Dengan fitur-fitur seperti Simulink untuk simulasi sistem dinamis, serta toolbox khusus untuk analisis statistik dan machine learning, MATLAB menyediakan solusi yang komprehensif untuk berbagai kebutuhan teknik dan data science (Attaway, 2016).

1. Analisis Numerik dan Optimasi

   MATLAB sering digunakan untuk menyelesaikan masalah matematis yang kompleks, seperti persamaan diferensial, analisis statistik, dan optimasi multiobjektif. Alat ini memberikan solusi numerik yang akurat dengan waktu pemrosesan yang singkat.

2. Desain dan Simulasi Sistem Dinamis

   Dengan bantuan Simulink, MATLAB menjadi alat utama dalam pemodelan sistem kontrol, simulasi kendaraan otonom, dan simulasi mekanika fluida. Misalnya, dalam industri otomotif, MATLAB digunakan untuk mendesain sistem pengereman otomatis atau model mesin.

3. Visualisasi dan Analisis Data

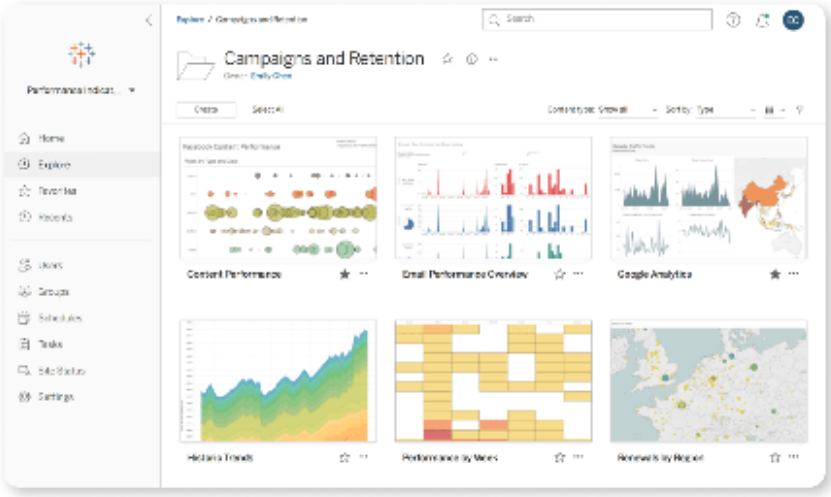
   MATLAB memungkinkan pengguna untuk mengolah data yang kompleks menjadi visualisasi yang mudah dipahami. Kemampuan ini sangat penting dalam penelitian dan presentasi hasil analisis kepada pemangku kepentingan.

4. Machine Learning dan Deep Learning

   Dengan dukungan toolbox khusus, MATLAB mendukung pengembangan model machine learning dan deep learning untuk klasifikasi, regresi, dan klasterisasi. MATLAB juga kompatibel dengan GPU untuk mempercepat pelatihan model.

12.3.6 Tableau

Tableau adalah salah satu alat visualisasi data terkemuka yang dirancang untuk mempermudah analisis data dan menyampaikan hasilnya dalam bentuk yang mudah dipahami. Dengan antarmuka pengguna yang intuitif dan kemampuan drag-and-drop, Tableau memungkinkan pengguna dengan latar belakang teknis maupun non-teknis untuk membuat grafik, dashboard, dan laporan yang menarik. Alat ini banyak digunakan di berbagai industri, termasuk keuangan, pemasaran, dan pendidikan, untuk membantu pengambilan keputusan berbasis data (Murray, 2020).



Gambar 12.7: Contoh Tampilan UI Tableau

(Sumber: tableau.com)

Tableau menawarkan berbagai fitur unggulan yang memudahkan pengguna dalam membuat visualisasi data secara interaktif dan intuitif (Tableau, 2023).

3. Antarmuka yang Mudah Digunakan
   Tableau dirancang untuk digunakan tanpa memerlukan keahlian pemrograman. Pengguna dapat dengan mudah mengimpor data dari berbagai sumber, seperti spreadsheet, database, atau layanan cloud, dan mulai membuat visualisasi dengan fitur drag-and-drop yang intuitif.
4. Kemampuan Integrasi dengan Berbagai Sumber Data
   Tableau mendukung integrasi dengan berbagai jenis data, termasuk:

- File lokal seperti Excel, CSV, dan JSON.
- Basis data seperti SQL Server, Oracle, dan MySQL.
- Layanan cloud seperti Google Sheets, Salesforce, dan Amazon Redshift (Jones, 2019).

5. Visualisasi Interaktif

   Tableau memungkinkan pengguna untuk membuat visualisasi interaktif, seperti: Grafik garis dan batang, Scatter plot, Heatmap, Peta geografis. Visualisasi ini dapat dikustomisasi dan dihubungkan dengan filter dinamis yang mempermudah eksplorasi data.

6. Kemampuan Membuat Dashboard dan Laporan

   Tableau memungkinkan pengguna untuk menggabungkan beberapa visualisasi ke dalam satu dashboard. Dashboard ini dapat diekspor ke berbagai format seperti PDF atau dipublikasikan secara online untuk kolaborasi tim (Stackowiak, 2020).

7. Komunitas dan Ekosistem yang Kuat

   Tableau memiliki komunitas pengguna yang besar dan aktif, serta menyediakan sumber daya seperti tutorial, forum, dan template dashboard untuk membantu pengguna baru memahami alat ini.

Tableau digunakan secara luas dalam berbagai sektor untuk memvisualisasikan data, membantu pengambilan keputusan yang lebih baik, dan menyajikan wawasan secara efektif melalui dashboard interaktif dan laporan yang menarik. Berikut adalah beberapa aplikasi Tableau di berbagai industri:

1. Keuangan

   Tableau digunakan untuk memvisualisasikan laporan keuangan, analisis risiko, dan tren pasar. Dashboard interaktif membantu tim keuangan memahami data secara lebih mendalam dan membuat keputusan yang lebih baik.

2. Pemasaran

   Dengan Tableau, tim pemasaran dapat memantau kinerja kampanye, menganalisis perilaku pelanggan, dan mengidentifikasi peluang pasar. Visualisasi data real-time memungkinkan tindakan cepat berdasarkan data yang akurat.

3. Pendidikan

   Tableau digunakan oleh institusi pendidikan untuk menganalisis data siswa, keberhasilan kurikulum, dan alokasi sumber daya.

4. E-commerce

   Dalam industri e-commerce, Tableau membantu dalam menganalisis tren pembelian, kinerja produk, dan perilaku pelanggan untuk meningkatkan pengalaman pengguna dan pendapatan.

12.4 Automasi dan Pipeline Data Science

Automasi dan pipeline dalam data science merupakan komponen esensial yang memastikan alur kerja data berjalan efisien, konsisten, dan dapat direproduksi. Dengan meningkatnya volume dan kompleksitas data, kebutuhan untuk mengotomatisasi proses pengumpulan, pembersihan, analisis, dan penyajian hasil menjadi semakin mendesak. Pipeline data science mengintegrasikan berbagai tahapan ini ke dalam alur kerja yang terstruktur, memungkinkan data scientist untuk fokus pada analisis dan pengambilan keputusan strategis.

Pipeline data science adalah serangkaian proses yang mengotomatisasi alur kerja data, mulai dari pengumpulan hingga penyajian hasil. Komponen utama dari pipeline ini meliputi:

1. Pengumpulan Data:
   Mengakuisisi data dari berbagai sumber seperti basis data, API, atau sensor IoT.
2. Pembersihan dan Transformasi Data:
   Membersihkan data dari kesalahan, mengisi nilai yang hilang, dan mentransformasikan data ke format yang sesuai untuk analisis.
3. Analisis dan Pemodelan:
   Menerapkan teknik statistik dan algoritma machine learning untuk mendapatkan wawasan dari data.
4. Validasi dan Evaluasi Model:
   Menilai kinerja model menggunakan metrik yang relevan untuk memastikan akurasi dan generalisasi.
5. Deployment:
   Menyebarkan model ke lingkungan produksi untuk digunakan dalam pengambilan keputusan atau aplikasi operasional.
6. Monitoring dan Pemeliharaan:
   Memantau kinerja model secara berkelanjutan dan melakukan pemeliharaan atau pembaruan sesuai kebutuhan.

Automasi dalam pipeline data science membawa beberapa keuntungan signifikan:

1. Efisiensi Waktu: Mengurangi waktu yang dibutuhkan untuk tugas-tugas manual berulang, memungkinkan penyelesaian proyek lebih cepat.
2. Konsistensi dan Reproduksibilitas: Memastikan bahwa proses yang sama menghasilkan output yang konsisten, penting untuk validasi dan audit.
3. Skalabilitas: Memungkinkan penanganan volume data yang lebih besar tanpa peningkatan proporsional dalam upaya manual.
4. Pengurangan Kesalahan Manusia: Mengurangi risiko kesalahan yang mungkin terjadi akibat intervensi manual.

Bab 13
Etika dan Privasi dalam Pengolahan Data

13.1 Pentingnya Etika dalam Pengolahan Data

Pengolahan data tidak dapat dilepaskan dari penerapan prinsip-prinsip etika yang bertujuan untuk melindungi privasi, hak individu, dan kepentingan masyarakat. Dalam konteks data science, etika menjadi landasan penting untuk memastikan bahwa data dikelola dengan tanggung jawab yang sesuai dengan nilai-nilai moral dan sosial.

13.1.1 Definisi Etika dalam Pengolahan Data

Etika dalam pengolahan data adalah seperangkat prinsip dan pedoman moral yang mengatur cara data dikumpulkan, diproses, dianalisis, dan digunakan. Etika ini mencakup tanggung jawab untuk memastikan bahwa pengelolaan data dilakukan secara adil, transparan, dan tidak merugikan individu atau kelompok tertentu. Dalam konteks data science, etika menjadi landasan untuk membangun kepercayaan antara pengguna data (perusahaan atau peneliti) dan pemilik data (individu atau masyarakat) (Floridi & Taddeo, 2016).

Seiring dengan pesatnya perkembangan teknologi, pengolahan data telah menjadi bagian penting dalam kehidupan sehari-hari. Data digunakan untuk mendukung pengambilan keputusan di berbagai sektor, termasuk kesehatan, pendidikan, bisnis, dan pemerintahan. Namun, tanpa panduan etika yang jelas, penggunaan data dapat menimbulkan risiko seperti diskriminasi, pelanggaran privasi, atau manipulasi informasi. Oleh karena itu, penerapan etika dalam pengolahan data adalah suatu keharusan untuk memastikan manfaat yang berkelanjutan dari teknologi berbasis data (Zwitter, 2014).

13.1.2 Prinsip-Prinsip Dasar Etika dalam Pengolahan Data

Untuk memastikan bahwa pengolahan data dilakukan secara bertanggung jawab, terdapat beberapa prinsip etika yang harus diterapkan:

Gambar 13.1: Prinsip Etika dalam Pengolahan Data

1. Keadilan (*Fairness*)

   Data harus digunakan dengan cara yang tidak mendiskriminasi individu atau kelompok tertentu. Dalam praktiknya, pengelolaan data harus dilakukan dengan hati-hati untuk menghindari pengaruh bias yang mungkin tercermin dalam data yang digunakan. Misalnya, algoritma machine learning yang dilatih pada data yang bias dapat memperkuat ketidakadilan dalam pengambilan keputusan. Oleh karena itu, penting untuk melakukan pengujian menyeluruh terhadap model dan algoritma guna memastikan bahwa mereka tidak menghasilkan hasil yang diskriminatif (Barocas et al., 2019).

2. Transparansi (*Transparency*)

   Proses pengumpulan dan penggunaan data harus dijelaskan dengan jelas kepada pemilik data. Transparansi dalam pengelolaan data berarti bahwa organisasi harus memberikan informasi yang mudah diakses dan dipahami oleh pemilik data mengenai bagaimana data mereka dikumpulkan, diproses, disimpan, dan digunakan. Hal ini bertujuan untuk menciptakan kepercayaan antara pemilik data dan pengelola data, serta memastikan bahwa tidak ada praktik tersembunyi yang melanggar hak privasi individu (Floridi, 2019).

3. Kebebasan dan Persetujuan (*Autonomy and Consent*)

   Pengumpulan data harus dilakukan dengan persetujuan eksplisit dari pemilik data. Pemilik data harus memahami dengan jelas tujuan pengumpulan dan penggunaan data mereka sebelum memberikan persetujuan. Selain itu, mereka memiliki hak untuk menarik persetujuan kapan saja jika merasa data mereka digunakan dengan cara yang tidak sesuai dengan tujuan awal. Prinsip ini menekankan pentingnya menghormati otonomi individu dalam pengelolaan data pribadi (Nissenbaum, 2010).

4. Tanggung Jawab (*Accountability*)

   Organisasi atau individu yang mengelola data harus bertanggung jawab atas setiap keputusan atau tindakan yang dilakukan berdasarkan data tersebut. Tanggung jawab ini mencakup pengelolaan yang tepat dan memastikan bahwa semua proses pengolahan data mematuhi prinsip-prinsip etika yang berlaku. Jika terjadi kesalahan atau pelanggaran dalam pengelolaan data, tindakan tersebut harus diakui, dilaporkan, dan diperbaiki secepat mungkin. Hal ini bertujuan untuk menjaga kepercayaan pemilik data dan memastikan bahwa data digunakan secara bertanggung jawab (Mittelstadt et al., 2016).

5. Keamanan dan Privasi (*Security and Privacy*)

   Data pribadi harus dilindungi dari akses yang tidak sah melalui langkah-langkah keamanan seperti enkripsi dan kontrol akses. Selain itu, data yang tidak lagi diperlukan harus dihapus secara berkala untuk mengurangi risiko penyalahgunaan. Langkah-langkah ini memastikan bahwa informasi sensitif tetap terlindungi dan tidak jatuh ke tangan pihak yang tidak bertanggung jawab. Dalam konteks pengolahan data yang masif, pengelola data memiliki tanggung jawab besar untuk memprioritaskan keamanan dan privasi sebagai bagian dari praktik pengelolaan yang etis (Tene & Polonetsky, 2013).

6. Manfaat Sosial (*Social Beneficence*)

   Data harus digunakan untuk tujuan yang memberikan manfaat positif bagi individu atau masyarakat. Penggunaan data harus diarahkan untuk menciptakan hasil yang berkontribusi pada kesejahteraan sosial, seperti meningkatkan efisiensi layanan publik atau mendukung penelitian kesehatan. Namun, dalam prosesnya, risiko penggunaan data yang dapat menimbulkan dampak negatif harus diminimalkan, misalnya dengan menghindari penyalahgunaan data untuk manipulasi atau eksploitasi. Dengan memastikan manfaat yang maksimal dan risiko yang minimal, data

dapat menjadi alat yang mendukung perkembangan sosial dan ekonomi yang berkelanjutan (Floridi & Taddeo, 2016).

Prinsip-prinsip etika dalam pengolahan data tidak hanya bertujuan untuk melindungi privasi dan hak individu, tetapi juga untuk membangun ekosistem data yang adil dan bertanggung jawab. Dengan mengikuti prinsip ini, organisasi dapat memastikan bahwa penggunaan data tidak hanya memberikan keuntungan ekonomi atau operasional, tetapi juga mendukung nilai-nilai sosial dan moral yang lebih luas. Dalam era digital ini, pengolahan data yang etis adalah kunci untuk menciptakan kepercayaan dan keberlanjutan dalam pemanfaatan teknologi.

13.2 Privasi dan Keamanan Data

Privasi dan keamanan data adalah pilar utama dalam pengelolaan informasi di era digital. Dengan meningkatnya volume data yang dihasilkan dan diproses setiap hari, memastikan perlindungan terhadap data pribadi dan informasi sensitif menjadi semakin krusial. Organisasi harus mengintegrasikan pendekatan strategis untuk melindungi data dari ancaman, baik melalui langkah-langkah teknologi maupun kebijakan yang mendukung hak privasi individu.

13.2.1 Konsep Privasi dan Hak Individu Terkait Data Pribadi

Privasi data adalah hak fundamental yang melindungi individu dari penyalahgunaan informasi pribadi mereka. Dalam konteks digital, privasi melibatkan kendali atas bagaimana data pribadi dikumpulkan, digunakan, dan dibagikan oleh organisasi atau pihak ketiga. Data pribadi mencakup informasi seperti nama, alamat, nomor identitas, riwayat kesehatan, dan informasi keuangan yang dapat digunakan untuk mengidentifikasi seseorang secara langsung atau tidak langsung (Solove, 2021).

Hak individu atas data pribadi mencakup beberapa aspek utama:

Gambar 13.2: Hak individu atas data pribadi

1. Hak untuk Diketahui
   Pemilik data berhak mengetahui bagaimana data mereka akan digunakan dan oleh siapa. Regulasi seperti GDPR menetapkan bahwa organisasi harus memberikan transparansi penuh dalam pengelolaan data pribadi (Voigt & dem Bussche, 2017).
2. Hak untuk Mengakses dan Memperbaiki Data
   Individu memiliki hak untuk mengakses data mereka yang disimpan oleh organisasi dan meminta perbaikan jika terdapat kesalahan atau informasi yang sudah tidak relevan.
3. Hak untuk Dihapus (*Right to be Forgotten*)
   Dalam beberapa jurisdiksi, individu berhak meminta penghapusan data pribadi mereka dari basis data organisasi jika tidak lagi relevan atau jika penggunaan data melanggar hukum (Tene & Polonetsky, 2013).
4. Hak untuk Membatasi Penggunaan Data
   Pemilik data dapat meminta pembatasan terhadap jenis pemrosesan tertentu, terutama jika data digunakan untuk tujuan yang berbeda dari yang telah disetujui.

Dalam era digital, penting untuk memastikan bahwa privasi data dihormati dan dilindungi oleh organisasi, karena kegagalan melakukannya dapat menyebabkan hilangnya kepercayaan publik dan konsekuensi hukum yang signifikan.

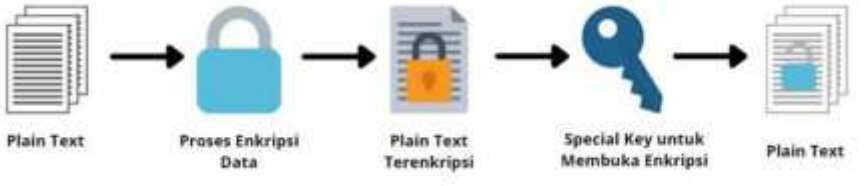
13.2.2 Teknologi Keamanan Data

Teknologi keamanan data adalah langkah-langkah teknis yang digunakan untuk melindungi data pribadi dari akses yang tidak sah, kerusakan, atau kebocoran. Beberapa teknologi utama meliputi:

Gambar 13.3: Teknologi Keamanan Data

1. Enkripsi

   Enkripsi adalah proses mengubah data menjadi format yang tidak dapat dibaca tanpa kunci dekripsi. Teknologi ini digunakan untuk melindungi data selama transmisi dan penyimpanan. Misalnya, protokol seperti TLS (Transport Layer Security) memastikan bahwa data yang dikirim melalui internet tetap aman (Stallings, 2020).



Gambar 13.4: Cara Kerja Enkripsi

Gambar 13.3 menjelaskan proses enkripsi dan dekripsi data, yang digunakan untuk melindungi informasi sensitif dari akses yang tidak sah. Berikut adalah penjelasan setiap tahap dalam gambar:

- Plain Text

  Data awal dalam bentuk teks biasa yang dapat dibaca tanpa proteksi. Contoh: dokumen, email, atau pesan.

- Proses Enkripsi Data

  Proses di mana data biasa (plain text) diubah menjadi format terenkripsi menggunakan algoritma enkripsi. Dalam tahap ini, data menjadi tidak dapat dibaca oleh pihak yang tidak memiliki akses.

- Plain Text Terenkripsi

  Data setelah dienkripsi dikenal sebagai ciphertext. Ciphertext hanya dapat diakses atau dibaca dengan menggunakan kunci enkripsi yang sesuai.

- Special Key untuk Membuka Enkripsi

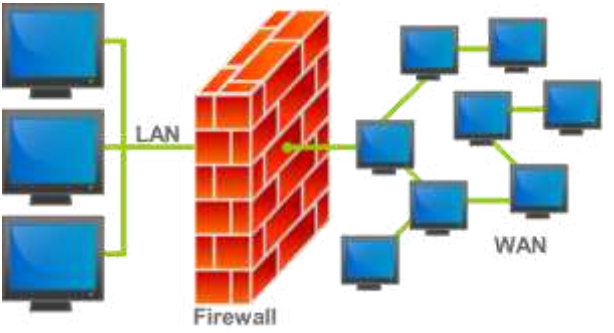
  Kunci khusus (encryption key) yang digunakan untuk mendekripsi data terenkripsi dan mengembalikannya ke bentuk asli. Kunci ini bisa berupa private key atau password yang aman.

- Plain Text

  Data asli yang berhasil didekripsi kembali menggunakan kunci yang benar, sehingga dapat digunakan atau dibaca seperti sebelumnya.

2. Firewall

   Firewall adalah sistem keamanan yang memantau dan mengontrol lalu lintas jaringan berdasarkan aturan keamanan yang ditentukan. Firewall berfungsi sebagai lapisan perlindungan pertama untuk mencegah akses yang tidak sah ke sistem jaringan.



Gambar 13.5: Firewall
(Sumber: wikipedia.com)

Gambar 13.4 menunjukkan peran firewall sebagai pengaman antara Local Area Network (LAN) dan Wide Area Network (WAN). Firewall bertindak sebagai penghalang yang memfilter lalu lintas data, hanya mengizinkan data yang dianggap aman untuk melewati jaringan lokal. LAN, yang terdiri dari perangkat komputer dalam satu area terbatas seperti kantor atau rumah, dilindungi dari ancaman eksternal yang dapat berasal dari WAN, yaitu jaringan luas seperti internet. Dengan demikian, firewall membantu mencegah serangan siber dan menjaga integritas serta keamanan data dalam jaringan lokal.

3. Otentikasi Multifaktor (MFA)

   MFA adalah metode keamanan yang memerlukan lebih dari satu faktor verifikasi untuk mengakses sistem. Biasanya, ini mencakup kombinasi kata sandi, kode OTP (One-Time Password), atau sidik jari. MFA mengurangi risiko akses tidak sah, bahkan jika salah satu faktor keamanan telah dikompromikan (Anderson, 2020).

Gambar 13.5 menjelaskan proses autentikasi berbasis konteks untuk mengamankan akses ke aplikasi dan data. Sistem ini memeriksa berbagai sinyal, termasuk identitas pengguna dan lokasi, perangkat yang digunakan, aplikasi yang diakses, dan risiko real-time. Berdasarkan analisis sinyal ini, setiap upaya akses diverifikasi dan diarahkan ke salah satu dari tiga tindakan: akses diizinkan (allow access), memerlukan autentikasi multifaktor (MFA), atau akses diblokir sepenuhnya (block access). Pendekatan ini memastikan keamanan berlapis dengan memitigasi risiko ancaman siber sambil memastikan akses yang sah ke data dan aplikasi penting.

Gambar 13.6: Otentikasi Multifaktor (MFA)

(Sumber: lean.microsoft.com)

4. Pemantauan Keamanan
   Teknologi seperti SIEM (Security Information and Event Management) digunakan untuk mendeteksi dan merespons ancaman keamanan secara real-time. Alat ini membantu organisasi mendeteksi aktivitas mencurigakan sebelum menjadi ancaman besar.

Implementasi teknologi ini adalah langkah kritis untuk melindungi data dari ancaman yang terus berkembang di dunia maya.

13.2.3 Ancaman Keamanan Data dan Langkah Mitigasi

Ancaman keamanan data semakin kompleks di era digital, dengan pelanggaran data menjadi salah satu masalah utama yang dihadapi organisasi. Beberapa ancaman umum meliputi (Fairuzabadi et al., 2023):

1. Pelanggaran Data (*Data Breach*)
   Pelanggaran data terjadi ketika data sensitif diakses oleh pihak yang tidak berwenang. Contoh kasus terkenal adalah pelanggaran data Equifax pada 2017 yang mengakibatkan kebocoran data pribadi lebih dari 147 juta individu (Tang, 2019).
2. Ransomware
   Serangan ransomware mengenkripsi data organisasi dan meminta tebusan untuk memulihkan akses. Contoh kasus adalah serangan WannaCry pada 2017 yang memengaruhi lebih dari 200.000 komputer di seluruh dunia.
3. Phishing
   Serangan phishing melibatkan pengelabuan pengguna untuk memberikan informasi sensitif seperti kata sandi atau informasi keuangan. Phishing sering dilakukan melalui email yang tampak meyakinkan.

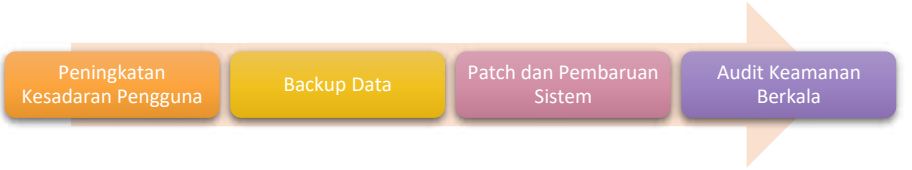
Berikut adalah daftar kejadian Pelanggaran Data (Data Breach), Ransomware, dan Phishing dari tahun 2017 hingga 2023:

Tabel 13.1: Contoh Kejadian Ancaman dari Tahun 2017 hingga 2023.

No	Jenis Serangan	Tahun	Deskripsi Kejadian
1	Pelanggaran Data	2017	Equifax Data Breach: Kebocoran informasi pribadi lebih dari 147 juta individu, termasuk nomor Jaminan Sosial, tanggal lahir, dan alamat.
2	Pelanggaran Data	2019	Capital One Data Breach: Kebocoran data pribadi lebih dari 100 juta pelanggan AS dan 6 juta di Kanada, termasuk nomor kartu kredit dan informasi keuangan lainnya.
3	Pelanggaran Data	2021	Facebook Data Breach: Informasi pribadi dari lebih dari 530 juta pengguna Facebook bocor ke internet, termasuk nama lengkap, lokasi, dan nomor telepon.
4	Pelanggaran Data	2023	T-Mobile Data Breach: Informasi dari lebih dari 37 juta pelanggan, termasuk nomor telepon, email, dan alamat, bocor akibat serangan terhadap sistem perusahaan.

No	Jenis Serangan	Tahun	Deskripsi Kejadian
5	Ransomware	2017	WannaCry Ransomware Attack: Serangan ransomware global yang memengaruhi lebih dari 200.000 komputer di 150 negara, mengunci file penting pengguna dan meminta pembayaran dalam bentuk Bitcoin untuk membukanya.
6	Ransomware	2021	Colonial Pipeline Attack: Serangan ransomware oleh kelompok DarkSide yang mengakibatkan penutupan salah satu pipa bahan bakar utama di Amerika Serikat, mengganggu pasokan energi di wilayah Pantai Timur.
7	Ransomware	2023	Royal Ransomware: Serangan ransomware yang menargetkan organisasi kesehatan di AS, menyebabkan gangguan layanan kesehatan dan permintaan pembayaran tebusan besar.
8	Phishing	2020	Google Docs Phishing Attack: Email phishing yang berpura-pura sebagai undangan berbagi dokumen Google, bertujuan untuk mencuri kredensial login pengguna.
9	Phishing	2021	Covid-19 Vaccine Phishing Scams: Penipuan yang mengatasnamakan penyedia vaksin Covid-19, meminta informasi pribadi dan pembayaran palsu dari individu yang ingin mendapatkan vaksin lebih awal.
10	Phishing	2023	Microsoft Phishing Emails: Serangan phishing yang menargetkan pengguna Microsoft dengan email palsu yang berisi tautan ke halaman login tiruan untuk mencuri kredensial login.

Ancaman keamanan data yang terus berkembang memerlukan pendekatan strategis untuk mitigasi risiko. Organisasi harus menerapkan kombinasi langkah-langkah teknis dan pelatihan sumber daya manusia guna melindungi data sensitif secara efektif. Langkah-langkah mitigasi berikut dapat membantu mengurangi risiko (Fairuzabadi et al., 2023):



Gambar 13.7: Langkah-langkah Mitigasi

1. Peningkatan Kesadaran Pengguna
   Pelatihan rutin kepada karyawan diperlukan untuk mengenali ancaman seperti phishing dan serangan sosial lainnya. Program pelatihan dapat mencakup simulasi serangan phishing untuk meningkatkan kemampuan deteksi mereka terhadap ancaman tersebut. Langkah ini dapat menurunkan kemungkinan pelanggaran data yang disebabkan oleh kelalaian manusia.
2. Backup Data
   Membuat salinan data secara rutin dan menyimpannya di lokasi yang aman, seperti layanan cloud terenkripsi, memastikan bahwa data dapat dipulihkan jika terjadi serangan ransomware. Pendekatan ini mengurangi dampak finansial dan operasional yang mungkin ditimbulkan oleh kehilangan data.
3. Patch dan Pembaruan Sistem
   Perangkat lunak dan sistem operasi harus diperbarui secara berkala untuk melindungi terhadap kerentanan yang baru ditemukan. Organisasi dapat mengimplementasikan sistem manajemen tambalan untuk memastikan semua perangkat selalu diperbarui tanpa jeda yang signifikan.
4. Audit Keamanan Berkala
   Penilaian keamanan yang terstruktur membantu organisasi mengidentifikasi dan memperbaiki celah keamanan sebelum dapat dieksploitasi. Audit ini mencakup pengujian penetrasi, analisis log, dan evaluasi kebijakan keamanan yang ada untuk memastikan kepatuhan terhadap standar industri terkini.

Dengan langkah-langkah ini, organisasi dapat mengurangi risiko serangan keamanan data dan melindungi informasi sensitif dengan lebih baik.

13.3 Kepatuhan Regulasi Data

Kepatuhan terhadap regulasi data adalah elemen kunci dalam menciptakan lingkungan digital yang aman dan terpercaya. Dengan menerapkan kebijakan perlindungan data yang sesuai dengan standar hukum, organisasi dapat

memastikan bahwa data pribadi individu terlindungi dari penyalahgunaan. Regulasi semacam ini juga memberikan kejelasan bagi organisasi dalam menjalankan bisnis berbasis data, menciptakan keseimbangan antara kebutuhan komersial dan hak privasi masyarakat.

13.3.1 Regulasi data global

Regulasi data global dirancang untuk melindungi hak privasi individu di era digital. Dua contoh utama adalah General Data Protection Regulation (GDPR) yang berlaku di Uni Eropa dan California Consumer Privacy Act (CCPA) yang berlaku di negara bagian California, Amerika Serikat. Kedua regulasi ini menetapkan standar tinggi untuk perlindungan data pribadi.

1. GDPR

   Diberlakukan pada tahun 2018, GDPR adalah salah satu regulasi data paling komprehensif di dunia. Regulasi ini menetapkan bahwa organisasi yang mengumpulkan atau memproses data warga Uni Eropa harus mematuhi prinsip-prinsip seperti transparansi, minimasi data, dan akuntabilitas (Voigt & dem Bussche, 2017). Selain itu, GDPR memberikan hak kepada individu, termasuk "*right to be forgotten*" dan hak akses terhadap data pribadi mereka.

2. CCPA

   Diterapkan pada tahun 2020, CCPA memberikan konsumen di California hak untuk mengetahui data apa yang dikumpulkan tentang mereka, tujuan penggunaannya, dan pihak ketiga yang menerima data tersebut. Konsumen juga memiliki hak untuk menolak penjualan data mereka. Regulasi ini menekankan transparansi dan kontrol konsumen atas data pribadi mereka (Greenleaf, 2020).

Tabel 13.2 merangkum perbedaan utama antara RUU PDP dan GDPR terkait perlindungan data pribadi (Kominfo, 2020).

Tabel 13.2: Perbedaan utama antara RUU PDP dan GDPR
(Sumber: kominfo.com)

Komponen	RUU PDP	GDPR
Pengecualian terhadap hak pemilik data	Secara penuh, berdasarkan beberapa area kepentingan yang diatur dalam RUU PDP	Secara parsial, berdasarkan prinsip kebutuhan dan proporsionalitas

Komponen	RUU PDP	GDPR
Pembatasan penyimpanan data pribadi	Membuka ruang perpanjangan periode penyimpanan data pribadi selama mekanisme dan tujuannya diatur dalam peraturan perundang-undangan	Periode penyimpanan data pribadi dapat diperpanjang untuk beberapa tujuan spesifik yang terdapat dalam GDPR
Kewajiban pengendali data pribadi	Mengatur secara umum, tanpa melihat tinggi rendahnya risiko pemrosesan data pribadi yang dilakukan	Diberlakukannya Data Protection Impact Assessment (DPIA) untuk pemrosesan data pribadi berisiko tinggi
Kewajiban prosesor data pribadi	Mengisyaratkan beberapa kewajiban pengendali data pribadi yang juga menjadi kewajiban prosesor data pribadi	Mengatur beberapa kewajiban prosesor data pribadi yang berbeda dari kewajiban pengendali data pribadi
Kebutuhan pengamanan data pribadi	Mengatur secara umum, berdasarkan kapasitas pengendali/pemroses data akan diatur dalam aturan turunan	Berdasarkan kapasitas dan kompetensi dari pengendali/pemroses data pribadi
Mekanisme cross-border data transfer	Tiga aspek pertimbangan yang sama dengan GDPR, namun tidak harus diikuti secara bertahap, melainkan sebagai berfungsi sebagai opsi pertimbangan	Tiga aspek pertimbangan dalam melakukan transfer data lintas batas yang harus diikuti secara bertahap: kelayakan PDP, perjanjian internasional/kontrak, persetujuan pemilik data pribadi
Mekanisme sanksi	Mengatur sanksi administratif untuk kelalaian terhadap kewajiban, dan sanksi pidana untuk perbuatan penyalahgunaan	Hanya mengatur sanksi administratif secara detail

Komponen	RUU PDP	GDPR
Otoritas perlindungan data pribadi	Otoritas perlindungan data pribadi independen yang dilaksanakan oleh Kemenkominfo	Otoritas perlindungan data pribadi independen yang dilaksanakan di tingkat Uni Eropa dan masing-masing negara anggota Uni Eropa

13.3.2 Regulasi Regional seperti PDPA dan Relevansi di Asia Tenggara

Di tingkat regional, regulasi data pribadi juga mulai diterapkan untuk mengimbangi kebutuhan perlindungan data dalam konteks lokal. Salah satu regulasi yang signifikan adalah Personal Data Protection Act (PDPA).

1. PDPA Singapura

   Diterapkan sejak 2012, PDPA menetapkan kerangka hukum yang rinci mengenai pengumpulan, penggunaan, dan pengungkapan data pribadi oleh organisasi. Regulasi ini bertujuan untuk menyeimbangkan perlindungan hak privasi individu dengan kebutuhan organisasi untuk menggunakan data dalam kegiatan bisnis yang sah. PDPA juga mewajibkan organisasi untuk memastikan transparansi dalam pengelolaan data, memberikan persetujuan yang eksplisit dari pemilik data, dan menerapkan langkah-langkah keamanan yang memadai untuk mencegah akses tidak sah. Pelanggaran terhadap PDPA dapat mengakibatkan denda yang signifikan, sehingga menekankan pentingnya kepatuhan terhadap peraturan ini (S. Lim, 2019).

2. Undang-Undang Perlindungan Data Pribadi di Indonesia

   Di Indonesia, Undang-Undang Perlindungan Data Pribadi (UU PDP) disahkan pada tahun 2022 untuk memberikan kerangka hukum yang komprehensif dalam melindungi data pribadi warga negara. UU ini menetapkan bahwa setiap pengumpulan, pemrosesan, dan penyimpanan data pribadi harus dilakukan berdasarkan prinsip-prinsip transparansi, persetujuan eksplisit, dan keamanan yang memadai. UU ini juga mewajibkan organisasi untuk memberikan mekanisme bagi individu untuk mengakses, memperbarui, atau menghapus data mereka jika diperlukan. Penegakan regulasi ini bertujuan untuk menciptakan lingkungan digital

yang lebih aman, sekaligus mendorong kepercayaan masyarakat terhadap pengelolaan data oleh organisasi (Dewan, 2022).

Regulasi seperti PDPA dan UU PDP menjadi langkah penting dalam melindungi hak privasi individu di kawasan Asia Tenggara yang mengalami pertumbuhan digital yang pesat.

13.3.3 Pentingnya Kepatuhan Hukum bagi Organisasi

Kepatuhan terhadap regulasi data bukan hanya kewajiban hukum tetapi juga strategi bisnis yang penting. Organisasi yang gagal mematuhi regulasi data menghadapi risiko denda yang signifikan, seperti yang terlihat dalam kasus pelanggaran GDPR oleh perusahaan teknologi besar (Voigt & dem Bussche, 2017). Selain itu, ketidakpatuhan dapat merusak reputasi organisasi dan mengurangi kepercayaan konsumen.

Di sisi lain, kepatuhan terhadap regulasi data dapat memberikan keuntungan kompetitif. Dengan menunjukkan komitmen terhadap privasi dan keamanan data, organisasi dapat meningkatkan kepercayaan pelanggan dan memperkuat hubungan dengan pemangku kepentingan. Implementasi kepatuhan juga memotivasi pengelolaan data yang lebih efisien, sehingga mendukung inovasi dan pertumbuhan bisnis yang berkelanjutan (Rosadi, 2015).

Daftar Pustaka

Abdul Kadhar, K. M., & Anand, G. (2021). Introduction to Data Science. In *Data Science with Raspberry Pi* (pp. 1–12). Apress. https://doi.org/10.1007/978-1-4842-6825-4_1

Aggarwal, C. C. (2015). *Outlier Analysis*. Springer.

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Springer.

Al-Haija, Q. A. (2022). Exploration of Tools for Data Science. In *Data Science with Semantic Technologies* (pp. 31–69). John Wiley & Sons, Ltd. https://doi.org/https://doi.org/10.1002/9781119865339.ch2

Albright, S. C., & Winston, W. L. (2017). *Business Analytic* (6th ed.). Change Learning.

Anderson, R. J. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems* (3rd ed.). Wiley.

Anitha, M., Kumari, V. S., Pillai, N. M., Jayarin, P. J., & David, D. B. (2024). Exploring Cutting-Edge Machine Learning and Data Mining Techniques for Enhancing Big Data Management with Advanced Algorithmic Strategies for Optimal Data Processing and Analysis. *2nd IEEE International Conference on Advances in Information Technology, ICAIT 2024 - Proceedings*. https://doi.org/10.1109/ICAIT61638.2024.10690283

Antonio Nunoand de Almeida, A. and N. L. (2022). Data Mining and Predictive Analytics for E-Tourism. In M. and G. U. and H. W. Xiang Zhengand Fuchs (Ed.), *Handbook of e-Tourism* (pp. 531–555). Springer International Publishing. https://doi.org/10.1007/978-3-030-48652-5_29

Arthi, K., Sankaradass, V., Parveen, N., & Muralidharan, J. (2023). Methods of cross-validation and bootstrapping. In *Toward Artificial General Intelligence: Deep Learning, Neural Networks, Generative AI*. De Gruyter. https://doi.org/10.1515/9783111323749-007

Attaway, S. (2016). *MATLAB: A Practical Introduction to Programming and Problem Solving* (4th ed.). Butterworth-Heinemann.

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and Machine*

Learning: Limitations and Opportunities.

Beniwal, M., Singh, A., & Kumar, N. (2024). Forecasting multistep daily stock prices for long-term investment decisions: A study of deep learning models on global indices. *Engineering Applications of Artificial Intelligence*, *129*. https://doi.org/10.1016/j.engappai.2023.107617

Bezdan, T., Strumberger, I., & Tuba, M. (2024). Optimizing Machine Learning for Breast Cancer Detection by Hybrid Metaheuristic Approach. *12th International Symposium on Digital Forensics and Security, ISDFS 2024*. https://doi.org/10.1109/ISDFS60797.2024.10527334

Biemer, P. P., & Lyberg, L. E. (2003). *Introduction to Survey Quality*. John Wiley \& Sons.

Boslaugh, S. (2007). *Secondary Data Sources for Public Health: A Practical Guide*. Cambridge University Press.

Bouwer, L. M., Dransch, D., Ruhnke, R., Rechid, D., Frickenhaus, S., & Greinert, J. (2022). Integrating Data Science and Earth Science: Challenges and Solutions. In *SpringerBriefs in Earth System Sciences*.

Brynjolfsson, E., & McAfee, A. (2014). *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton \& Company.

Cao, L. (2018). *Data Science Thinking: The Next Scientific, Technological and Economic Revolution*. Springer. https://doi.org/10.1007/978-3-319-93031-2

Carneiro, T., Da Silva, J. A. M., & Nepomuceno, T. (2020). *Performance Evaluation of Google Colab and Jupyter Notebook for Data Analysis*. Springer.

Chambers, B., & Zaharia, M. (2018). *Spark: The Definitive Guide: Big Data Processing Made Simple*. O'Reilly Media.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2021). A Simple Framework for Contrastive Learning of Visual Representations. *Proceedings of the 38th International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.2002.05709

Choi, T.-M., Wallace, S. W., & Wang, Y. (2018). Big data analytics in operations management. *Production and Operations Management*, *27*(10), 1868–1881. https://doi.org/10.1111/poms.12838

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.

Cielen, D., Meysman, A. D. B., & Ali, M. (2016). *Introducing Data Science*.

Cochran, W. G. (1977). *Sampling Techniques* (3rd ed.). John Wiley \& Sons.

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, *13*(6), 377–387. https://doi.org/10.1145/362384.362685

Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.

Dasu, T., & Johnson, T. (2003). *Exploratory Data Mining and Data Cleaning*. Wiley-Interscience.

Davenport, T. H., & Dyché, J. (2014). Big data at work: dispelling the myths, uncovering the opportunities. In *Choice Reviews Online* (Vol. 51, Issue 11). Harvard Business Review Press. https://doi.org/10.5860/choice.51-6260

David Freedman Robert Pisani, & Purves, R. (2007). *Statistics*. W. W. Norton \& Company.

David S. Moore William I. Notz, & Fligner, M. A. (2017). *The Basic Practice of Statistics*. W. H. Freeman.

Dewan, M. (2022). The Development of Data Privacy Law in Indonesia. *Journal of Southeast Asian Law*, *15*(2), 89–105.

Dhar, V. (2013). Data science and prediction. *Communications of the ACM*, *56*(12), 64–73.

Dignum, V. (2019). *Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way*. Springer.

Dimotikalis, Y., Karagrigoriou, A., Parpoula, C., & Skiadas, C. H. (2021a). Applied Modeling Techniques and Data Analysis 1: Computational Data Analysis Methods and Tools. In *Applied Modeling Techniques and Data Analysis 1: Computational Data Analysis Methods and Tools*. wiley. https://doi.org/10.1002/9781119821588

Dimotikalis, Y., Karagrigoriou, A., Parpoula, C., & Skiadas, C. H. (2021b). Applied Modeling Techniques and Data Analysis 2: Financial, Demographic, Stochastic and Statistical Models and Methods. In *Applied Modeling Techniques and Data Analysis 2: Financial, Demographic, Stochastic and Statistical Models and Methods*. wiley. https://doi.org/10.1002/9781119821724

Dinov, I. D. (2023). *Data Science and Predictive Analytics: Biomedical and Health Applications using R*. Springer. https://link.springer.com/book/10.1007/978-3-031-17483-4

Domingos, P. (2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books.

Downey, A. B. (2011). Think Stats: Probability and Statistics for Programmers. In *Psychological Bulletin* (Vol. 70, Issue 2).

Elliott, M. (2020). *Using Data Analytics and Decision-Making Tools for Agribusiness and Extension Education*. https://doi.org/10.13140/RG.2.2.14129.74080

Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2007). Duplicate Record Detection: A Survey. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, *19*(1), 399–421.

Elmasri, R., & Navathe, S. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson Education.

EMC Education Services. (2015). Data Science & Big Data Analytics. In *Data Science & Big Data Analytics*. John Wiley \& Sons, Inc. https://doi.org/10.1002/9781119183686

Erl, T., Khattak, W., & Buhler, P. (2016). *Big Data Fundamentals: Concepts, Drivers \& Techniques* (1st ed.). Prentice Hall.

Fairuzabadi, M., Adytia, P., Wahyuni, Prastyo, P. H., Resha, M., Anwar, N., Intan, I., Amiruddin, M. R. K., Wulan, N., Sekti, B. A., & Erna, A. (2024). *Machine Learning: Konsep, Algoritma dan Implementasi*. Yayasan Kita Menulis.

Fairuzabadi, M., Imam, I. P. S., Ekowicaksono, Arifah, F. N., Kesuma, R. I., Anwar, N., Setiawan, A., & Lontaan, R. J. (2025). *Deep Learning untuk Pemula: Memahami Algoritma, Tools, dan Masa Depan AI*. Yayasan Kita Menulis.

Fairuzabadi, M., Pangaribuan, J. J., Moedjahedy, J. H., Sihotang, J. I., Simarmata, J., Andryanto, A., Jaya, A. K., Sasongko, D., Turnip, T. N., Suardinata, S., & others. (2023). *Keamanan Sistem Informasi dan Kriptografi*. Yayasan Kita Menulis.

Fairuzabadi, M., Sinambela, M., Taju, S. W., Syahrani, A., Arni, S., Yuliansyah, H., Saputra, A. I. H., Stephane, I., Pakpahan, A., Lubis, M., & Liem, A. T. (2024). *Data Science: Sebuah Pengantar untuk Pemula*. Yayasan Kita Menulis.

Floridi, L. (2019). *The Logic of Information: A Theory of Philosophy as Conceptual Design*. Oxford University Press.

Floridi, L., & Taddeo, M. (2016). What is data ethics? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and*

Engineering Sciences, *374*(2083), 20160360.

Gama, J. (2010). *Knowledge Discovery from Data Streams*. CRC Press.

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

Gholamiangonabadi, D., Kiselov, N., & Grolinger, K. (2020). Deep Neural Networks for Human Activity Recognition with Wearable Sensors: Leave-One-Subject-Out Cross-Validation for Model Selection. *IEEE Access*, *8*, 133982 – 133994. https://doi.org/10.1109/ACCESS.2020.3010715

Gilbert, J. R., & Strang, G. (1991). *Introduction to Linear Algebra*. Wellesley-Cambridge Press.

Glass, R., & Callahan, S. (2014). *The Big Data-Driven Business: How to Use Big Data to Win Customers, Beat Competitors, and Boost Profits*. Wiley.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. https://www.deeplearningbook.org

Govindarajan, M. (2020). Introduction to data science. In *Handbook of Research on Engineering, Business, and Healthcare Applications of Data Science and Analytics*. IGI Global. https://doi.org/10.4018/978-1-7998-3053-5.ch001

Granger, B. E., & Pérez, F. (2020). *Jupyter: Tools for Open Source Computational Research*. Springer.

Groves, R. M., Fowler, F. J., Couper, M. P., Lepkowski, J. M., Singer, E., & Tourangeau, R. (2009). *Survey Methodology*. John Wiley \& Sons.

Hanselman, D., & Littlefield, B. (2011). *Mastering MATLAB* (1st ed.). Pearson.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Healy, K. (2018). *Data Visualization: A Practical Introduction*. Princeton University Press.

Higham, D. J., & Higham, N. J. (2016). *MATLAB Guide* (3rd ed.). Society for Industrial and Applied Mathematics.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Hudiburgh, L. M., & Garbinsky, D. (2020). Data Visualization: Bringing Data to Life in an Introductory Statistics Course. *Journal of Statistics Education*, *28*(3), 262–279. https://doi.org/10.1080/10691898.2020.1796399

Husain, H. A. (2023). Jenis-Jenis Database? Intip Lebih Jauh Macam Jenis Database Yuk! In *Codepolitan*. https://www.codepolitan.com/blog/jenis-jenis-database-intip-lebih-jauh-macam-jenis-database-yuk/

Hylton, A., Lim, I., Moy, M., & Short, R. (2022). Interpreting a topological measure of complexity for decision boundaries. In *Data Analysis and Related Applications, Volume 1: Computational, Algorithmic and Applied Economic Data Analysis* (Vol. 9). wiley. https://doi.org/10.1002/9781394165513.ch16

Idrissi, A. (2023). *Modern Artificial Intelligence and Data Science: Tools, Techniques and Systems*. Springer Nature Switzerland. https://books.google.com/books/about/Modern_Artificial_Intelligence_and_Data.html?id=zVjYzwEACAAJ

Ismail, M. N., Kallow, S. M., Ridah, M. J., Abu-Alshaeer, M. J., & Khlaponin, Y. (2024). Quantitative Insights and Challenges in Big Data from a Statistical Perspective. *Journal of Ecohumanism*, *3*(5), 290 – 307. https://doi.org/10.62754/joe.v3i5.3907

Jaakkola, H., & Thalheim, B. (2020). Sixty years - And more - And data modelling. *Frontiers in Artificial Intelligence and Applications*, *333*, 56 – 75. https://doi.org/10.3233/FAIA200820

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510.

Johnson, R. A., & Bhattacharyya, G. K. (2014). *Statistics: Principles and Methods*. Wiley.

Johnston, M. P. (2017). Secondary data analysis: A method of which the time has come. *Qualitative and Quantitative Methods in Libraries (QQML)*, *3*(3), 619–626.

Jolliffe, I. T. (2011). *Principal Component Analysis*. Springer.

Jones, B. (2019). *Communicating Data with Tableau: Designing, Developing, and Delivering Data Visualizations*. O'Reilly Media. https://www.oreilly.com/library/view/communicating-data-

with/9781449372019/

Karau, H., & Warren, R. (2017). *High-Performance Spark: Best Practices for Scaling and Optimizing Apache Spark*. O'Reilly Media.

Kluyver, T., Ragan-Kelley, B., & Pérez, F. (2016). Jupyter Notebooks: A Publishing Format for Reproducible Computational Workflows. *Springer*.

Kominfo. (2020). *UU PDP akan Permudah Pertukaran Data dengan Negara Lain*. https://aptika.kominfo.go.id/2020/11/uu-pdp-akan-permudah-pertukaran-data-dengan-negara-lain/

Kotsiantis, S. B. (2020). Supervised Machine Learning: A Review of Classification Techniques. *Informatica*. https://doi.org/10.31449/inf.v31i3.1487

Kroese, D. P., Botev, Z. I., Taimre, T., & Vaisman, R. (2019). Data Science and Machine Learning: Mathematical and Statistical Methods. In *Data Science and Machine Learning: Mathematical and Statistical Methods*. https://doi.org/10.1201/9780367816971

Krotov, V., & Silva, L. (2018). Legality and Ethics of Web Scraping. In *Journal of Theoretical and Applied Electronic Commerce Research* (Vol. 13, Issue 2).

Kshatri, S. S., Singh, D., Goswami, T., & Sinha, G. R. (2022). Introduction to statistical modeling in machine learning. In *Statistical Modeling in Machine Learning: Concepts and Applications*. Elsevier. https://doi.org/10.1016/B978-0-323-91776-6.00007-5

Kshetri, N. (2014). Big data's impact on privacy, security and consumer welfare. *Telecommunications Policy*, *38*(11), 1134–1145. https://doi.org/10.1016/j.telpol.2014.10.002

Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*.

Larose, D. T., & Larose, C. D. (2014). *Discovering Knowledge in Data: An Introduction to Data Mining* (2nd ed.). Wiley.

Larracy, R., Phinyomark, A., & Scheme, E. (2021). Machine Learning Model Validation for Early Stage Studies with Small Sample Sizes. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2314 – 2319. https://doi.org/10.1109/EMBC46164.2021.9629697

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Leo Breiman Jerome H. Friedman, R. A. O., & Stone, C. J. (1984).

Classification and Regression Trees. *Wadsworth & Brooks/Cole Advanced Books & Software*.

Leung, C. K. (2021). Data Science for Big Data Applications and Services: Data Lake Management, Data Analytics and Visualization. *Advances in Intelligent Systems and Computing*, *899 AISC*, 28 – 44. https://doi.org/10.1007/978-981-15-8731-3_3

Lim, S. (2019). Understanding PDPA: Personal Data Protection Act. *Journal of Data Protection and Privacy*, *3*(1), 45–58.

Lim, W. M. (2024). What Is Quantitative Research? An Overview and Guidelines. *Australasian Marketing Journal*, *1*(24). https://doi.org/10.1177/14413582241264622

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1–167. https://doi.org/10.2200/S00416ED1V01Y201204HLT016

Liu, C. (2024). Research on Corporate Financial Risk Prediction and Early Warning System Based on Big Data Analysis. *Learning and Analytics in Intelligent Systems*, *42*, 209 – 218. https://doi.org/10.1007/978-3-031-70598-4_20

Lohr, S. L. (2010). *Sampling: Design and Analysis*. Cengage Learning.

Long, P., & Siemens, G. (2011). Penetrating the Fog: Analytics in Learning and Education. *EDUCAUSE Review*, *46*(5), 30–32. http://search.proquest.com.proxy.library.vanderbilt.edu/docview/964183308/13AF5BC47C138E29FF2/5?accountid=14816%5Cnhttps://login.proxy.library.vanderbilt.edu/login?url=http://search.proquest.com/docview/964183308/13AF5BC47C138E29FF2/5?accountid=14816

Lu, J. (2022). Data Science in the Business Environment: Architecture, Process and Tools. *Communications in Computer and Information Science*, *1528 CCIS*, 279 – 293. https://doi.org/10.1007/978-3-030-95502-1_22

Maheshwari, A. K. (2015). *Data Analytics Made Accessible For Beginners*. https://www.amazon.com/dp/B071Z8QCJQ/ref=sspa_dk_detail_0?psc=1

Mammen, E., Marron, J. S., Turlach, B. A., & Wand, M. P. (2001). A General Projection Framework for Constrained Smoothing. *Statistical Science*, *16*(3), 232–248. https://doi.org/10.1214/ss/1009213727

Marino, S., Xu, J., Zhao, Y., Zhou, N., Zhou, Y., & Dinov, I. D. (2018). Controlled feature selection and compressive big data analytics: Applications to biomedical and health studies. *PLoS ONE*, *13*(8).

https://doi.org/10.1371/journal.pone.0202674

Marr, B. (2017). *Data Strategy: How to Profit from a World of Big Data, Analytics and the Internet of Things*. Kogan Page Publishers.

McCreary, D., & Kelly, A. (2013). Making Sense of NoSQL - A guide for managers and the rest of us. In *Manning Publications*. Manning. http://www.dama.org/files/public/ia_pe_2013-11-McCrearyDan.pdf

McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.

McKinsey Global Institute. (2016). *The Age of Analytics: Competing in a Data-Driven World*. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-age-of-analytics-competing-in-a-data-driven-world

McLachlan, G. J. (2004). *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley \& Sons.

Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. *Big Data \& Society*, *3*(2), 2053951716679679.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5th ed.). Wiley.

Montgomery, D. C., & Runger, G. C. (2018). *Applied Statistics and Probability for Engineers*. Wiley.

Mukherjee, S., & Srinivasa Rao, Y. (2022). Auto-ML Web-application for Automated Machine Learning Algorithm Training and evaluation. *2022 IEEE 7th International Conference for Convergence in Technology, I2CT 2022*. https://doi.org/10.1109/I2CT54291.2022.9825329

Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media.

Muniasamy, M., Anandhavalli, N., Naim, A., & Kumar, A. (2024). *Data Visualization Tools for Business Applications*. IGI Global. https://books.google.com/books/about/Data_Visualization_Tools_for_Business_Ap.html?id=fughEQAAQBAJ

Murray, D. G. (2020). *Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software* (2nd ed.). Wiley. https://www.wiley.com/en-us/Tableau+Your+Data%21%3A+Fast+and+Easy+Visual+Analysis+with+Tableau+Software%2C+2nd+Edition-p-9781119001195

Nelli, F. (2015). Python Data Analytics: Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language. In *Python*

Data Analytics: Data Analysis and Science Using Pandas, Matplotlib, and the Python Programming Language. https://doi.org/10.1007/978-1-4842-0958-5

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559–569.

Nguyen, T. T., Zhou, L., Spiegler, V., Shin, H., & Lim, M. K. (2020). Big Data Analytics in Supply Chain Management: A State-of-the-Art Literature Review. *Computers & Operations Research*. https://doi.org/10.1016/j.cor.2020.104912

Nissenbaum, H. (2010). *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford University Press.

Nourbakhsh, Z., & Habibi, N. (2023). Combining LSTM and CNN methods and fundamental analysis for stock price trend prediction. *Multimedia Tools and Applications*, *82*(12), 17769–17799. https://doi.org/10.1007/s11042-022-13963-0

Olorunnimbe, K., & Viktor, H. (2023). Deep learning in the stock market—a systematic survey of practice, backtesting, and applications. *Artificial Intelligence Review*, *56*(3), 2057–2109. https://doi.org/10.1007/s10462-022-10226-0

Patton, M. Q. (2002). *Qualitative Research and Evaluation Methods*. Sage Publications.

Plevris, V., Solorzano, G., Bakas, N. P., & Ben Seghier, M. E. A. (2022). INVESTIGATION OF PERFORMANCE METRICS IN REGRESSION ANALYSIS AND MACHINE LEARNING-BASED PREDICTION MODELS. *World Congress in Computational Mechanics and ECCOMAS Congress*. https://doi.org/10.23967/eccomas.2022.155

Pratap, R. (2010). *Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers* (7th ed.). Oxford University Press.

Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media.

Rafało, M. (2022). Cross validation methods: Analysis based on diagnostics of thyroid cancer metastasis. *ICT Express*, *8*(2), 183 – 188. https://doi.org/10.1016/j.icte.2021.05.001

Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, *2*(1), 3.

https://doi.org/10.1186/2047-2501-2-3

Rahayu, A., Pangestu, G., Yulianto, Y., & Wicaksono, D. W. (2024). The importance of derivative validation in data science process. *AIP Conference Proceedings*, *2927*(1). https://doi.org/10.1063/5.0192597

Rahm, E., & Do, H. H. (2000). Data Cleaning: Problems and Current Approaches. *IEEE Data Engineering Bulletin*, *23*(4), 3–13.

Rajaraman, A., & Ullman, J. D. (2011). *Mining of Massive Datasets* (1st ed.). Cambridge University Press.

Roberts, D. C. (2024). *Introduction to Databases: A Focus on Practical Solutions*. Jada Press.

Rosadi, S. D. (2015). Asian Data Privacy Laws, Trade and Human Rights Perspectives by Graham Greenleaf. In *PADJADJARAN Jurnal Ilmu Hukum (Journal of Law)* (Vol. 2, Issue 2). Oxford University Press. https://doi.org/10.22304/pjih.v2n2.a11

Rubin, D. B. (2004). *Multiple Imputation for Nonresponse in Surveys*. John Wiley \& Sons.

Seraj, A., Mohammadi-Khanaposhtani, M., Daneshfar, R., Naseri, M., Esmaeili, M., Baghban, A., Habibzadeh, S., & Eslamian, S. (2022). Cross-validation. In *Handbook of HydroInformatics: Volume I: Classic Soft-Computing Techniques*. Elsevier. https://doi.org/10.1016/B978-0-12-821285-1.00021-X

Shan, C., Chen, W., Wang, H., & Song, M. (2015). *The Data Science Handbook*. Harvard Business Review Press.

Sharma, R. (2021). *Practical Machine Learning with Jupyter Notebook*. Packt Publishing.

Shen, H., Welch, W. J., & Hughes-Oliver, J. M. (2011). Efficient, adaptive cross-validation for tuning and comparing models, with application to drug discovery. *Annals of Applied Statistics*, *5*(4), 2668 – 2687. https://doi.org/10.1214/11-AOAS491

Shmueli, G., & Koppius, O. R. (2011). Predictive analytics in information systems research. *MIS Quarterly*, *35*(3), 553–572.

Shumeiko, D., & Rozora, I. (2021). Handling Missing Values in Machine Learning Regression Problems. *CEUR Workshop Proceedings*, *3106*, 211–219.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill Education.

Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2003). *Designing and*

Managing the Supply Chain: Concepts, Strategies, and Case Studies (2nd ed.). McGraw-Hill.

Singh, P., Jha, M., Sharaf, M., El-Meligy, M. A., & Gadekallu, T. R. (2023). Harnessing a Hybrid CNN-LSTM Model for Portfolio Performance: A Case Study on Stock Selection and Optimization. *IEEE Access*, *11*, 104000–104015. https://doi.org/10.1109/ACCESS.2023.3317953

Smalter, A., Huan, J., Yi, J., & Lushington, G. H. (2008). *GPD: A Graph Pattern Diffusion Kernel for Accurate Graph Classification with Applications in Cheminformatics*.

Solove, D. J. (2021). *Understanding Privacy*. Harvard University Press.

Song, H. (2024). *Applied Graph Data Science: Graph Algorithms and Platforms, Knowledge Graphs, Neural Networks, and Applied Use Cases*. Elsevier. https://shop.elsevier.com/books/applied-graph-data-science/raj/978-0-443-29654-3

Stackowiak, R. (2020). *Visualizing Data with Tableau: An Introduction to Data Visualization, Dashboards & Storytelling*. Addison-Wesley. https://www.pearson.com/store/p/visualizing-data-with-tableau/P100000270363

Strang, G. (2016). *Introduction to Linear Algebra*. Wellesley-Cambridge Press.

Strauch, C. (2012). *NoSQL Databases*.

Suri, N., & Cabri, G. (2014). *Adaptive, Dynamic, and Resilient Systems*. CRC Press, Taylor & Francis Group.

Suyanto, Kurniawan Nur Ramadhani, & Kurniawan Nur Ramadhani. (2019). *Deep Learning Modernisasi Machine Learning untuk Big Data*. Penerbit Informatika. https://inlislite.undiksha.ac.id/opac/detail-opac?id=14625

Tableau. (2023). *Tableau Documentation*.

Tanimura, C. (2021). SQL for data analysis : advanced techniques for transforming data into insights. In *O'Reilly Media, Inc.* O'Reilly Media.

Teate, R. M. P. (2021). *SQL for Data Scientists. A Beginner's Guide for Building Datasets for Analysis*. Wiley.

Tene, O., & Polonetsky, J. (2013). Big data for all: Privacy and user control in the age of analytics. *Northwestern Journal of Technology and Intellectual Property*, *11*(5), 239–273.

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications*, *30*(2), 169–190. https://doi.org/10.3233/AIC-170729

Thompson, S. K. (2012). *Sampling*. John Wiley \& Sons.

Tranquillin, M., Lakshmanan, V., & Tekiner, F. (2023). *Architecting Data and Machine Learning Platforms: Enable Analytics and AI-Driven Innovation in the Cloud*. O'Reilly Media, Inc. https://books.google.com/books/about/Architecting_Data_and_Machine_Learning_P.html?id=T4jcEAAAQBAJ

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science \& Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Verma, V. K., Saxena, K., & Banodha, U. (2024). Analysis Effect of K Values Used in K Fold Cross Validation for Enhancing Performance of Machine Learning Model with Decision Tree. *Communications in Computer and Information Science*, *2053 CCIS*, 374 – 396. https://doi.org/10.1007/978-3-031-56700-1_30

Voigt, P., & dem Bussche, A. (2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide* (1st ed.). Springer.

Voyles, I. T., & Roy, C. J. (2014). Evaluation of model validation techniques in the presence of uncertainty. *16th AIAA Non-Deterministic Approaches Conference*. https://doi.org/10.2514/6.2014-0120

Wang, C., & Wang, Z.-H. (2020). A network-based toolkit for evaluation and intercomparison of weather prediction and climate modeling. *Journal of Environmental Management*, *268*. https://doi.org/10.1016/j.jenvman.2020.110709

Wang, S. (2023). A Stock Price Prediction Method Based on BiLSTM and Improved Transformer. *IEEE Access*, *11*, 104211–104223. https://doi.org/10.1109/ACCESS.2023.3296308

Wang, Z., Zhu, J., & Zhang, H. (2021). Explainable AI for Medical Applications. *IEEE Transactions on Medical Imaging*. https://doi.org/10.1109/TMI.2021.3068253

Wedel, M., & Kamakura, W. A. (2000). *Market Segmentation: Conceptual and Methodological Foundations*. Springer.

White, T. (2015). *Hadoop: The Definitive Guide* (4th ed.). O'Reilly Media.

Wickham, H., & Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.

Yellapu, V. (2018). Descriptive statistics. *International Journal of Academic Medicine*, *4*(1). https://doi.org/10.4103/IJAM.IJAM

Zaharia, M., Wendell, P., Konwinski, A., & Karau, H. (2015). *Learning Spark: Lightning-Fast Big Data Analysis*. O'Reilly Media.

Zhang, C., Sjarif, N. N. A., & Ibrahim, R. (2024). 1D-CapsNet-LSTM: A deep learning-based model for multi-step stock index forecasting. *Journal of King Saud University - Computer and Information Sciences*, *36*(2). https://doi.org/10.1016/j.jksuci.2024.101959

Zhiyanov, A. (2023). *SQL Window Functions Explained*. Independently published.

Zikopoulos, P. C., & Eaton, C. (2012). *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill.

Zuboff, S. (2019). *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. PublicAffairs.

Zwitter, A. (2014). Big Data Ethics. *Big Data \& Society*, *1*(2), 1–6.

Biodata Penulis

Nova Agustina, S.T., M.Kom., lahir di Bandung, Jawa Barat, pada 16 Agustus 1993. Memiliki pengalaman dalam pengembangan aplikasi, khususnya di bidang Android, *Website*, dan *Data Science*. Dengan latar belakang pendidikan Teknik Informatika dari Sekolah Tinggi Teknologi Bandung dan Universitas Amikom Yogyakarta, telah mengajar selama lebih dari tujuh tahun di bidang Pemrograman Mobile, Pemrograman Web, dan *Artificial Intelligence*. Selain itu, aktif dalam penelitian di bidang *Machine Learning* dan telah mempublikasikan beberapa karya ilmiah. Saat ini mengajar di Universitas Teknologi Bandung, yang sebelumnya dikenal sebagai Sekolah Tinggi Teknologi Bandung, serta berkomitmen untuk berbagi pengetahuan dan menerapkan keahlian di dunia akademis dan industri.

Email: nova@utb-univ.ac.id

Ni Nyoman Utami Januhari, SH., M.Kom. Lahir di Denpasar, Bali, pada 25 Januari 1983. Meraih gelar Sarjana Hukum pada Universitas Udayana tahun 2005 dan menyelesaikan program Magister Teknik Informatika (ERESHA) pada tahun 2012. Saat ini, bekerja sebagai dosen pada Program Studi Sistem Komputer, Fakultas Informatika dan Komputer pada ITB STIKOM Bali. Selain berperan sebagai pengajar, aktif sebagai tenaga ahli dalam berbagai proyek audit dan keamanan informasi, serta pengadaan sistem informasi di instansi pemerintah daerah dan lembaga pendidikan tinggi. Beberapa proyek penting yang pernah digarap antara lain Audit Sistem Informasi Pemerintah Daerah, Proyek Pengadaan Sistem Informasi. Telah menulis berbagai modul pelatihan dan materi seminar dengan topik manajemen risiko, transformasi digital dan data digital. Aktif dalam penelitian dan pengabdian kepada masyarakat, dengan fokus pada pengelolaan

arsip digital, strategi transformasi digital, dan audit sistem informasi. Berbagai publikasi telah dimuat dalam jurnal nasional dan internasional.
Email: amik@stikom-bali.ac.id

Dr. Padrul Jana, S.Pd. M. Sc. adalah dosen program studi pendidikan matematika, Fakultas Keguruan dan Ilmu Pendidikan, Universitas PGRI Yogyakarta. Fokus penelitiannya adalah pada pendidikan matematika, penerapan matematika, dan statistika khususnya Fuzzy Portofolio. Aktif dalam kegiatan pengajaran, penelitian, dan pengabdian kepada masyarakat. Selain itu, beliau juga berperan dalam kegiatan Merdeka Belajar Kampus Merdeka (MBKM). Meliputi kegiatan Pertukaran Mahasiswa Merdeka (PMM), Kampus Mengajar (KM), Kegiatan Magang dan Studi Independen Bersertifikat (MSIB), dan Program Kreativitas Mahasiswa (PKM). Penulis dapat dihubungi melalui email: padrul.jana@upy.ac.id.

I Ketut Dedy Suryawan, S.Kom., M.Kom., lahir di Tabanan, Bali, pada 29 Mei 1978. Berlatar belakang pendidikan Teknik Informatika dari Universitas Kristen Duta Wacana Yogyakarta dan STMIK Eresha Jakarta, memberikan peluang dan kesempatan menjadi dosen pengajar di ITB STIKOM Bali sampai saat ini. Memiliki pengalaman dalam menangani kerjasama pengembangan aplikasi maupun konsultan di berbagai lembaga, pemerintah daerah dan kementrian. Beberapa proyek penting yang pernah dikerjakan antara lain Tim Ahli ranperda SPBE Kabupaten Badung, Tim penyusunan Kebijakan Persandian dan Keamanan Informasi Kota Denpasar, Tim Audit IT Inspektorat kabupaten Badung, Tim Pengembangan Sistem perijinan Online di Kabupaten Badung dan Propinsi Bali, tim Pengembangan Aplikasi E-Planning Bapelitbang Kabupaten Tabanan, Tim Pengembangan Aplikasi Layanan Angkutan Sekolah Terintegrasi Dinas Perhubungan Kota Denpasar dan lainnya. Selain itu, aktif dalam pengabdian dan penelitian dengan fokus kepada sistem informasi, *software quality*, big data dan telah mempublikasikan beberapa karya ilmiah.
Email: Dedymeng@gmail.com

Dr. Meilany Nonsi Tentua, S.Si., M.T., lahir di Jakarta, pada 12 Mei 1973. Memiliki pengalaman dalam Kecerdasan Buatan, khususnya di bidang *Machine Learning*, *Data Science, Natural Language Processing, Image Processing* dan *Signal Processing*. Dengan latar belakang pendidikan Ilmu Komputer dari Universitas Gadjah Mada, telah mengajar selama lebih dari delapan belas tahun di bidang *Artificial Intelligence*. Selain itu, aktif dalam penelitian di bidang *Machine Learning*, *Data Science, Natural Language Processing* dan *Image Processing* dan telah mempublikasikan beberapa karya ilmiah. Saat ini mengajar di Universitas PGR Yogyakarta, serta berkomitmen untuk berbagi pengetahuan dan menerapkan keahlian di dunia akademis dan industri. Email: meilany@upy.ac.id

Ester Lumba, S.Si., M.Kom., lahir di Palu, Sulawesi Tengah, pada 07 Agustus 1971. Meraih gelar Sarjana Ilmu Komputer dari Fakultas MIPA Universitas Kristen Immanuel (UKRIM) Yogyakarta pada tahun 1998 dan menyelesaikan program Magister Ilmu Komputer di Universitas Budi Luhur Jakarta pada tahun 2008. Saat ini, bekerja sebagai dosen pada beberapa perguruan tinggi swasta di JABODETABEK. Menjadi Trainer independent di instansi pemerintah maupun swasta. Terlibat dalam pengembangan aplikasi perangkat lunak sebagai sistem analis dan project manager untuk aplikasi bisnis.

Email: estlumba@gmail.com

Nuk Ghurroh Setyoningrum, S.Kom., M.Cs., lahir di Semarang, Jawa Tengah pada 23 Agustus 1984. Adalah alumni Sarjana Komputer dari Universitas Stikubank Semarang pada tahun 2007 di program studi Sistem Informasi Fakultas Teknologi Informasi dan menyelesaikan program Magister ilmu Komputer di Universitas Gadjah Mada Yogyakarta pada tahun 2010 dengan mengambil konsentrasi Ilmu Komputer Fakultas MIPA. Sekarang Sedang menempuh studi Doktoral Informatika di Universitas AMIKOM Yogyakarta. Penulis mengabdi sebagai Dosen Tetap di Universitas Cipasung Tasikmalaya dan aktif mengajar sebagai Tutor di Universitas Terbuka sejak tahun 2019 sampai sekarang. Beberapa buku sudah dituliskan seperti Sistem Informasi Manajemen, Keamanan Sistem Informasi, Analisa Sistem Informasi. Motivasi penulis untuk terus berkarya dalam bidang pendidikan agar turut serta memajukan pendidikan di Indonesia.

Email: nuke@uncip.ac.id

R. Hafid Hardyanto, M. Pd., lahir di Sleman, Yogyakarta, pada 5 Desember 1987. Meraih gelar Sarjana Pendidikan Teknik Mekatronika dari Fakultas Teknik Universitas Negeri Yogyakarta pada tahun 2021 dan menyelesaikan program Magister Pendidikan Teknologi Kejuruan bidang Studi Teknologi Informasi di Universitas Negeri Yogyakarta pada tahun 2015. Saat ini, bekerja sebagai dosen pada Program Sarjana Informatika Fakultas Sains dan Teknologi Universitas PGRI Yogyakarta (UPY). Penulis juga aktif dalam penelitian dan pengabdian kepada masyarakat, dengan fokus pada IoT, Jaringan Sensor, dan Robotika. Berbagai publikasi telah dimuat dalam jurnal nasional dan internasional.

Email: hafid@upy.ac.id

Firdiyan Syah, S.Kom., M.Kom., lahir di Yogyakarta pada 31 Juli 1990. Memiliki pengalaman di bidang pengolahan citra digital, kecerdasan buatan, serta pengembangan aplikasi berbasis teknologi. Dengan latar belakang pendidikan Teknik Informatika dari STMIK AMIKOM Yogyakarta dan Universitas Amikom Yogyakarta, telah mengajar selama lebih dari lima tahun di bidang Signal and Image Processing, Technopreneur, dan Human-Computer Interaction. Selain itu, aktif dalam penelitian yang berfokus pada metode deteksi wajah, pengembangan sistem berbasis Android, serta penerapan kecerdasan buatan dalam berbagai aspek teknologi. Saat ini, ia bekerja sebagai dosesn di Universitas PGRI Yogyakarta dan telah mempublikasikan berbagai karya ilmiah di jurnal nasional dan internasional. Selain penelitian, ia juga aktif dalam pengabdian kepada masyarakat melalui berbagai pelatihan digital, seperti pembuatan video pembelajaran untuk sekolah luar biasa dan peningkatan kualitas produk bagi pengrajin lokal melalui digital marketing.
Email: firdiyan@upy.ac.id

Assoc. Prof. Ir. Nizirwan Anwar, MT, IPM, Ph.D (C), ASEAN.Eng saat ini berkarir sebagai Dosen Tetap Jenjang Studi Strata Satu (S1) Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Esa Unggul di bawah naungan Yayasan Pendidikan Kemala Bangsa (YPKB). Penulis lahir di kota Bandung (Jawa Barat) tanggal 24 Juli 1964, menyelesaikan pendidikan S1 dari Program Studi Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran Bandung, Jawa Barat (1989) dan melanjutkan Program Studi Teknik Elektro Fakultas Teknik (dh Program Studi Pascasarjana) Universitas Indonesia, Depok Jawa Barat (dh DKI Jakarta) menyelesaikan studinya (1995). Candidate Ph.D (ICT) Programme Research Area Data Mining at Asia eUniversity (AeU). Dan bidang profesional memperoleh gelar Insinyur Profesional Madya (IPM) tahun 2022; *Certified* ASEAN *Engineer* (ASEAN.Eng) tahun 2023. Riwayat singkat perjalanan karir penulis sebagai seorang pendidik (dosen) dalam melaksanakan dan mengabdikan pada Tri Dharma Perguruan Tinggi sudah 35 tahun – sekarang. Alhamdulillah telah

menghasilkan beberapa buku referensi; karya ilmiah ter-publikasi (ISSN/ISBN dan ter-indeks (DOI) serta bereputasi (Nasional/ Internasional), penelitian dalam bidang kajian *Internet of Things* (IoT), Digital Forensik, Kriptografi, Bibliometrik dan *Data Science*. Hingga pernah/saat ini aktif di beberapa organisasi profesi (IAII, ADI, ACM, IEEE, ASIOTI, BKI PII, APTIKOM, IAENG), Asesor Dosen Nasional BKD (NIRA), Reviewer Jurnal Ilmiah Terakreditasi SINTA (Kemendikbud DIKTI); Komite Ilmiah Konferensi (Nasional/Internasional), Pengurus Badan Kejuruan Informatika (BKI) dan Asesor Majelis Uji Kompetensi (MUK) Persatuan Insinyur Indonesia (PII) (2021- saat ini).

Email: nizirwan.anwar@esaunggul.ac.id

I Made Adi Purwantara, S.T., M.Kom., lahir di Tabanan, Bali, pada 14 Oktober 1980. Memiliki pengalaman dalam pengembangan aplikasi, khususnya di bidang Android, Desktop dan Website. Dengan latar belakang pendidikan Jurusan Teknik Informatika dari Universitas Pembangunan Nasional Veteran Yogyakarta dan STMIK Eresha Jakarta, telah mengajar sejak tahun 2005. Selain itu, aktif dalam penelitian di bidang *Deep Learning* dan telah mempublikasikan beberapa karya ilmiah. Berpengalaman mengajar di kampus Institut Teknologi dan Bisnis STIKOM Bali dan di kampus Politeknik Nasional.

Email: adipurwa@gmail.com

Muhammad Fairuzabadi, S.Si., M.Kom., lahir di Bulukumba, Sulawesi Selatan, pada 26 September 1974. Meraih gelar Sarjana Ilmu Komputer dari Fakultas MIPA Universitas Kristen Immanuel (UKRIM) Yogyakarta pada tahun 1998 dan menyelesaikan program Magister Ilmu Komputer di Universitas Gadjah Mada pada tahun 2006. Saat ini, bekerja sebagai dosen pada Program Sarjana Informatika Fakultas Sains dan Teknologi Universitas PGRI Yogyakarta (UPY). Selain berperan sebagai pengajar, aktif sebagai tenaga ahli dalam berbagai proyek pengembangan sistem informasi, master plan, dan blueprint teknologi informasi dan komunikasi (TIK) di instansi pemerintah daerah, kementerian, serta lembaga. Beberapa proyek

penting yang pernah digarap antara lain Masterplan Jogja Smart Province, Cetak Biru Sistem Informasi Terintegrasi PDLKWS Kementerian Kehutanan dan Lingkungan Hidup, serta Rencana Induk Sistem Pemerintahan Berbasis Elektronik (SPBE) Pemerintah Kota Metro. Telah menulis lebih dari 25 buku dengan topik sistem informasi, artificial intelligence (AI), data science, dan deep learning. Aktif dalam penelitian dan pengabdian kepada masyarakat, dengan fokus pada pengembangan sistem informasi, sistem pakar, dan data science. Berbagai publikasi telah dimuat dalam jurnal nasional dan internasional.

Email: fairuz@upy.ac.id

Prahenusa Wahyu Ciptadi, S.T., M.T., lahir di Ponorogo pada 27 Desember 1984. Menyelesaikan pendidikan S1 Teknik Telekomunikasi di STT Telkom Bandung pada tahun 2008 dan melanjutkan studi S2 Teknologi Informasi di Institut Teknologi Bandung (ITB) yang diselesaikan pada tahun 2010. Saat ini, menjabat sebagai Dosen Tetap di Program Studi Informatika, Universitas PGRI Yogyakarta (UPY). Berbekal pengalaman profesional di berbagai perusahaan ternama seperti PT Telkom, PT INTI, dan TechMahindra Indonesia, memiliki keahlian di bidang Teknologi Internet of Things (IoT), sistem informasi, serta jaringan komunikasi. Sejak bergabung di UPY pada Februari 2016, aktif mengembangkan ilmu pengetahuan dan teknologi melalui penelitian dan pengabdian masyarakat. Sebagai peneliti dan akademisi, berbagai hasil karyanya telah dipublikasikan dalam jurnal dan prosiding ilmiah nasional maupun internasional, seperti *IOP Conference Series* dan *Journal of Physics*. Fokus penelitian mencakup penerapan IoT untuk budidaya hidroponik, smart aquaponics, sistem keamanan rumah berbasis IoT, serta sistem monitoring dan kontrol berbasis sensor. Selain itu, turut berkontribusi dalam pengembangan aplikasi teknologi informasi yang mendukung pendidikan dan kehidupan sehari-hari.

Email: nusa@upy.ac.id