



DESAIN WEB

Menguasai **HTML**, **CSS**,
JavaScript, dan **Bootstrap**
untuk Pemula

Muhammad Fairuzabadi • Oktoverano Hendrik Lengkong
Ferawaty • Ida Wahyuni • Indah Clara Sari • Janner Simarmata
Setia Wardani • Syarifah Fitrah Ramadhani
Reymon Rotikan • Stenly Adam • Pujianti Wahyuningsih
Billy Eden William Asrul • Suwatri Jura



DESAIN WEB

Menguasai **HTML**, **CSS**,
JavaScript, dan **Bootstrap**
untuk Pemula

UU 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Perlindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- a. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- b. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- c. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- d. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).

Desain Web: Menguasai HTML, CSS, JavaScript, dan Bootstrap untuk Pemula

Muhammad Fairuzabadi, Oktoverano Hendrik Lengkong Ferawaty, Ida
Wahyuni, Indah Clara Sari, Janner Simarmata
Setia Wardani, Syarifah Fitrah Ramadhani
Reymon Rotikan, Stenly Adam, Pujianti Wahyuningsih
Billy Eden William Asrul, Suwatri Jura



Penerbit Yayasan Kita Menulis

Desain Web: Menguasai HTML, CSS, JavaScript, dan Bootstrap untuk Pemula

Copyright © Yayasan Kita Menulis, 2023

Penulis:

Muhammad Fairuzabadi, Oktoverano Hendrik Lengkong
Ferawaty, Ida Wahyuni, Indah Clara Sari, Janner Simarmata
Setia Wardani, Syarifah Fitrah Ramadhani
Reymon Rotikan, Stenly Adam, Pujianti Wahyuningsih
Billy Eden William Asrul, Suwatri Jura

Editor: Matias Julyus Fika Sirait

Desain Sampul: Devy Dian Pratama, S.Kom.

Penerbit

Yayasan Kita Menulis

Web: kitamenulis.id

e-mail: press@kitamenulis.id

WA: 0821-6453-7176

IKAPI: 044/SUT/2021

Muhammad Fairuzabadi., dkk.

Desain Web: Menguasai HTML, CSS, JavaScript, dan Bootstrap untuk
Pemula

Yayasan Kita Menulis, 2023

xx; 206 hlm; 16 x 23 cm

ISBN: 978-623-342-941-2

Cetakan 1, Agustus 2023

- I. Desain Web: Menguasai HTML, CSS, JavaScript, dan
Bootstrap untuk Pemula
- II. Yayasan Kita Menulis

Katalog Dalam Terbitan

Hak cipta dilindungi undang-undang

Dilarang memperbanyak maupun mengedarkan buku tanpa
izin tertulis dari penerbit maupun penulis

Kata Pengantar

Selamat datang dalam buku ini yang akan memperkenalkan Anda pada dunia desain web menggunakan HTML, CSS, JavaScript, dan Bootstrap. Bagi pemula yang ingin menguasai dasar-dasar pembuatan halaman web yang menarik dan responsif, buku ini merupakan panduan yang tepat.

Bagian pertama akan membawa Anda dalam perjalanan mengenal HTML (Hypertext Markup Language), bahasa markah yang mendasari setiap halaman web di internet. Anda akan mempelajari cara membuat daftar, tabel, mengatur hiperlink, menampilkan gambar, dan memasukkan media ke dalam halaman web Anda.

Setelah Anda menguasai HTML, bab berikutnya akan memperkenalkan Anda pada CSS (Cascading Style Sheets). Dengan CSS, Anda akan dapat memberikan tampilan yang menarik dan estetis pada halaman web Anda. Anda akan memahami tentang model kotak dan tata letak untuk menyusun elemen-elemen secara visual.

Bagian ketiga fokus pada JavaScript, bahasa pemrograman yang memberikan interaktivitas dan dinamisme pada halaman web Anda. Anda akan mempelajari konsep variabel, tipe data, operator, serta bagaimana mengontrol aliran dan perulangan di dalam program JavaScript. Fungsi dan objek juga akan dijelaskan dengan jelas untuk memberikan pemahaman yang komprehensif.

Terakhir, bagian keempat akan mengenalkan Anda pada Bootstrap, kerangka kerja (framework) populer untuk membangun tata letak responsif dengan mudah. Anda akan memahami konsep-konsep penting

seperti sistem grid Bootstrap dan berbagai komponen siap pakai yang mempercepat proses desain web Anda.

Dengan mengikuti buku ini secara terstruktur, diharapkan Anda akan memiliki landasan yang kokoh dalam desain web dan dapat membangun halaman web yang menarik, interaktif, dan ramah perangkat seluler. Selamat belajar dan semoga Anda sukses dalam menguasai HTML, CSS, JavaScript, dan Bootstrap!

Penulis

Daftar Isi

Kata Pengantar	v
Daftar Isi	vii
Daftar Gambar	xiii
Daftar Tabel.....	xix

BAGIAN 1: HTML (Hypertext Markup Language):

Bab 1 Pengenalan Desain Web dan HTML

1.1 Situs Web	1
1.1.1 Sejarah Situs Web.....	2
1.1.2 Situs Web Statis	3
1.1.3 Situs Web Dinamis	4
1.2 Desain Web.....	5
1.2.1 Perkembangan Desain Web.....	6
1.2.2 Prinsip Desain Web yang Efektif	9
1.3 Pengenalan HTML	11
1.3.1 Sejarah Perkembangan HTML.....	11
1.3.2 Struktur Dasar HTML	12
1.3.3 Heading.....	14
1.3.4 Elemen Paragraph dan Line Break.....	16
1.3.5 Elemen Penformatan	18

Bab 2 Daftar dan Tabel

2.1 Pendahuluan.....	21
2.2 Tabel dalam HTML	21
2.2.1 Tabel secara umum.....	21
2.2.2 Tabel Border.....	23
2.3 List dalam HTML.....	28
2.3.1 HTML: Unordered List	28
2.3.2 HTML: Ordered List	29
2.3.3 HTML: Description List.....	30

Bab 3 Hyperlink, Gambar dan Media

3.1 Hyperlink.....	33
3.2 Jenis Link	35
3.3 Atribut Link.....	37
3.4 Link dengan Gambar.....	38
3.5 Link Ke Alamat Email	39
3.6 Menampilkan Gambar	40
3.7 Ukuran Gambar	42
3.8 Audio	44
3.9 Format Audio	45
3.10 Atribut Audio.....	46
3.10.1 Controls	46
3.10.2 Autoplay.....	47
3.10.3 Loop.....	47
3.10.4 Muted	48
3.11 Video di HTML.....	48

Bab 4 Formulir dan Elemen Input

4.1 Menggunakan Elemen Input Dasar.....	51
4.1.1 Input Teks (Text Input)	51
4.1.2 Input Kata Sandi (Password Input)	52
4.1.3 Input Angka (Number Input).....	52
4.1.4 Input Pilihan (Radio Button Input)	52
4.1.5 Input Kotak Centang (Checkbox Input)	53
4.1.6 Input Tanggal (Date Input)	53
4.2 Membuat Formulir Kontak.....	53
4.2.1 Validasi Input dengan HTML5 Atribut.....	56
4.2.2 Menggunakan Pseudoclass untuk Visual Validasi	58
4.3 Menerapkan CSS pada Formulir.....	59
4.3.1 Menggunakan Pseudo-Element untuk Styling.....	63
4.3.2 Membuat Tombol Submit yang Menarik.....	64
4.4 Formulir dengan Beberapa Kolom.....	66

BAGIAN 2: CSS (Cascading Style Sheets):**Bab 5 Pengenalan CSS**

5.1 Pengenalan Dasar CSS.....	73
5.1.1 Sejarah dan Perkembangan CSS	74
5.1.2 Perkembangan Versi CSS	74
5.1.3 Fungsi dan Kegunaan CSS	75

5.1.4 Cara Kerja CSS.....	76
5.2 Cara Menulis CSS di HTML.....	77
5.2.1 Internal CSS	77
5.2.2 Eksternal CSS	78
5.2.3 Inline CSS	78
5.3 Sintaks Dasar CSS.....	79
5.3.1 Struktur Dasar CSS.....	79
5.3.2 Penulisan Komentar di CSS.....	80
5.4 Memahami Selector di CSS.....	81
5.5 Menggunakan Warna di CSS	82
5.5.1 Mengetahui Properti Color.....	82
5.5.2 Nilai yang Valid untuk Warna.....	82

Bab 6 Model Kotak dan Tata Letak

6.1 Model Kotak	85
6.1.1 Area Konten	86
6.1.2 Area Padding.....	86
6.1.3 Area Border.....	86
6.1.4 Area Margin.....	86
6.2 Tata Letak Kotak Fleksibel CSS	89

BAGIAN 3: JavaScript:

Bab 7 Pengenalan Java Script

7.1 Apa itu Java Script ?.....	91
7.1.1 Sejarah dan Perkembangan JavaScript	91
7.1.2 Penerapan JavaScript dalam Pengembangan Web	94
7.2 Penyisipan JavaScript pada Halaman HTML	95
7.2.1 Menggunakan Script Tag.....	95
7.2.2 Menyisipkan File JavaScript Eksternal.....	97
7.3 Variabel dan Tipe Data dalam JavaScript.....	99
7.3.1 Mendeklarasikan Variabel dengan Benar.....	99
7.3.2 Memahami Berbagai Tipe Data (String, Number, Boolean, Array, Object).....	100
7.4 Menggunakan Komentar pada Kode JavaScript.....	102
7.4.1 Jenis Komentar dalam JavaScript	102
7.4.2 Penggunaan Komentar untuk Dokumentasi Kode	103

Bab 8 Variabel, Tipe Data, dan Operator

8.1 Variabel	105
8.2 Tipe Data	109
8.2.1 Tipe data String	109
8.2.2 Tipe data Numerik	110
8.2.3 Tipe data BigInt.....	110
8.2.4 Tipe data Boolean	110
8.2.5 Tipe data Undefined.....	111
8.2.6 Tipe data Null	111
8.2.7 Tipe Data Symbol	111
8.2.8 Tipe Data Object	112
8.2.9 Tipe Data Array.....	113
8.3 Operator.....	114
8.3.1 Operator Aritmatika	115
8.3.2 Operator Penugasan	116
8.3.3 Operator Logika	116
8.3.4 Operator Perbandingan	117

Bab 9 Aliran Kontrol dan Perulangan

9.1 Pengambilan Keputusan	119
9.1.1 if	120
9.1.2 if..else	121
9.1.3 Ternary Operator: "?"	122
9.1.4 Switch	123
9.2 Pengulangan.....	125
9.2.1 while.....	125
9.2.2 do..while	126
9.2.3 for	127
9.2.4 for..of.....	128
9.2.5 for..in.....	130

Bab 10 Fungsi dan Objek

10.1 Pengertian dan Pembuatan Fungsi	133
10.1.1 Mengetahui Konsep Fungsi dalam Pemrograman	133
10.1.2 Mendeklarasikan Fungsi.....	135
10.2 Menggunakan Fungsi Bawaan JavaScript.....	136
10.2.1 Menggunakan Fungsi Math.....	136
10.2.2 Menggunakan Fungsi String	137
10.2.3 Menggunakan Fungsi Date.....	139

10.3 Objek Dalam JavaScript	141
10.3.1 Konsep Object Dalam JavaScript	141
10.3.2 Membuat Object Dengan Object Literal dan Constructor	142
10.4 Menerapkan Fungsi pada Object.....	143
10.4.1 Membuat Metode Pada Object.....	143
10.4.2 Penggunaan “this“ Dalam Metode Object	144

BAGIAN 4 Bootstrap:

Bab 11 Pengenalan Bootstrap

11.1 Pendahuluan.....	145
11.2 Sejarah Singkat Bootstrap.....	146
11.3 Kelebihan dan Kekurangan Bootstrap	148
11.4 File Utama Bootstrap	151
11.5 Cara Penggunaan Bootstrap	152
11.5.1 Menggunakan CDN	152
11.5.2 Download Secara Manual	155

Bab 12 Sistem Grid Bootstrap

12.1 Memahami Sistem Grid pada Bootstrap	157
12.2 Konsep dan Struktur Grid.....	159
12.3 Container dan Container Fluid	163
12.4 Struktur Dasar Grid dalam Bootstrap	165
12.4.1 Membuat Row dan Col	165
12.4.2 Pengaturan Jumlah Kolom dan Lebar Kolom	166
12.5 Mengatur Tampilan Responsif dengan Grid System.....	168
12.5.1 Menggunakan Breakpoint untuk Perangkat Berbeda.....	168
12.5.2 Menyusun Tata Letak Responsif dengan Grid System.....	169
12.6 Kustomisasi Grid System	171
12.6.1 Menggunakan Offset dan Push-Pull.....	171
12.6.2 Mengatur Margin dan Padding dalam Grid	172

Bab 13 Komponen Bootstrap

13.1 Komponen Bootstrap	175
13.1.1 Menggunakan Carousel untuk Slideshow.....	175
13.1.2 Membuat Navigasi yang interaktif (Navbar)	177
13.1.3 Button Groups	179
13.1.4 Pagination	180
13.1.5 Label dan Badges	182
13.1.6 Jumbotron untuk menampilkan pesan	183

13.1.7 Card	182
13.1.8 Form	185
13.1.9 Footer di Bootstrap 5	186
13.1.10 Mengatur Tampilan Gambar.....	188
Daftar Pustaka	191
Biodata Penulis	201

Daftar Gambar

Gambar 1.1: Tim Berners Lee menciptakan World Wide Web	2
Gambar 1.2: Blog Pribadi merupakan contoh Situs Web Dinamis	5
Gambar 1.3: Perkembangan Desain Web.....	6
Gambar 1.4: Contoh Responsive Web Design	7
Gambar 1.5: Contoh elemen web material design	8
Gambar 1.6: Contoh Mobile-First Design	8
Gambar 1.7: Prinsip Desain Web yang Efektif	9
Gambar 1.8: Contoh Penggunaan Heading.....	15
Gambar 1.9: Contoh Penggunaan Paragraph dan Line Break.....	18
Gambar 1.10: Contoh Penformatan dalam HTML	20
Gambar 2.1: Hasil Kode HTML Standard Untuk Tabel Secara Umum	23
Gambar 2.2: Gambaran hasil Style CSS untuk menambahkan garis batas.....	23
Gambar 2.3: Gambaran hasil kode HTML untuk unordered list.....	29
Gambar 2.4: Gambaran hasil kode HTML untuk ordered list	30
Gambar 2.5: Gambaran hasil kode HTML untuk description list.....	31
Gambar 3.1: Syntax Link (Ahmad Muhandian, n.d.).....	33
Gambar 3.2: Tampilan Website.....	34
Gambar 3.3: Website Google	34
Gambar 3.4: Tampilan Utama Internal Link	35
Gambar 3.5: Tampilan Halaman Contact	35
Gambar 3.6: External Link.....	36
Gambar 3.7: Tampilan Google pada External Link.....	36
Gambar 3.8: Tampilan Website menggunakan Attribute Target = _blank	38
Gambar 3.9: Tampilan Website Google	38
Gambar 3.10: Link menggunakan Gambar	39
Gambar 3.11: Hasil Tampilan Link menggunakan Gambar	39
Gambar 3.12: Tampilan Link ke Email	40
Gambar 3.13: Hasil Tampilan Link ke Email.....	40
Gambar 3.14: Perintah untuk Menampilkan Gambar.....	40
Gambar 3.15: Teddy	41
Gambar 3.16: Style	44

Gambar 3.17: Perintah Audio	45
Gambar 3.18: Audio	46
Gambar 3.19: Autoplay	47
Gambar 3.20: Perintah Penambahan Video.....	48
Gambar 3.21: Winnie The Pooh.....	49
Gambar 4.1: Contoh Formulir Kontak.....	54
Gambar 4.2: Contoh Formulir Dengan CSS	60
Gambar 4.3: Contoh Formulir Dengan Beberapa Kolom	67
Gambar 5.1: Cara Kerja CSS.....	76
Gambar 5.2: Contoh Internal CSS.....	77
Gambar 5.3: Contoh Eksternal CSS.....	78
Gambar 5.4: Contoh Inline CSS.....	79
Gambar 5.5: CSS Syntax	79
Gambar 5.6: Struktur Dasar CSS.....	80
Gambar 5.7: Contoh Penulisan Komentar di CSS.....	80
Gambar 5.8: Contoh Memberikan Warna pada Teks.....	82
Gambar 6.1: Model Kotak	85
Gambar 6.2: Contoh 1 Elemen HTML.....	88
Gambar 6.3: Contoh 2 Elemen HTML.....	88
Gambar 8.1: Ilustrasi variabel.....	105
Gambar 8.2: penamaan variabel.....	106
Gambar 8.3: contoh kasus variabel	106
Gambar 8.4: contoh kasus variabel yang benar.....	107
Gambar 8.5: contoh kasus variabel yang salah.....	107
Gambar 8.6: Contoh kasus variabel tipe angka	107
Gambar 8.7: Hasil variabel tipe angka	108
Gambar 8.8: Contoh kasus variabel tipe string.....	108
Gambar 8.9: Hasil source code variabel tipe string.....	108
Gambar 8.10: Tipe data String.....	110
Gambar 8.11: Tipe data Numerik.....	110
Gambar 8.12: Tipe data BigInt	110
Gambar 8.13: Tipe data Boolean.....	111
Gambar 8.14: Tipe data Undefined.....	111
Gambar 8.15: Tipe data Null	111
Gambar 8.16: Tipe data Symbol.....	112
Gambar 8.17: Tipe data Object.....	112
Gambar 8.18: Tipe data Objek 2	113
Gambar 8.19: Tipe data Objek 3	113
Gambar 8.20: Tipe data Objek 4	113

Gambar 8.21: Tipe data Array	114
Gambar 8.22: Mengakses elemen array.....	114
Gambar 8.23: Mengubah elemen array.....	114
Gambar 8.24: Increment Postfix dan Prefix	115
Gambar 8.25: Decrement Postfix dan Prefix.....	116
Gambar 8.26: Operator Logika And	116
Gambar 9.1: Contoh kode program dengan pernyataan if.....	120
Gambar 9.2: Contoh kode program dengan pernyataan if.else	122
Gambar 9.3: Contoh kode program ternary operator	122
Gambar 9.4: Sintaks pernyataan switch (Javascript.info).....	124
Gambar 9.5: Contoh kode program switch.....	124
Gambar 9.6: Contoh implementasi perulangan while.....	126
Gambar 9.7: Contoh implementasi perulangan do..while	127
Gambar 9.8: Contoh implementasi perulangan for	128
Gambar 9.9: Contoh Program Menggunakan Perulangan for..of pada sebuah array	129
Gambar 9.10: Contoh program dengan perulangan for..in.....	130
Gambar 11.1: Logo Bootstrap 5	146
Gambar 11.2: Perkembangan Framework Bootstrap.....	147
Gambar 11.3: Content Delivery Network (CDN).....	153
Gambar 11.4: Membuka Folder	153
Gambar 11.5: Membuat HTML baru.....	153
Gambar 11.6: Source Code pada HTML belajar-bootstrap	154
Gambar 11.7: <head> pada Source Code belajar-bootstrap.....	154
Gambar 11.8: <body> pada Source Code belajar-bootstrap	154
Gambar 11.9: Hasil Source code belajar-bootstrap.....	154
Gambar 11.10: Download Bootstrap	155
Gambar 11.11: Ekstrak File Bootstrap.....	155
Gambar 11.12: Source Code pada file index.html	155
Gambar 11.13: Lokasi alamat file bootstrap.....	156
Gambar 11.14: Tampilan kode CSS bootstrap.....	156
Gambar 12.1: Penggunaan Sintaks “container”	158
Gambar 12.2: Struktur Grid Bootstrap.....	159
Gambar 12.3: Membuat baris atau row.....	160
Gambar 12.4: Membuat baris atau row.....	160
Gambar 12.5: Tampilan Layar Penuh 3 Kolom	161
Gambar 12.6: Tampilan Layar Smartphone 3 Kolom	161
Gambar 12.7: Sintaks membuat 2 Kolom berbeda	161
Gambar 12.8: Tampilan 2 Kolom berbeda Layar Penuh.....	162

Gambar 12.9: Tampilan 2 Kolom berbeda Layar Smartphone.....	162
Gambar 12.10: Menggunakan Break point berbeda ukuran layar	162
Gambar 12.11: Menggunakan Break point berbeda ukuran layar Dekstop .	162
Gambar 12.12: Menggunakan Break point berbeda ukuran layar Smartphone	163
Gambar 12.13: Menggunakan Container dan Container-fluid.....	164
Gambar 12.14: Tampilan menggunakan Container.....	164
Gambar 12.15: Tampilan menggunakan Container-fluid.....	164
Gambar 12.16: Perintah membuat baris/row	165
Gambar 12.17: format membuat column.....	165
Gambar 12.18: Penggunaan column dan breakpoint serta ukuran	165
Gambar 12.19: Tampilan membuat 2 column dan 2 row	166
Gambar 12.20: Sintaks membuat 2 column dan 2 row.....	166
Gambar 12.21: Tampilan membuat column dan row	167
Gambar 12.22: Sintaks membuat 1 row dengan 4 col	167
Gambar 12.23: Tampilan membuat 1 row dengan 4 col	167
Gambar 12.24: Sintaks breakpoint medium (mg) dan large (lg).....	169
Gambar 12.25: Sintaks CSS letak responsif.....	171
Gambar 12.26: Sintaks push dan pull column.....	172
Gambar 12.27: Class pengaturan margin.....	172
Gambar 12.28: Class pengaturan padding	173
Gambar 12.29: Class pengaturan padding	173
Gambar 12.30: Class pengaturan padding	173
Gambar 13.1: Source code membuat slideshow	176
Gambar 13.2: Tampilan slideshow pada halaman website	177
Gambar 13.3: Source Code Navbar	178
Gambar 13.4: Tampilan Navbar	179
Gambar 13.5: Source code pagination	181
Gambar 13.6: Tampilan pagination.....	182
Gambar 13.7: Souce code label	182
Gambar 13.8: Tampilan source code label	182
Gambar 13.9: Tampilan source code dan hasil dari badget.....	182
Gambar 13.10: Source Code Jumbotron.....	183
Gambar 13.11: Tampilan jumbotron.....	183
Gambar 13.12: Souce Code membuat Card	184
Gambar 13.13: Tampilan Card	185
Gambar 13.14: Souce Code membuat form	186
Gambar 13.15: Tampilan form	186
Gambar 13.16: Souce Code membuat footer	187

Gambar 13.17: Tampilan footer	188
Gambar 13.18: Source code mengatur tampilan gambar	189
Gambar 13.19: Tampilan gambar thumbnail, circle dan rounded	190

Daftar Tabel

Tabel 1.1: Ringkasan Sejarah Perkembangan HTML	12
Tabel 2.1: Atribut Tipe urutan dalam list	30
Tabel 3.1: Format Gambar	41
Tabel 3.2: Perbandingan Ukuran Gambar	42
Tabel 3.3: Format Audio	45
Tabel 7.1: Sejarah Perkembangan JavaScript.....	92
Tabel 8.1: Operator Aritmatika (Tarigan, 2020)	115
Tabel 8.2: Operator Penugasan (Wibowo, 2017).....	116
Tabel 8.3: Operator Logika (Albar, 2021)	116
Tabel 8.4: Operator Perbandingan (Wibowo, 2017).....	117
Tabel 11.1: Perbandingan Bootstrap 4 dan 5	150
Tabel 12.1: Ukuran Breakpoint	159
Tabel 13.1: Penggunaan button group	179
Tabel 13.2: Kelas paginaton.....	181

Bab 1

Pengenalan Desain Web dan HTML

1.1 Situs Web

Sebuah situs web (website) adalah kumpulan halaman web yang terhubung satu sama lain dan biasanya berada di server web yang dapat diakses melalui internet atau jaringan lokal. Situs web ini berisi informasi yang disediakan oleh individu, kelompok, atau organisasi (Muttaqin et al., 2023). Seluruh situs yang dapat diakses publik di internet dikenal sebagai World Wide Web (WWW). Meskipun sebagian besar halaman beranda situs web dapat diakses secara bebas, beberapa situs menerapkan pembatasan akses untuk alasan keamanan, privasi, atau tujuan komersial.

Halaman web adalah berkas teks biasa yang ditulis dalam bahasa markup seperti HTML atau XHTML. Berkas tersebut diatur dan dikombinasikan dengan instruksi-instruksi untuk membentuk tampilan halaman yang dapat dilihat di peramban web. Pengguna mengakses halaman web melalui protokol HTTP, tetapi situs web yang lebih aman dapat mengimplementasikan protokol HTTPS untuk meningkatkan keamanan dan privasi.

1.1.1 Sejarah Situs Web

Tim Berners-Lee, seorang ilmuwan komputer, diakui sebagai penemu situs web. Pada tahun 1991, situs web pertama kali muncul sebagai hasil dari usahanya untuk memfasilitasi pertukaran dan pembaruan informasi di antara para peneliti di CERN, tempat dia bekerja. Penggunaan situs web awalnya terbatas pada lingkungan penelitian dan akademis. Namun, seiring berjalannya waktu, situs web mengalami perkembangan pesat dan menjadi salah satu pilar utama *World Wide Web* (WWW). Pada tanggal 30 April 1993, CERN secara resmi mengumumkan bahwa WWW dapat diakses secara gratis oleh publik, membuka pintu bagi semua orang untuk menjelajahi dan berkontribusi dalam ruang informasi global yang baru (Banks & Card, 2008).



Gambar 1.1: Tim Berners Lee menciptakan World Wide Web

Dalam konteks modern, situs web bukan hanya menjadi wadah untuk pertukaran informasi, tetapi juga platform penting untuk berbagai tujuan. Individu, organisasi, dan perusahaan memanfaatkan situs web untuk berbagi pengetahuan, memasarkan produk dan layanan, menyediakan platform komunikasi, dan bahkan sebagai sumber penghasilan melalui iklan atau penjualan online.

Situs web saat ini menawarkan beragam konten, mulai dari teks dan gambar hingga video dan aplikasi interaktif. Penggunaan teknologi HTML, CSS, JavaScript, dan bahasa pemrograman lainnya telah mengubah situs web dari sekadar tampilan teks statis menjadi antarmuka interaktif yang menarik dan fungsional. Bentuk situs web juga sangat beragam. Ada situs web tunggal yang menyajikan informasi sederhana, ada blog yang berfungsi sebagai jurnal

online, ada situs e-commerce yang memfasilitasi pembelian produk, dan ada situs jejaring sosial yang memungkinkan pengguna berinteraksi dan berbagi konten. Selain itu, situs web telah memengaruhi hampir setiap aspek kehidupan kita (Prasetyadi, 2011).

1.1.2 Situs Web Statis

Situs web statis adalah jenis situs web yang kontennya ditampilkan secara tetap atau tidak berubah, tanpa ada interaksi atau dinamika dari pengguna. Halaman-halaman dalam situs web statis telah ditentukan sebelumnya dan tidak ada perubahan yang terjadi saat pengguna mengaksesnya. Setiap kali pengguna mengunjungi situs web statis, ia akan melihat halaman yang sama tanpa adanya interaksi atau kustomisasi.

Karakteristik dan ciri-ciri situs web statis antara lain:

1. **Konten Tetap:** Konten pada situs web statis telah ditentukan dan disiapkan sebelumnya oleh pembuat situs. Tidak ada mekanisme yang memungkinkan perubahan konten secara otomatis atau tergantung pada interaksi pengguna.
2. **Kode Statis:** Halaman-halaman dalam situs web statis biasanya ditulis dengan kode HTML dan CSS. Hal ini berbeda dengan situs web dinamis yang menggunakan bahasa pemrograman seperti PHP, JavaScript, atau Python untuk menghasilkan konten yang dinamis.
3. **Tampilan Konsisten:** Karena konten pada situs web statis tidak berubah, tampilan dan tata letak halaman akan tetap konsisten setiap kali diakses.
4. **Kinerja Cepat:** Karena tidak ada pemrosesan server yang terlibat untuk menghasilkan konten secara dinamis, situs web statis cenderung memiliki kinerja yang lebih cepat dan muat lebih cepat pada peramban.
5. **Keterbatasan Interaktivitas:** Karena sifat statisnya, situs web ini memiliki keterbatasan dalam hal interaktivitas dengan pengguna. Umumnya tidak ada formulir atau elemen interaktif lainnya yang memungkinkan pengguna berinteraksi langsung dengan situs.

1.1.3 Situs Web Dinamis

Situs web dinamis adalah jenis situs web yang memiliki konten yang dapat berubah dan disesuaikan secara dinamis berdasarkan interaksi dan permintaan dari pengguna. Halaman-halaman dalam situs web dinamis dibuat dengan menggunakan bahasa pemrograman seperti PHP, JavaScript, Python, atau ASP.NET untuk menghasilkan konten secara real-time sesuai dengan permintaan pengguna (PawPrints Learning Technologies, 2014). Berbeda dengan situs web statis yang memiliki konten tetap, situs web dinamis memungkinkan pengguna untuk berinteraksi dengan konten, melakukan pencarian, mengisi formulir, berbelanja, dan banyak lagi.

Karakteristik dan ciri-ciri situs web dinamis antara lain:

1. **Konten Dinamis:** Konten pada situs web dinamis dapat berubah secara otomatis berdasarkan permintaan dan interaksi pengguna. Misalnya, konten dapat diperbarui secara real-time dari database atau sumber data lainnya.
2. **Bahasa Pemrograman:** Situs web dinamis menggunakan bahasa pemrograman server-side seperti PHP, Python, atau ASP.NET untuk mengolah permintaan dan menghasilkan konten dinamis.
3. **Interaktivitas:** Situs web dinamis menyediakan elemen interaktif seperti formulir, kotak pencarian, tombol, dan elemen lainnya yang memungkinkan pengguna berinteraksi langsung dengan situs.
4. **Manajemen Konten:** Situs web dinamis biasanya dilengkapi dengan sistem manajemen konten (content management system/CMS) yang memungkinkan pengelola situs untuk dengan mudah mengupdate dan mengelola konten situs secara efisien.
5. **Personalisasi:** Dengan situs web dinamis, pengguna dapat menerima konten yang disesuaikan berdasarkan preferensi atau riwayat mereka. Misalnya, situs e-commerce dapat menampilkan rekomendasi produk yang relevan dengan riwayat pembelian pengguna.
6. **Kinerja Teroptimasi:** Meskipun situs web dinamis memerlukan pemrosesan server untuk menghasilkan konten, upaya dilakukan untuk mengoptimasi kinerja dan waktu muat halaman agar tetap cepat dan responsif.

Beberapa contoh situs web dinamis termasuk platform media sosial, situs e-commerce, situs berita, situs perbankan, dan blog pribadi. Kemampuan untuk memberikan konten yang berubah dan berinteraksi dengan pengguna membuat situs web dinamis menjadi lebih fleksibel dan relevan untuk berbagai kebutuhan dan tujuan.



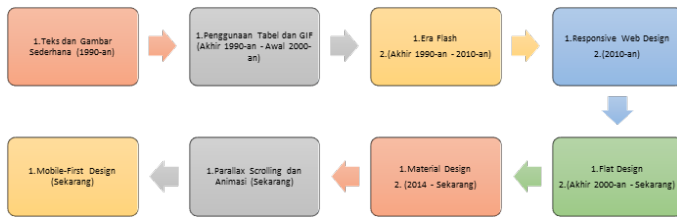
Gambar1.2: Blog Pribadi merupakan contoh Situs Web Dinamis (<http://fairuzelsaid.upy.ac.id>)

1.2 Desain Web

Pengertian Desain Web adalah proses merencanakan, menciptakan, dan mengatur tampilan serta struktur sebuah halaman atau situs web untuk mencapai tujuan tertentu. Desain web mencakup aspek visual dan fungsional dari suatu situs web, termasuk layout, warna, tipografi, grafik, dan interaksi pengguna. Tujuan utama dari desain web adalah memberikan pengalaman yang baik bagi pengguna dan memastikan situs web dapat berfungsi dengan baik.

1.2.1 Perkembangan Desain Web

Perkembangan desain web telah mengalami transformasi yang signifikan sejak awal penggunaan internet hingga saat ini. Berikut adalah beberapa tahap penting dalam perkembangan desain web (Jamsa et al., 2002):



Gambar 1.3: Perkembangan Desain Web

1. Teks dan Gambar Sederhana (1990-an)

Pada awal penggunaan internet, desain web didominasi oleh teks sederhana dan sedikit gambar. Situs-situs web pada masa ini cenderung memiliki tampilan yang sederhana dan minim interaktivitas. HTML masih menjadi bahasa pemrograman utama untuk membuat tampilan dan struktur halaman web.

2. Penggunaan Tabel dan GIF (Akhir 1990-an-Awal 2000-an)

Pada periode ini, penggunaan tabel HTML menjadi populer untuk menyusun tata letak situs web. GIF menjadi format gambar yang sering digunakan untuk animasi dan elemen visual. Situs-situs web mulai memiliki lebih banyak gambar dan elemen grafis.

3. Era Flash (Akhir 1990-an-2010-an)

Adobe Flash menjadi salah satu teknologi utama untuk animasi dan interaktivitas pada situs web. Banyak situs web menggunakan elemen Flash untuk menyajikan konten multimedia dan interaksi yang lebih kompleks. Namun, Flash memiliki keterbatasan dalam hal aksesibilitas dan dukungan perangkat seluler.

4. Responsive Web Design (2010-an)

Dengan semakin banyaknya perangkat seluler yang digunakan untuk mengakses internet, desain web responsif menjadi suatu keharusan. Teknik ini memungkinkan situs web untuk menyesuaikan tampilan dan tata letaknya dengan berbagai ukuran layar, dari desktop hingga

ponsel pintar. Penggunaan CSS media queries menjadi kunci dalam menciptakan desain web responsif.



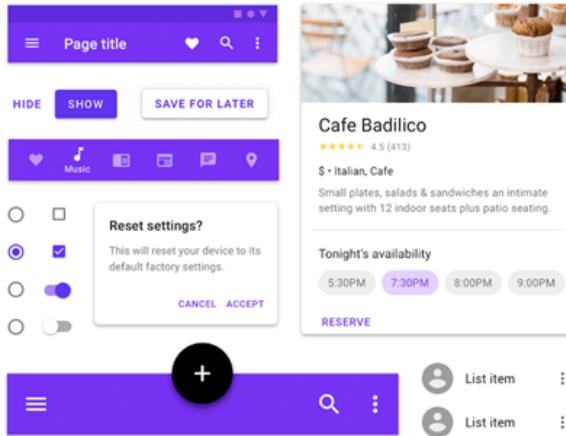
Gambar 1.4: Contoh Responsive Web Design
(<https://www.bluecorona.com/faq/what-is-a-responsive-website-design/>)

5. Flat Design (Akhir 2000-an-Sekarang)

Desain datar (flat design) mulai populer pada akhir 2000-an dan terus berkembang hingga sekarang. Desain ini mengedepankan kesederhanaan, warna solid, dan minimalisme. Flat design menyederhanakan elemen visual dan fokus pada konten yang mudah dibaca dan dipahami oleh pengguna.

6. Material Design (2014-Sekarang)

Material Design dikembangkan oleh Google sebagai panduan desain untuk menciptakan tampilan dan pengalaman yang konsisten di berbagai platform dan perangkat. Material Design menggunakan elemen realistis dan efek bayangan untuk memberikan kesan lapisan dan kedalaman, sambil tetap mempertahankan kesederhanaan desain.



Gambar 1.5: Contoh elemen web material design

(<https://www.justinmind.com/blog/flat-design-vs-material-design-differences/>)

7. Parallax Scrolling dan Animasi (Sekarang)

Parallax scrolling dan animasi yang halus semakin banyak digunakan untuk meningkatkan interaktivitas dan keterlibatan pengguna dalam situs web. Animasi memberikan pengalaman visual yang menarik dan menyenangkan, sementara parallax scrolling memberikan efek kedalaman dan gerak halaman yang lebih dinamis.



Gambar 1.6: Contoh Mobile-First Design (<https://medium.com/swlh/8-excellent-mobile-first-design-examples-for-designer-inspiration-d5c90d465221>)

8. Mobile-First Design (Sekarang)

Seiring pertumbuhan penggunaan perangkat seluler, pendekatan mobile-first design semakin populer. Dalam mobile-first design, desain situs web dipikirkan terlebih dahulu untuk perangkat seluler, kemudian ditingkatkan untuk tampilan desktop. Pendekatan ini memastikan pengalaman pengguna yang optimal pada perangkat seluler, yang semakin dominan dalam akses internet.

1.2.2 Prinsip Desain Web yang Efektif

Desain web yang efektif memastikan bahwa situs web dapat memberikan pengalaman yang baik bagi pengguna, mudah dinavigasi, dan dapat mencapai tujuan yang diinginkan. Berikut adalah beberapa prinsip desain web yang efektif (Mastra & Dharmawan, 2018):



Gambar 1.7: Prinsip Desain Web yang Efektif

1. Kesesuaian dengan Tujuan

Setiap situs web harus memiliki tujuan spesifik, seperti menyampaikan informasi, menjual produk, atau mengumpulkan data pengguna. Prinsip ini menekankan pentingnya desain web yang sesuai dengan tujuan tersebut. Semua elemen desain, termasuk tata letak, warna, dan konten, harus dirancang dengan tujuan tertentu dan mengarahkan pengguna untuk mencapainya.

2. Sederhana dan Intuitif

Desain web yang efektif harus sederhana dan intuitif. Pengguna harus dengan mudah dapat menavigasi situs dan menemukan informasi yang mereka cari tanpa kesulitan. Tata letak yang teratur, menu yang jelas, dan navigasi yang mudah dipahami adalah contoh dari desain web yang sederhana dan intuitif.

3. Responsif dan Adaptif

Penggunaan perangkat seluler semakin meningkat, sehingga desain web harus responsif dan dapat menyesuaikan diri dengan berbagai ukuran layar. Prinsip ini memastikan situs web tetap terlihat dan berfungsi dengan baik pada desktop, tablet, dan ponsel pintar.

4. Kesesuaian Warna

Penggunaan warna yang tepat dapat memengaruhi suasana dan pesan yang ingin disampaikan oleh situs web. Desain web yang efektif mempertimbangkan harmoni warna, kontras yang jelas, dan pemilihan warna yang sesuai dengan merek atau tujuan situs.

5. Konsistensi Desain

Prinsip konsistensi mengacu pada penggunaan elemen desain yang sama di seluruh situs web. Konsistensi dalam tipografi, ikon, tata letak, dan elemen visual lainnya memberikan kesan profesional dan membantu pengguna memahami bagaimana berinteraksi dengan situs.

6. Kecepatan Muat Halaman

Desain web yang efektif mempertimbangkan kecepatan muat halaman. Pengguna cenderung meninggalkan situs yang membutuhkan waktu lama untuk dimuat. Mengoptimalkan gambar dan kode, serta menggunakan teknik caching, adalah beberapa cara untuk meningkatkan kecepatan muat halaman.

7. Tampilan yang Menarik

Desain web yang menarik dan estetik dapat menarik perhatian pengguna dan meningkatkan keterlibatan mereka. Penggunaan elemen visual seperti gambar, video, dan animasi yang tepat dapat memberikan kesan yang menarik.

8. Aksesibilitas

Prinsip aksesibilitas menekankan pentingnya membuat situs web dapat diakses oleh semua pengguna, termasuk mereka yang memiliki keterbatasan fisik atau gangguan penglihatan. Menggunakan label yang deskriptif untuk elemen interaktif, memastikan navigasi dapat diakses dengan keyboard, dan menyediakan transkrip atau alternatif

untuk media adalah beberapa langkah untuk meningkatkan aksesibilitas situs web.

9. Penggunaan Ruang Negatif (Whitespace)

Penggunaan ruang negatif yang tepat memberikan tampilan yang bersih, teratur, dan mudah dibaca. Ruang negatif membantu mengatur tata letak dan memisahkan elemen desain, sehingga pengguna dapat fokus pada konten yang penting.

10. Tes dan Peningkatan Berkelanjutan

Desain web yang efektif melibatkan pengujian dan perbaikan berkelanjutan. Mengumpulkan umpan balik dari pengguna, mengamati perilaku mereka di situs, dan menerapkan perbaikan berdasarkan temuan adalah cara untuk meningkatkan efektivitas desain web.

1.3 Pengenalan HTML

HTML (Hypertext Markup Language) adalah bahasa markup yang digunakan untuk membuat struktur dan konten halaman web. HTML memberikan struktur dan elemen yang diperlukan untuk menampilkan teks, gambar, video, audio, formulir, dan elemen lainnya di halaman web. Bahasa ini berfungsi sebagai tulang punggung dari setiap situs web karena mendefinisikan bagaimana konten akan diorganisasi dan ditampilkan di browser.

1.3.1 Sejarah Perkembangan HTML

HTML terus mengalami perkembangan dan penyempurnaan untuk menjawab kebutuhan situs web modern. HTML5 adalah versi terbaru yang paling luas digunakan dan terus berkembang seiring dengan kemajuan teknologi web. Berikut adalah ringkasan sejarah dan perkembangan HTML dalam bentuk tabulasi (Saputra, 2019):

Tabel 1.1: Ringkasan Sejarah Perkembangan HTML

Tahun	Peristiwa
1989	Tim Berners-Lee, seorang ilmuwan komputer di CERN, menciptakan konsep HyperText.
1990	Tim Berners-Lee menciptakan HTML (Hypertext Markup Language) sebagai bahasa markup pertama untuk halaman web.
1993	HTML versi 1.0 diterbitkan sebagai spesifikasi pertama.
1995	HTML versi 2.0 diperkenalkan, termasuk beberapa fitur baru seperti tabel dan formulir.
1997	HTML versi 3.2 menjadi standar HTML pertama yang diterbitkan oleh W3C.
1999	HTML versi 4.01 dirilis, menyediakan lebih banyak kontrol untuk desainer web.
2000	XHTML (eXtensible Hypertext Markup Language) diperkenalkan sebagai varian XML dari HTML 4.01.
2008	HTML5 mulai dikembangkan oleh W3C untuk menyempurnakan dan memperluas fitur HTML.
2011	W3C mengeluarkan Working Draft untuk HTML5.
2014	HTML5 menjadi rekomendasi resmi W3C, menghadirkan elemen dan atribut baru untuk semantik dan fungsionalitas yang lebih baik.
2016	W3C mengumumkan pengembangan HTML 5.1 dan HTML 5.2 untuk memperbaiki spesifikasi HTML5.
2017	HTML 5.2 menjadi rekomendasi resmi W3C.
2019	W3C mulai mengembangkan HTML 5.3 dan merilis HTML 5.3 sebagai Working Draft.

1.3.2 Struktur Dasar HTML

Struktur dasar dokumen HTML terdiri dari beberapa elemen yang membentuk struktur utama halaman web (Duckett, 2011). Berikut adalah struktur dasar dokumen HTML:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
```

```
<title>Judul Halaman</title>
</head>
<body>
  <!--Konten halaman web akan ditempatkan di sini-->
</body>
</html>
```

Penjelasan untuk setiap bagian struktur dasar dokumen HTML:

1. `<!DOCTYPE html>`
Mendefinisikan tipe dokumen HTML yang digunakan dalam halaman web. Dalam contoh di atas, digunakan tipe dokumen HTML5.
2. `<html>`
Elemen ini merupakan elemen induk dari seluruh halaman web dan menandakan bahwa dokumen adalah dokumen HTML. Semua elemen di dalam halaman web akan ditempatkan di antara tag pembuka `<html>` dan tag penutup `</html>`.
3. `<head>`
Bagian ini berisi informasi meta tentang halaman web, seperti judul (yang ditampilkan di bilah judul peramban), tautan ke stylesheet eksternal, dan meta data lainnya. Informasi di dalam elemen `<head>` tidak ditampilkan secara langsung di halaman web, tetapi memengaruhi cara halaman web ditampilkan dan diindeks oleh mesin pencari.
4. `<meta charset="UTF-8">`
Elemen ini mendefinisikan karakter encoding yang digunakan dalam halaman web. Dalam contoh di atas, menggunakan UTF-8 sebagai karakter encoding, yang mendukung berbagai karakter dan bahasa.
5. `<title>`
Elemen ini digunakan untuk memberikan judul halaman web. Judul ini akan ditampilkan di bilah judul peramban dan juga digunakan sebagai judul saat halaman web ditandai atau disimpan sebagai bookmark.

6. <body>

Elemen ini berisi seluruh konten yang akan ditampilkan di halaman web. Semua teks, gambar, video, formulir, dan elemen lainnya akan ditempatkan di dalam elemen <body>. Konten yang ditempatkan di sini akan ditampilkan di peramban saat pengguna mengakses halaman web.

1.3.3 Heading

Setelah memahami struktur dasar ini, kita dapat menambahkan elemen-elemen HTML lainnya di dalam elemen <body> untuk membentuk tampilan dan konten halaman web secara lengkap. Heading adalah elemen dalam bahasa markup seperti HTML yang digunakan untuk menandai tingkatan judul dalam sebuah halaman web. Elemen heading digunakan untuk memberikan hierarki pada konten halaman web, dengan <h1> sebagai judul utama dan tingkat heading yang lebih rendah seperti <h2>, <h3>, <h4>, <h5>, dan <h6> sebagai subjudul yang semakin mendalam (Jubilee Enterprise, 2018).

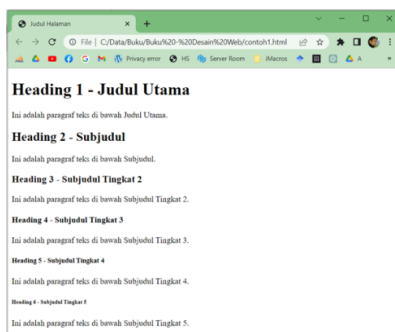
Heading biasanya digunakan untuk membuat teks berbeda ukuran dan tampilan, menyoroti bagian-bagian penting dari halaman web, dan membantu pengguna dan mesin pencari memahami struktur konten halaman tersebut. Heading juga berfungsi sebagai petunjuk bagi pembaca untuk memahami isi dan topik yang akan dibahas dalam bagian-bagian tertentu dari halaman web. Berikut adalah contoh penggunaan elemen heading (judul) dalam struktur dasar dokumen HTML:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Judul Halaman</title>
</head>
<body>
  <h1>Heading 1-Judul Utama</h1>
  <p>Ini adalah paragraf teks di bawah Judul Utama.</p>
```

```
<h2>Heading 2-Subjudul</h2>
<p>Ini adalah paragraf teks di bawah Subjudul.</p>
<h3>Heading 3-Subjudul Tingkat 2</h3>
<p>Ini adalah paragraf teks di bawah Subjudul Tingkat 2.</p>
<h4>Heading 4-Subjudul Tingkat 3</h4>
<p>Ini adalah paragraf teks di bawah Subjudul Tingkat 3.</p>
<h5>Heading 5-Subjudul Tingkat 4</h5>
<p>Ini adalah paragraf teks di bawah Subjudul Tingkat 4.</p>
<h6>Heading 6-Subjudul Tingkat 5</h6>
<p>Ini adalah paragraf teks di bawah Subjudul Tingkat 5.</p>
</body>
</html>
```

Penjelasan:

1. Elemen `<h1>` hingga `<h6>` digunakan untuk menandai tingkatan judul dalam halaman web. `<h1>` adalah judul utama, dan tingkat judul menurun dari `<h2>` hingga `<h6>`. Semakin tinggi nomor tingkatan heading, semakin rendah tingkatannya.
2. Elemen `<p>` digunakan untuk menandai paragraf teks di bawah setiap judul. Paragraf teks ini akan berada di bawah judul sesuai tingkat heading yang digunakan.



Gambar 1.8: Contoh Penggunaan Heading

1.3.4 Elemen Paragraph dan Line Break

Elemen `<p>` (Paragraph) dan `
` (Line Break) adalah elemen dalam bahasa markup HTML yang digunakan untuk mengatur tampilan dan struktur teks dalam halaman web (Abdulloh, 2022).

Elemen `<p>` (Paragraph):

Elemen `<p>` digunakan untuk menandai dan mengatur paragraf teks dalam halaman web. Setiap kali elemen `<p>` digunakan, teks yang berada di dalamnya akan dianggap sebagai satu paragraf terpisah. Elemen `<p>` secara otomatis menambahkan jarak kosong (enter) sebelum dan sesudah paragraf, sehingga membuat paragraf terlihat terpisah satu sama lain.

Contoh penggunaan elemen `<p>`:

```
<p>Ini adalah paragraf pertama.</p>
```

```
<p>Ini adalah paragraf kedua.</p>
```

Dalam contoh di atas, kita memiliki dua paragraf yang terpisah di dalam halaman web.

Elemen `
` (Line Break):

Elemen `
` digunakan untuk menyisipkan pemisah baris (line break) dalam halaman web. Saat elemen `
` digunakan, teks setelahnya akan muncul dalam baris berikutnya tanpa membentuk paragraf baru.

Contoh penggunaan elemen `
`:

```
<p>Ini adalah baris pertama.<br>Ini adalah baris kedua.</p>
```

Dalam contoh di atas, teks "Ini adalah baris kedua." akan muncul dalam baris berikutnya tanpa membentuk paragraf baru.

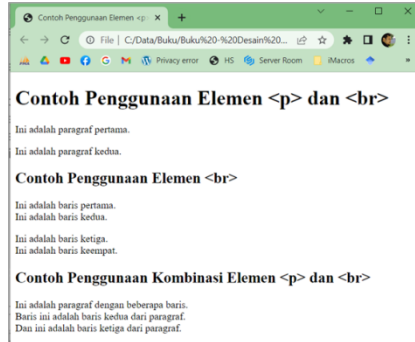
Perbedaan utama antara elemen `<p>` dan `
` adalah bahwa elemen `<p>` digunakan untuk membentuk paragraf terpisah, sedangkan elemen `
` digunakan untuk mengatur pemisah baris dalam satu paragraf. Penggunaan elemen `<p>` lebih cocok untuk mengelompokkan teks yang berkaitan menjadi paragraf terpisah, sedangkan elemen `
` cocok digunakan ketika kita ingin memisahkan baris dalam satu paragraf tanpa membentuk paragraf baru.

Berikut adalah contoh kode lengkap penggunaan elemen `<p>` (Paragraph) dan `
` (Line Break) dalam halaman web:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Contoh Penggunaan Elemen &lt;p&gt; dan
&lt;br&gt;</title>
</head>
<body>
  <h1>Contoh Penggunaan Elemen &lt;p&gt; dan &lt;br&gt;</h1>
  <p>Ini adalah paragraf pertama.</p>
  <p>Ini adalah paragraf kedua.</p>
  <h2>Contoh Penggunaan Elemen &lt;br&gt;</h2>
  <p>Ini adalah baris pertama.<br>Ini adalah baris kedua.</p>
  <p>Ini adalah baris ketiga.<br>Ini adalah baris keempat.</p>
  <h2>Contoh Penggunaan Kombinasi Elemen &lt;p&gt; dan
&lt;br&gt;</h2>
  <p>Ini adalah paragraf dengan beberapa baris.<br>
  Baris ini adalah baris kedua dari paragraf.<br>
  Dan ini adalah baris ketiga dari paragraf.</p>
</body>
</html>
```

Penjelasan:

Dalam contoh di atas, elemen `<p>` digunakan untuk membentuk paragraf terpisah. Setiap elemen `<p>` menandai satu paragraf yang diatur dengan jarak kosong di antara mereka.



Gambar 1.9: Contoh Penggunaan Paragraph dan Line Break

Elemen `
` digunakan untuk memisahkan baris dalam satu paragraf tanpa membentuk paragraf baru. Setiap elemen `
` menambahkan pemisah baris (line break) di lokasi tersebut, sehingga teks berikutnya akan muncul pada baris berikutnya.

1.3.5 Elemen Penformatan

Elemen pemformatan dalam HTML digunakan untuk memberikan tampilan khusus atau gaya tertentu pada teks atau elemen lain dalam halaman web. Berikut adalah beberapa elemen-formatting yang umum digunakan (Tutang, 2020):

1. ``-Bold:
Elemen `` digunakan untuk membuat teks menjadi tebal (bold).
2. `<i>`-Italic:
Elemen `<i>` digunakan untuk membuat teks menjadi miring (italic).
3. `<u>`-Underline:
Elemen `<u>` digunakan untuk memberikan garis bawah pada teks.
4. `<s>`-Strikethrough:
Elemen `<s>` digunakan untuk menampilkan teks dengan garis coret di tengahnya.
5. `<sup>`-Superscript:
Elemen `<sup>` digunakan untuk menampilkan teks sebagai superskrip (terangkat di atas garis).

6. `<sub>`-Subscript:

Elemen `<sub>` digunakan untuk menampilkan teks sebagai subskrip (turturunkan di bawah garis).

7. `<code>`-Kode Sumber:

Elemen `<code>` digunakan untuk menampilkan kode atau fragmen kode sumber komputer dalam bentuk teks monospace.

Berikut adalah contoh lengkap penggunaan elemen pemformatan dalam HTML tanpa menggunakan CSS:

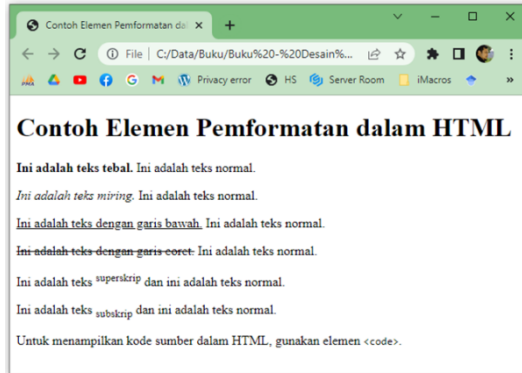
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Contoh Elemen Pemformatan dalam HTML</title>
</head>
<body>
  <h1>Contoh Elemen Pemformatan dalam HTML</h1>
  <p><b>Ini adalah teks tebal.</b> Ini adalah teks normal.</p>
  <p><i>Ini adalah teks miring.</i> Ini adalah teks normal.</p>
  <p><u>Ini adalah teks dengan garis bawah.</u> Ini adalah teks normal.</p>
  <p><s>Ini adalah teks dengan garis coret.</s> Ini adalah teks normal.</p>
  <p>Ini adalah teks <sup>superskrip</sup> dan ini adalah teks normal.</p>
  <p>Ini adalah teks <sub>subskrip</sub> dan ini adalah teks normal.</p>
  <p>Untuk menampilkan kode sumber dalam HTML, gunakan elemen <code>&lt;code&gt;</code>.</p>
</body>
```


</html>

Penjelasan:

Contoh di atas menunjukkan penggunaan elemen pemformatan seperti , <i>, <u>, <s>, <sup>, dan <sub>. Setiap elemen ini memberikan efek format tertentu pada teks sesuai dengan namanya.

Elemen <code> digunakan untuk menampilkan kode sumber dalam teks monospace.



Gambar 1.10: Contoh Penformatan dalam HTML

Bab 2

Daftar dan Tabel

2.1 Pendahuluan

Hypertext Markup Language (HTML) merupakan bahasa bertanda dasar dalam pembuatan dan pengaturan konten pada halaman web. Dalam HTML, terdapat beberapa elemen yang sangat penting, yang secara umum digunakan untuk menyajikan data secara terstruktur, yaitu List dan Tabel (Sharma, 2018). Dalam penulisan HTML diperlukan juga pendukung untuk memaksimalkan tampilan dari HTML agar lebih interaktif dan berfungsi maksimal menggunakan CSS (Cascading Style Sheet). CSS adalah spesifikasi W3C untuk penulisan style sheet dalam penggunaannya untuk menjelaskan sebuah dokumen yang di presentasikan menggunakan bahasa bertanda, lebih tepatnya, untuk menata dokumen Web (Queirós, 2018).

2.2 Tabel dalam HTML

2.2.1 Tabel secara umum

Tabel secara fungsional digunakan untuk mempresentasikan data dalam bentuk kolom serta baris. Setiap sel dalam tabel berisi data atau konten

tertentu. Struktur dasar tabel terdiri dari beberapa elemen:(Foundation Website Creation with CSS, XHTML, and JavaScript, 2008)

1. `<table>`: Elemen ini menandai awal dan akhir dari tabel.
2. `<thead>`: Elemen ini berisi baris judul atau header tabel.
3. `<tr>`: Elemen ini menandai baris dalam tabel.
4. `<th>`: Elemen ini digunakan untuk membuat sel judul (header) dalam tabel.
5. `<tbody>`: Elemen ini berisi isi dari tabel.
6. `<td>`: Elemen ini digunakan untuk membuat sel data dalam tabel.

Berikut ini adalah contoh kodenya secara umum.

...

```
<table>
  <thead>
    <tr>
      <th>First Column</th>
      <th>Second Column</th>
      <th>Third Column</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>First Row, First Column</td>
      <td>First Row, Second Column</td>
      <td>First Row, Third Column</td>
    </tr>
    <tr>
      <td>Second Row, First Column</td>
      <td>Second Row, Second Column</td>
      <td> Second Row, Third Column </td>
```

```
</tr>
</tbody>
</table>
...

```

First Column	Second Column	Third Column
First Row, First Column	First Row, Second Column	First Row, Third Column
Second Row, First Column	Second Row, Second Column	Second Row, Third Column

Gambar 2.1: Hasil Kode HTML Standard Untuk Tabel Secara Umu
(w3schools, 2023)

2.2.2 Tabel Border

Tampilan umum tabel pada pembahasan sebelumnya tidak menyertakan garis batas atau border. Dalam HTML seperti pembahasan pada awal bab ini, memerlukan dukungan dari CSS untuk memaksimalkan tampilan (Adams et al., 2007). Untuk menambahkan border diperlukan style dari CSS dengan pembahasan sebagai berikut.

Kode CSS untuk menambahkan garis pembatas:

```
...
<style>
  table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
  }
</style>
...

```


Gambar 2.2: Gambaran hasil Style CSS untuk menambahkan garis batas
(w3schools, 2023)

Dalam kode di atas untuk ukuran ketebalan border dapat disesuaikan dalam ukuran pixel, dan untuk warnanya juga dapat disesuaikan dengan kebutuhan. Sementara untuk `border-collapse` adalah salah satu properti dalam CSS yang digunakan untuk mengontrol bagaimana tepi (border) dari sel-sel tabel berinteraksi satu sama lain. Properti ini hanya berlaku untuk elemen-elemen tabel seperti `<table>`, `<thead>`, `<tbody>`, `<tfoot>`, `<tr>`, dan `<td>` atau `<th>`.

Ada dua nilai yang dapat digunakan untuk properti `border-collapse`:

1. `collapse`

Ini adalah nilai default. Ketika digunakan, tepi dari sel-sel tabel akan digabungkan (collapsed) sehingga tidak ada jarak antara tepi dari sel-sel yang berdekatan. Ini menghasilkan tampilan yang lebih rapi dan konsisten. Jika terdapat dua sel yang berdekatan, border-nya hanya akan ditampilkan sekali di antara mereka.

2. `separate`

Dengan nilai ini, setiap sel dalam tabel memiliki tepi (border) yang terpisah dan tidak digabungkan dengan sel-sel tetangganya. Hal ini dapat menghasilkan efek yang lebih jelas untuk setiap sel dan tepi yang lebih terlihat di antara sel-sel yang berdekatan.

Contoh penggunaan:

```
```css
table {
 border-collapse: collapse;/* atau separate */
}
td, th {
 border: 1px solid black;
}
```
```

Dalam contoh di atas, jika `border-collapse` diatur sebagai `collapse`, maka tepi dari sel-sel akan digabungkan dan hanya satu garis akan muncul di antara sel-sel yang berdekatan. Jika diatur sebagai `separate`, maka setiap sel akan

memiliki tepi terpisah dan garis akan terlihat lebih jelas di antara sel-sel yang berdekatan.

Pilihan antara `collapse` dan `separate` tergantung pada desain tampilan yang diinginkan. Jika anda ingin tampilan yang lebih bersih dan teratur, `collapse` bisa menjadi pilihan yang baik. Namun, jika anda ingin lebih mengutamakan penekanan pada setiap sel dan tepinya, maka `separate` mungkin lebih cocok.

Berikut ini adalah beberapa atribut yang terkait dengan elemen tabel dalam HTML:(Bowers, Synodinos and Sumner, 2011)

1. Border

Atribut ini memiliki fungsi untuk mengatur ketebalan border (garis batas pada tepi) pada tabel dan sel-selnya. Nilai yang diberikan adalah bilangan bulat yang menunjukkan ketebalan border dalam piksel. Dengan contoh kode sebagai berikut:

```
...  
<table border="1">  
  <!--Isi tabel-->  
</table>  
...
```

2. Width

Atribut ini memiliki fungsi untuk mengatur seberapa lebar tabel. Diukur dalam nilai piksel (px) atau persentase (%) dari lebar kontainer induk.

Dengan contoh kode sebagai berikut:

```
...  
<table width="80%">  
  <!--Isi tabel-->  
</table>  
...
```

3. Cellpadding

Atribut ini digunakan untuk menentukan jarak antara isi sel dengan batas sel (padding). Nilai yang diberikan adalah bilangan bulat yang menunjukkan jarak dalam piksel.

Dengan contoh kode sebagai berikut:

```
...
```

```
<table cellpadding="10">
  <!--Isi tabel-->
</table>
...

```

4. Cellspacing

Atribut ini digunakan untuk menentukan jarak antara sel-sel dalam tabel (spacing). Nilai yang diberikan adalah bilangan bulat yang menunjukkan jarak dalam piksel.

Dengan contoh kode sebagai berikut:

```
...
<table cellspacing="5">
  <!--Isi tabel-->
</table>
...

```

5. Align

Atribut ini digunakan untuk mengatur alignment (penyusunan) tabel dalam halaman. Nilai yang umum adalah left, center, dan right.

Dengan contoh kode sebagai berikut:

```
...
<table align="center">
  <!--Isi tabel-->
</table>
...

```

6. Summary

Atribut ini digunakan untuk memberikan deskripsi singkat tentang isi tabel. Ini dapat memberikan informasi tambahan kepada pengguna saat mengakses tabel.

Dengan contoh kode sebagai berikut:

```
...
<table summary="kesimpulan dari data yang ada">
  <!--Isi tabel-->
</table>
...

```

7. Bordercolor

Atribut ini digunakan untuk mengatur warna border tabel. Nilai yang diberikan bisa berupa nama warna atau kode warna dalam heksadesimal.

Dengan contoh kode sebagai berikut:

```
...  
<table bordercolor="#ff0000">  
  <!--Isi tabel-->  
</table>  
...
```

8. Bgcolor

Atribut ini berfungsi untuk menentukan latar warna pada tabel bagian belakang. Nilai yang diberikan bisa berupa nama warna atau kode warna dalam heksadesimal.

Dengan contoh kode sebagai berikut:

```
...  
<table bgcolor="#f2f2f2">  
  <!--Isi tabel-->  
</table>  
...
```

9. Frame

Atribut ini digunakan untuk mengatur tampilan garis tepi (border) tabel. Nilai yang umum adalah void, above, below, hside, lhs, rhs, vside, dan box.

Dengan contoh kode sebagai berikut:

```
...  
<table frame="hside">  
  <!--Isi tabel-->  
</table>  
...
```

10. Rules

Atribut ini digunakan untuk mengatur garis pemisah antara sel-sel dalam tabel. Nilai yang umum adalah none, groups, rows, dan cols.

Dengan contoh kode sebagai berikut:

```
...
```



```
<table rules="rows">
  <!--Isi tabel-->
</table>
...

```

11. Caption

Elemen `<caption>` dapat digunakan di dalam elemen `<table>` untuk menambahkan judul atau keterangan pada tabel.

Dengan contoh kode sebagai berikut:

```
...
<table>
  <caption>Data Mahasiswa</caption>
  <!--Isi tabel-->
</table>
...

```

Atribut-atribut ini memberikan fleksibilitas dalam mengatur tampilan dan perilaku tabel dalam halaman web Anda. Ingatlah bahwa beberapa atribut mungkin sudah tidak disarankan atau didukung dalam praktik pengembangan web yang lebih modern.

2.3 List dalam HTML

List dalam HTML digunakan untuk menyajikan item-item dalam bentuk urutan tertentu. Terdapat tiga jenis list dalam HTML: Unordered List (Daftar tak berurutan), Ordered List (Daftar berurutan), dan Description List (Daftar deskripsi)(Brooks, 2011). Berikut adalah pembahasannya.

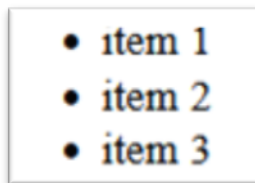
2.3.1 HTML: Unordered List

Daftar atau List yang tidak berurut menggunakan awalan tag `` dan daftar setiap item menggunakan awalan tag ``

Daftar setiap item atau list menggunakan poin (•) secara otomatis jika tidak diganti (Melorose, Perroy and Careas, 2015).

Berikut adalah contoh kodenya dan hasilnya bisa dilihat pada gambar 2.3

```
...  
<html>  
<body>  
<ul>  
  <li>item 1</li>  
  <li>item 2</li>  
  <li>item 3</li>  
</ul>  
</body>  
</html>  
...
```



Gambar 2.3: Gambaran hasil kode HTML untuk unordered list(w3schools, 2023)

2.3.2 HTML: Ordered List

Daftar yang diurutkan menggunakan awalan tag `` list setiap item diawali menggunakan tag ``

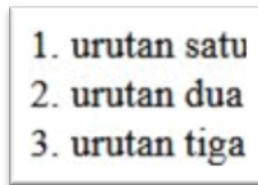
List item akan menggunakan angka secara default Berikut adalah contoh kodenya dan hasilnya bisa dilihat pada gambar 2.4

```
...  
<html>  
<body>  
<ol type="1">
```

```

<li> urutan satu </li>
<li> urutan dua </li>
<li> urutan tiga </li>
</ol> </body>
</html>
...

```



Gambar 2.4: Gambaran hasil kode HTML untuk ordered list (w3schools, 2023)

Jenis atribut dari tag ``, mendefinisikan jenis penanda item daftar yang berurutan, berikut adalah daftar atributnya, yang bisa dilihat pada tabel 2.1 dibawah ini.

Tabel 2.1: Atribut Tipe urutan dalam list (w3schools, 2023)

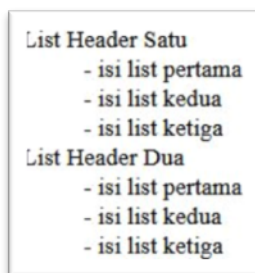
Type	Description
type="1"	Item list untuk menampilkan angka (default)
type="A"	Item list untuk menampilkan nomor huruf besar
type="a"	Item list untuk menampilkan nomor huruf kecil
type="I"	Item list untuk menampilkan angka romawi dengan huruf besar
type="i"	Item list untuk menampilkan angka romawi dengan huruf kecil

2.3.3 HTML: Description List

Daftar deskripsi adalah daftar istilah dengan deskripsi masing-masing istilah. Tag `<dl>` mendefinisikan daftar deskripsi, tag `<dt>` mendefinisikan istilah (nama) dan tag `<dd>` menjelaskan setiap istilah (Meyer, 2004).

Berikut adalah contoh kodenya dan hasilnya bisa dilihat pada gambar 2.5

```
...  
<html>  
<body>  
<dl>  
  <dt>List Header Satu</dt>  
  <dd>-isi list pertama</dd>  
  <dd>-isi list kedua</dd>  
  <dd>-isi list ketiga</dd>  
  <dt>List Header Dua</dt>  
  <dd>-isi list pertama</dd>  
  <dd>-isi list kedua</dd>  
  <dd>-isi list ketiga</dd>  
</dl>  
</body>  
</html>  
...
```



Gambar 2.5: Gambaran hasil kode HTML untuk description list (w3schools, 2023)

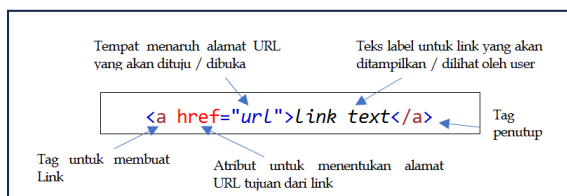
Bab 3

Hyperlink, Gambar dan Media

3.1 Hyperlink

Di dalam pembuatan web, salah satu bagian yang tidak bisa dihindari adalah pembuatan hyperlink. Dengan adanya hyperlink, user dapat langsung menuju halaman lain dengan menekan link yang ada. Link dapat di pilih saat kursor mouse berada diatas link. Kursor akan berubah menjadi sebuah tangan. Link tidak selamanya berbentuk teks, tetapi link juga dapat berbentuk sebuah gambar atau elemen lain.

Link di HTML dapat dibuat dengan menggunakan tag “<a>”. Tag tersebut kemudian digabungkan dengan atribut “href”, yang digunakan untuk menentukan alamat URL yang akan dituju. Berikut adalah penjelasan penulisan link:

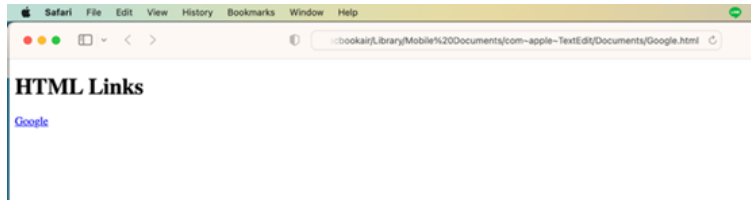


Gambar 3.1: Syntax Link (Ahmad Muhardian, n.d.)

Contoh penggunaan Syntax:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>HTML Links</h1>
    <p><a href="https://www.Google.com/">Google</a></p>
  </body>
</html>
```

Contoh diatas akan membawa user ke tampilan output seperti dibawah ini:



Gambar 3.2: Tampilan Website (W3School, n.d.)

Saat link di-klik, maka user akan dibawa ke halaman Google, sesuai dengan perintah yang diberikan di dalam perintah, seperti gambar dibawah ini:



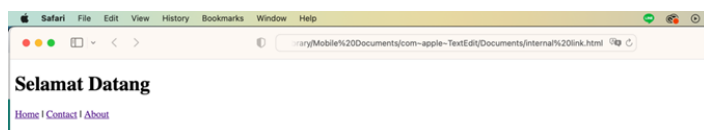
Gambar 3.3: Website Google (W3School, n.d.)

3.2 Jenis Link

Link pada HTML terbagi atas 2 kelompok, yaitu:

1. Internal Link

Link internal merupakan link yang akan langsung menuju ke halaman web yang di-klik. Contohnya:



Gambar 3.4: Tampilan Utama Internal Link

Jika di klik tanda “Contact“, maka halaman yang tampak akan berubah sesuai dengan perintah yang sudah diberikan, sebagai berikut:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Internal Link</title>
  </head>
  <body>
    <h1>Selamat Datang</h1>
    <p>
      <a href="index.html" >Home</a> |
      <a href="contact.html" >Contact</a> |
      <a href="about.html" >About</a>
    </p>
  </body>
</html>
```

Berikut adalah tampilan kontak pada website:



Gambar 3.5: Tampilan Halaman Contact

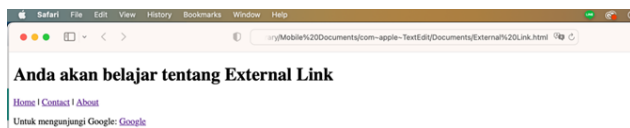
NB: Pastikan file untuk internal ini tersimpan dalam folder yang sama.

2. External Link

Sedangkan untuk External Link, untuk membuatnya sama dengan internal link. Beda antara external link dan internal link adalah external link langsung memberikan alamat URL di dalam perintah yang diberikan. Contoh perintah untuk external link:

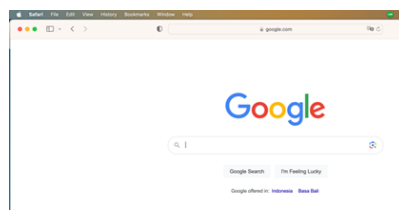
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Internal Link</title>
  </head>
  <body>
    <h1>Anda akan belajar tentang External Link</h1>
    <p>
      <a href="index.html" >Home</a> |
      <a href="contact.html" >Contact</a> |
      <a href="about.html" >About</a>
    </p>
    <p>
      Untuk mengunjungi
      <a href="https://www.google.com">Google</a>
    </p>
  </body>
</html>
```

Berikut adalah tampilan pada external link yang sesuai dengan kode yang telah dijalankan:



Gambar 3.6: External Link

Jika yang diklik pada kata Google yang dibawah, maka tampilan akan berubah menjadi:



Gambar 3.7: Tampilan Google pada External Link

3.3 Atribut Link

Untuk link yang diberikan, juga dapat langsung dibuka di browser yang sedang dipakai. Cara penggunaannya adalah dengan menambahkan perintah “Target”

“Target” merupakan perintah untuk membuka dokumen atau halaman tertentu yang telah dituliskan di dalam perintah tersebut.

Atribut “Target” terdiri dari 4 bagian (Suryana, n.d.), yaitu:

1. `_self`
Perintah ini merupakan perintah untuk membuka dokumen di window yang sama, yang sedang aktif digunakan.
2. `_blank`
Perintah ini merupakan perintah untuk membuka dokumen di tab atau di window yang baru.
3. `_parent`
Perintah ini merupakan perintah untuk membuka dokumen di bingkai Parent/induk.
4. `_top`
Perintah ini merupakan perintah untuk membuka seluruh dokumen di seluruh isi jendela.

Contoh penggunaan “`_blank`”:

```
<!DOCTYPE html>
<html>
<body>

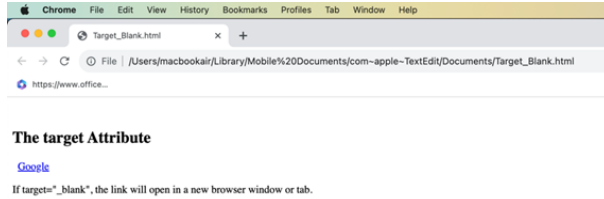
<h2>The target Attribute</h2>

<a href="https://www.google.com/" target="_blank">Google</a>

<p>If target="_blank", the link will open in a new browser window or tab.</p>

</body>
</html>
```

Tampilan website akan nampak seperti gambar dibawah ini:



Gambar 3.8: Tampilan Website menggunakan Attribute Target = _blank

Jika di-klik kata Google, maka website Google akan ditampilkan di halaman tab baru.



Gambar 3.9: Tampilan Website Google

3.4 Link dengan Gambar

Tidak hanya menggunakan tulisan untuk dijadikan menjadi link, tetapi gambar juga bisa dijadikan sebuah link. Contohnya:

```
<!DOCTYPE html>
<html>
<body>

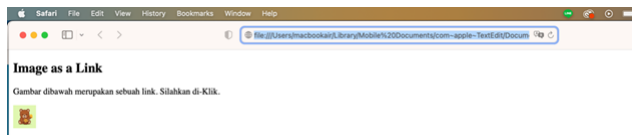
<h2>Image as a Link</h2>

<p>Gambar dibawah merupakan sebuah link. Silahkan di-Klik.</p>

<a href="https://www.google.com/search?q=gambar+beruang&sa_esv=556502029&rlz=1C5CHFA_enID1047ID1049&sxsrif=AB5stBicXG00xtIL0yjKvFLI-19ylzobw:1691923570997&tbm=isch&source=iu&ictx=1&vet=1&fir=YgF2b3pSILBPjM%252C_44HmSV8BjuyUM%252C_%253BPpgi2xxwoawUM%252CmtY83yc5wRz8JM%252C_%253B_frEIKz.ft3ul.M%252C3v2_h4AVfgiexM%252C_%253B3_bj1YWPz99dLM%252C7yEr10iG30TWUM%252C_6usg=AI4_-kRuyFj1t8TA2ESYX8k01B5rneTwz06sa=X&sqi=2&ved=2ahUKEwiyuYe7utmAAxWI6igHRRSbtKQ_h168AgnEAE#imgsrc=Ppgi2xxwoawUM"></a>

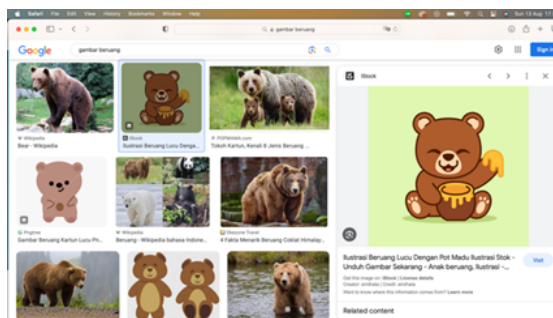
</body>
</html>
```

Hasil yang akan diperoleh saat menjalankan perintah diatas adalah:



Gambar 3.10: Link menggunakan Gambar (Murhadian, 2020)

Setelah meng-klik gambar beruang yang ada, maka tampilan akan berubah menjadi seperti dibawah ini:



Gambar 3.11: Hasil Tampilan Link menggunakan Gambar

3.5 Link Ke Alamat Email

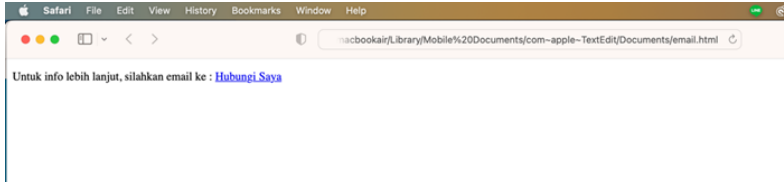
Di HTML, juga dapat membuat link untuk ke alamat email yang diinginkan. Untuk melakukan ini, dapat digunakan perintah “mailto“. Perintah ini akan membawa user ke alamat email yang disediakan di dalam coding. Contoh:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Link ke Amail</title>
  </head>
  <body>
    <p>
      Untuk info lebih lanjut, silahkan email ke :
      <a href="mailto:tes@gmail.com" target="_blank"
        title="kirin email ke Admin">Hubungi Saya</a>
    </p>
  </body>
</html>

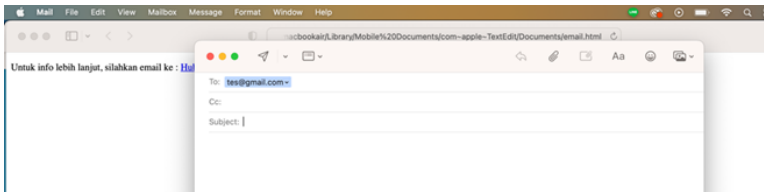
```

Hasil dari perintah diatas adalah:



Gambar 3.12: Tampilan Link ke Email

Jika kata “hubungi saya” di klik, maka tampilan yang akan ditampilkan adalah:

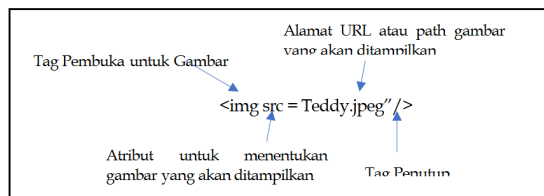


Gambar 3.13: Hasil Tampilan Link ke Email

3.6 Menampilkan Gambar

Di dalam HTML, selalu mempunyai gambar yang digunakan untuk membuat website menjadi lebih menarik. Perintah yang digunakan untuk menampilkan gambar ada disebutkan juga dibagian atas, yaitu saat membuat link menggunakan gambar. Perintah tersebut adalah “”. Perintah “” biasanya harus diikuti oleh perintah “src” yang di mana merupakan singkatan dari source yang artinya sumber. Jika perintah “” tidak diikuti oleh kata “src”, maka gambar yang diinginkan tidak dapat ditampilkan.

Contoh penggunaan perintah menampilkan gambar di website:



Gambar 3.14: Perintah untuk Menampilkan Gambar (Murhadian, 2020)

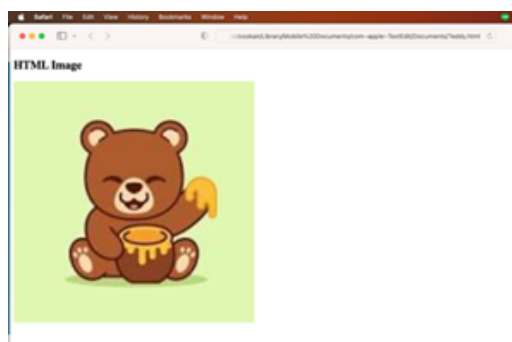
Contoh pemanggilan gambar:

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Image</h2>


</body>
</html>
```

Hasil yang diperoleh dari perintah diatas adalah:



Gambar 3.15: Teddy (W3Schools, n.d.)

Note: Gambar beruang yang ditampilkan sudah disimpan di tempat yang sama dengan file pemanggilan.

Tipe gambar yang dapat dipakai berbagai browser seperti Chrome, Edge, Firefox, Safari dan Opera adalah sebagai berikut:

Tabel 3.1: Format Gambar (W3Schools, n.d.)

Singkatan	Kepanjangan	File Extension
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico,.cur
JPEG	Joint Photographic Expert Group image	.jpg,.jpeg,.jfif,.jpeg,.jpp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg

3.7 Ukuran Gambar

Ukuran gambar yang dimasukkan di dalam HTML dapat ditentukan dengan cara menambahkan perintah:

1. Height

Perintah Height digunakan untuk menentukan tinggi gambar yang akan ditampilkan.

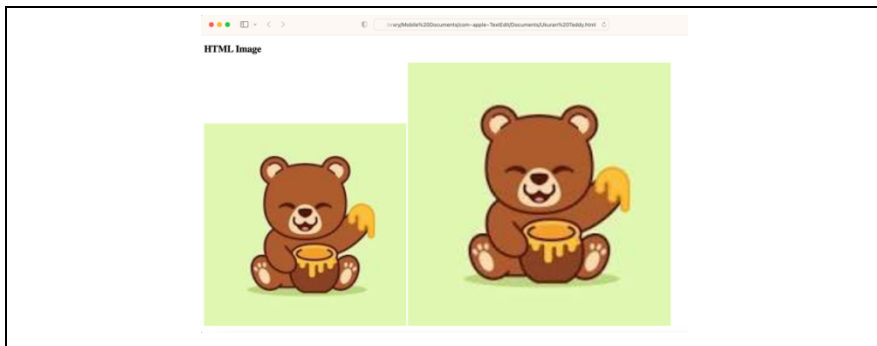
2. Width

Perintah Width digunakan untuk menentukan lebar gambar yang akan ditampilkan.

Kedua perintah ini ada digunakan pada perintah sebelumnya saat menampilkan gambar, yaitu gambar 3.15. jika perintah tersebut diubah, akan nampak perbedaan perintah sebelumnya dengan yang sekarang.

Tabel 3.2: Perbandingan Ukuran Gambar

<pre><!DOCTYPE html> <html> <body> <h2>HTML Image</h2> </body> </html></pre>	<pre><!DOCTYPE html> <html> <body> <h2>HTML Image</h2> </body> </html></pre>
--	--



Sesuai yang nampak pada tabel 3.2 diatas, adanya perbedaan ukuran pada gambar yang ditampilkan, di mana gambar pertama berukuran 500 untuk tinggi dan lebar. Sedangkan gambar kedua nampak lebih besar, yaitu berukuran 650 untuk tinggi dan lebar gambar.

Jika ukuran yang digunakan untuk ditampilkan di dalam browser tidak mengalami perubahan pada ukuran gambar, maka dapat digunakan “Style“. Contoh Penggunaan Style:

```
<!DOCTYPE html>
<html>
<head>
<style>
  /* This style sets the width of all images to 100%: */
  img {
    width: 100%;
  }
</style>
</head>
<body>
<h2>Width/Height Attributes or Style?</h2>
```

`<p>`The first image uses the width attribute (set to 128 pixels), but the style in the head section overrides it, and sets the width to 100%.`</p>`


```

```

<p>The second image uses the style attribute to set the width to 128 pixels, this will not be overridden by the style in the head section:</p>

```

```

```
</body>
```

```
</html>
```

Hasil dari perintah diatas adalah:



Gambar 3.16: Style (W3Schools, n.d.)

3.8 Audio

Di dalam HTML, selain menampilkan gambar, juga dapat menambahkan audio. Sebelumnya audio dapat ditambahkan dengan menambahkan atau menggabungkan program lain seperti Macromedia Flash. Sekarang audio bisa ditambahkan langsung dari perintah di dalam HTML tanpa harus melalui program lain. Berikut adalah perintah untuk membuat audio.



Gambar 3.17: Perintah Audio (Ahmad Muhardian, n.d.)

3.9 Format Audio

Berikut adalah beberapa format audio yang dapat diputar di HTML:

Tabel 3.3: Format Audio (W3schools, n.d.)

Format	Container	MIME type
PCM	WAV	audio/wav
MP3	MP3	audio/mpeg
AAC	MP4	audio/mp4
AAC	AAC	audio/aac
AAC	AAC	audio/aacp
Vorbis	Ogg	audio/ogg
Vorbis	WebM	audio/webm
Opus	Ogg	audio/ogg
Opus	WebM	audio/webm
FLAC	FLAC	audio/flac
FLAC	Ogg	audio/ogg

Gunakanlah format seperti MP3 dan MP4, karena format tersebut relatif kecil.

3.10 Atribut Audio

Atribut audio yang banyak digunakan adalah:

1. Controls
2. Autoplay
3. Loop
4. Muted

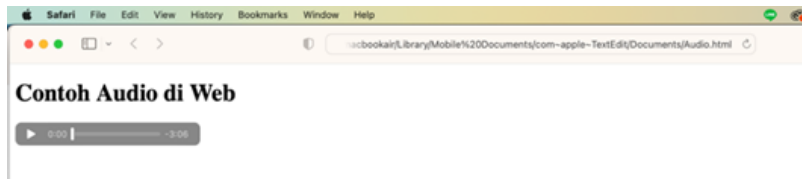
3.10.1 Controls

Atribut ini berguna untuk mengaktifkan tombol yang banyak digunakan di audio. Seperti tombol play, pause, stop, scroll, dan volume.

Berikut adalah contoh kode untuk audio.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Audio di HTML</title>
</head>
<body>
  <h1>Contoh Audio di Web</h1>
  <audio controls>
    <source src="Weird Genius - Lathi (ft. Sara Fajira).mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
  </audio>
</body>
</html>
```

Hasil dari perintah diatas adalah:



Gambar 3.18: Audio

Note: lagu yang digunakan berada di tempat yang sama dengan file pemanggil

3.10.2 Autoplay

Atribut ini berguna untuk melakukan pemutaran audio secara otomatis.

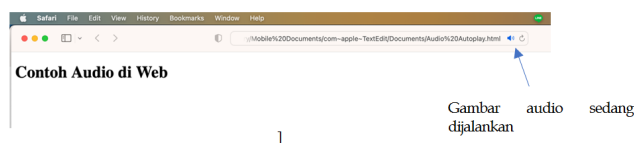
Contoh perintah untuk menjalankan perintah Autoplay:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Audio di HTML</title>
</head>
<body>
  <h1>Contoh Audio di Web</h1>

  <audio autoplay>
    <source src="Weird Genius - Lathi (ft. Sara Fajira).mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
  </audio>

</body>
</html>
```

Hasil dari perintah diatas adalah:



Gambar 3.19: Autoplay

Dapat dilihat dari gambar diatas, bahwa perintah begitu dijalankan audio lsg diputarkan tanpa adanya player.

3.10.3 Loop

Atribut ini digunakan untuk memutar audio berulang – ulang. Berikut adalah contoh coding:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Audio di HTML</title>
</head>
<body>
  <h1>Contoh Audio di Web</h1>

  <audio controls and audio loop>
    <source src="Weird Genius - Lathi (ft. Sara Fajira).mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
  </audio>

</body>
</html>
```

3.10.4 Muted

Atribut ini digunakan untuk me-muted-kann audio. Berikut adalah contoh perintah yang dijalankan:

3.11 Video di HTML

Selain menambahkan audio, di HTML juga dapat menambahkan video. Berikut adalah contoh perintah yang digunakan:

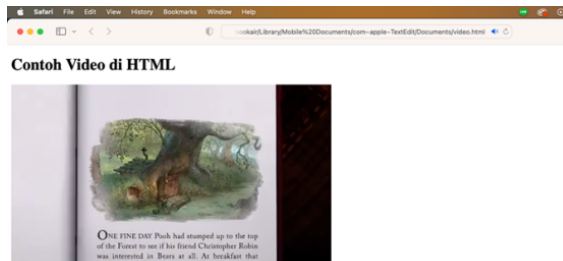


Gambar 3.20: Perintah Penambahan Video (Ahmad Muhandian, n.d.)

Contoh perintah untuk menambahkan video di HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Video di HTML</title>
</head>
<body>
  <h1>Contoh Video di HTML</h1>
  <video controls>
    <source src="" type="video/webm" />
    Browsermu tidak mendukung tag ini, upgrade donk!
  </video>
</body>
</html>
```

Hasil dari perintah diatas adalah:



Gambar 3.21: Winnie The Pooh

Video dari perintah diatas berada di folder yang sama dengan file yang akan dijalankan. Jika berbeda folder, maka perlu dituliskan path dari tempat penyimpanan video yang akan dipanggil.

Contoh penggalan perintah yang menggunakan pemanggilan video di tempat yang berbeda:

```
<video>
```

```
<source src="video/nama-video.mp4">
```


Bab 4

Formulir dan Elemen Input

4.1 Menggunakan Elemen Input Dasar

Desain web menggunakan elemen input dasar adalah tentang cara memanfaatkan elemen HTML input untuk mengumpulkan informasi dari pengguna melalui formulir di halaman web (Winarno and Zaki, 2015). Elemen input ini memberikan berbagai cara bagi pengguna untuk memasukkan data, seperti teks, angka, pilihan, dan lainnya. Berikut adalah beberapa contoh elemen input dasar yang umum digunakan dalam desain web:

4.1.1 Input Teks (Text Input)

Elemen input teks memungkinkan pengguna untuk memasukkan teks dengan menggunakan keyboard. Contoh penerapannya adalah saat pengguna diminta untuk mengisi nama, alamat, atau komentar pada formulir. Elemen ini didefinisikan dengan tag: `<input type="text">`.

Contoh kode HTML:

Nama:

```
<label for="nama">Nama: </label>
```

```
<input type="text" id="nama" name="nama">
```


4.1.2 Input Kata Sandi (Password Input)

Input kata sandi adalah varian dari elemen input teks yang mengaburkan teks yang dimasukkan pengguna, sehingga teks yang dimasukkan tidak terlihat. Ini berguna untuk meminta pengguna memasukkan kata sandi dengan aman. Elemen ini didefinisikan dengan tag: `<input type="password">`.

Contoh kode HTML:

Kata Sandi:

```
<label for="katasandi">Kata Sandi:</label>
<input type="password" id="katasandi" name="katasandi">
```

4.1.3 Input Angka (Number Input)

Input angka memungkinkan pengguna untuk memasukkan angka atau nilai numerik. Dalam beberapa browser, elemen ini juga menyediakan tombol peningkatan dan penurunan yang memudahkan pengguna untuk memilih nilai. Elemen ini didefinisikan dengan tag: `<input type="number">`.

Contoh kode HTML:

Usia:

```
<label for="usia">Usia:</label>
<input type="number" id="usia" name="usia">
```

4.1.4 Input Pilihan (Radio Button Input)

Input radio memungkinkan pengguna untuk memilih satu opsi dari beberapa pilihan. Jika ada beberapa pilihan, hanya satu pilihan yang dapat dipilih oleh pengguna. Elemen ini didefinisikan dengan tag `<input type="radio">` dengan menggunakan atribut `'name'` yang sama pada opsi yang berbeda untuk mengelompokkan pilihannya.

Contoh kode HTML:

Jenis Kelamin: Pria Wanita

```
<label>Jenis Kelamin:</label>
<input type="radio" id="1" name="jeniskelamin" value="Pria">
```

```
<label for="laki"> Pria </label>  
<input type="radio" id="p" name="jeniskelamin" value="Wanita">  
<label for="perempuan"> Wanita </label>
```

4.1.5 Input Kotak Centang (Checkbox Input)

Input checkbox memungkinkan pengguna untuk memilih satu atau beberapa pilihan dari beberapa opsi yang tersedia. Elemen ini didefinisikan dengan tag `<input type="checkbox">`.


Contoh kode HTML:

```
Hobi (bisa pilih lebih dari satu):  Membaca  Olahraga  
<label>Hobi (bisa pilih lebih dari satu):</label>  
<input type="checkbox" id="hobi1" name="hobi" value="Membaca">  
<label for="hobi1">Membaca</label>  
<input type="checkbox" id="hobi2" name="hobi" value="Olahraga">  
<label for="hobi2">Olahraga</label>
```

4.1.6 Input Tanggal (Date Input)

Input tanggal memungkinkan pengguna untuk memilih tanggal dari kalender yang ditampilkan. Elemen ini didefinisikan dengan tag `<input type="date">`.

Contoh kode HTML:

```
Tanggal Lahir:    
<label for="tgl_lahir">Tanggal Lahir:</label>  
<input type="date" id="tgl_lahir" name="tgl_lahir">
```

4.2 Membuat Formulir Kontak

Membuat formulir kontak adalah salah satu contoh penerapan penggunaan elemen input dasar dalam desain web. Formulir kontak digunakan untuk memungkinkan pengguna mengirimkan pesan atau informasi kepada pemilik

situs atau pengelola situs. Contoh Formulir kontak dapat dilihat pada Gambar 4.1.

Formulir Kontak

Nama:

Email:

Pesan:

Gambar 4.1: Contoh Formulir Kontak

Berikut adalah kode HTML sederhana untuk membuat formulir kontak pada:

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulir Kontak</title>
</head>
<body>
  <h1>Formulir Kontak</h1>
  <form action="/proses_formulir_kontak" method="post">
    <label for="nama">Nama:</label>
    <input type="text" id="nama" name="nama" required><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"
required><br><br>
    <label for="pesan">Pesan:</label>
    <textarea id="pesan" name="pesan"
required></textarea><br><br>
```

```
<input type="submit" value="Kirim">
</form>
</body>
</html>
```

Penjelasan:

1. `<form>`
Tag ini digunakan untuk mengelompokkan elemen-elemen formulir. Atribut `action` menentukan URL atau alamat tujuan di mana data formulir akan dikirimkan. Atribut `method` menentukan metode pengiriman data, dalam contoh ini, kita menggunakan metode POST.
2. `<label>`
Tag ini digunakan untuk memberikan label pada elemen input agar pengguna tahu apa yang harus diisi dalam elemen input tersebut. Atribut `for` pada label harus mencocokkan nilai atribut `id` dari elemen input yang ingin dilabeli.
3. `<input type="text">`
Elemen ini adalah input teks yang memungkinkan pengguna memasukkan teks.
4. `<input type="email">`
Elemen ini adalah input khusus untuk memvalidasi alamat email yang dimasukkan pengguna.
5. `<textarea>`
Elemen ini adalah input teks area yang memungkinkan pengguna memasukkan teks dalam jumlah yang lebih banyak daripada elemen input teks biasa.
6. `<input type="submit">`
Elemen ini adalah tombol "Kirim" yang akan mengirimkan data formulir ketika pengguna mengkliknya.

Catatan:

Atribut `required` pada elemen input dan `textarea` memastikan bahwa pengguna harus mengisi kolom tersebut sebelum mengirimkan formulir. Pembahasan

mengenai validasi input dengan atribut HTML5 akan dibahas lebih lengkap pada Sub Bab 4.2.1. Selain itu, Anda perlu mengatur action pada elemen form untuk menentukan skrip atau layanan yang akan memproses data formulir setelah pengguna mengirimkannya. Contoh di atas menggunakan "/proses_formulir_kontak" sebagai contoh alamat yang harus diganti dengan alamat yang sesuai dengan skrip atau layanan server Anda.

4.2.1 Validasi Input dengan HTML5 Atribut

HTML5 memperkenalkan beberapa atribut bawaan yang memungkinkan validasi input data di sisi klien (di perangkat pengguna) tanpa perlu menggunakan skrip tambahan. Atribut-atribut ini memanfaatkan fitur HTML5 dan JavaScript untuk memberikan umpan balik kepada pengguna tentang kesalahan dalam pengisian formulir sebelum data dikirimkan ke server.

Berikut adalah beberapa contoh atribut validasi input HTML5:

1. Atribut `required`

Atribut `required` digunakan pada elemen input atau elemen `textarea` untuk memastikan bahwa pengguna harus mengisi bidang tersebut sebelum mengirimkan formulir.

Contoh:

```
<label for="nama">Nama:</label>  
<input type="text" id="nama" name="nama" required>
```

2. Atribut `type="email"`

Atribut `type="email"` pada elemen input memastikan bahwa nilai yang dimasukkan pengguna harus berupa alamat email yang valid.

Contoh:

```
<label for="email">Email:</label>  
<input type="email" id="email" name="email" required>
```

3. Atribut `type="url"`

Atribut `type="url"` pada elemen input memastikan bahwa nilai yang dimasukkan pengguna harus berupa URL yang valid.

Contoh:

```
<label for="situsweb">Situs Web:</label>  
<input type="url" id="situsweb" name="situsweb" required>
```

4. Atribut type="number" dengan min dan max

Atribut type="number" pada elemen input memungkinkan input hanya menerima nilai numerik. Kita juga dapat menetapkan nilai minimum dan maksimum yang diperbolehkan.

Contoh:

```
<label for="usia">Usia:</label>
<input type="number" id="usia" name="usia" min="18"
max="80" required>
```

5. Atribut pattern

Atribut pattern memungkinkan kita untuk menggunakan ekspresi reguler untuk memvalidasi nilai yang dimasukkan pengguna. Ini memungkinkan validasi lebih kustom sesuai dengan kebutuhan.

Contoh: Memvalidasi input hanya menerima angka dengan tepat 6 digit.

```
<label for="kodepos">Kode Pos (hanya angka 6
digit):</label>
<input type="text" id="kodepos" name="kodepos"
pattern="[0-9]{6}" required>
```

6. Atribut maxlength dan minlength

Atribut maxlength membatasi jumlah karakter yang dapat dimasukkan oleh pengguna, sementara minlength memastikan input memiliki jumlah karakter minimal tertentu.

Contoh: Input hanya menerima maksimal 10 karakter dan minimal 5 karakter.

```
<label for="username">Username (5-10 karakter):</label>
<input type="text" id="username" name="username"
minlength="5"
maxlength="10" required>
```

Perlu diperhatikan bahwa validasi input di sisi klien dengan menggunakan atribut HTML5 ini tidak sepenuhnya aman, karena pengguna masih dapat memanipulasi data sebelum mengirimkannya ke server. Oleh karena itu, tetap diperlukan validasi lebih lanjut di sisi server untuk memastikan keamanan dan integritas data. Atribut HTML5 tersebut lebih mengenai memberikan umpan

balik yang berguna dan meningkatkan pengalaman pengguna dalam mengisi formulir.

4.2.2 Menggunakan Pseudoclass untuk Visual Validasi

Menggunakan pseudo-class di Cascading Style Sheets (CSS) adalah cara yang efektif untuk memberikan tampilan visual yang berbeda pada elemen formulir berdasarkan validasi inputnya. Ketika input lulus validasi atau ketika pengguna menginteraksinya, kita dapat mengubah tampilan elemen tersebut menggunakan pseudo-class seperti: `valid`, `invalid`, `focus`, dll.

Berikut adalah contoh penggunaan beberapa pseudo-class untuk visual validasi:

1. Pseudo-class: `valid` dan `invalid`

Pseudo-class: `valid` diterapkan pada elemen input yang memiliki nilai yang valid sesuai dengan aturan validasi yang ditentukan, misalnya input email yang valid atau input angka dalam batas yang ditentukan. Pseudo-class: `invalid`, sebaliknya, diterapkan pada elemen input yang tidak valid.

Contoh untuk menyoroti input teks dengan border berbeda ketika valid dan tidak valid:

```
input:valid {  
  border: 2px solid green;  
}  
input:invalid {  
  border: 2px solid red;  
}
```

2. Pseudo-class: `focus`

Pseudo-class: `focus` digunakan untuk memberikan tampilan ketika elemen input sedang dalam fokus. Ini terjadi ketika pengguna mengklik atau menyorot elemen input tersebut.

Contoh untuk memberikan efek ketika input mendapat fokus:

```
input:focus {  
  outline: 2px solid blue;}  
}
```

3. Pseudo-class:placeholder-shown

Pseudo-class:placeholder-shown memungkinkan kita mengatur tampilan elemen input berdasarkan apakah teks placeholder ditampilkan atau tidak. Pseudo-class ini berguna untuk memberikan tampilan yang lebih intuitif saat pengguna mulai mengisi input.

Contoh untuk menyembunyikan label ketika placeholder ditampilkan:

```
label {  
  display: block;  
}  
input:not(:placeholder-shown) + label {  
  display: none;  
}
```

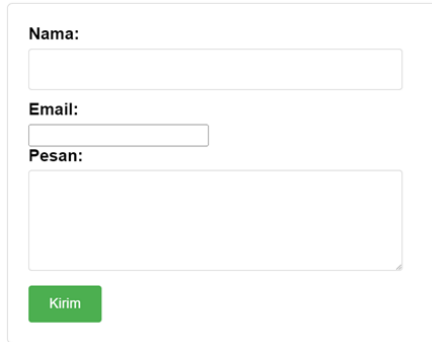
Dalam contoh di atas, ketika teks placeholder tidak ditampilkan (pengguna telah memasukkan teks ke dalam input), label akan disembunyikan menggunakan `display: none`.

Penggunaan pseudo-class dalam CSS memungkinkan desainer dan pengembang untuk memberikan umpan balik visual yang bermanfaat kepada pengguna selama proses validasi dan interaksi dengan formulir (Setiawan, 2018). Dengan kombinasi pseudo-class dalam CSS dan validasi HTML5, kita dapat meningkatkan kualitas dan kenyamanan pengalaman pengguna di situs web kita.

4.3 Menerapkan CSS pada Formulir

Untuk menerapkan CSS pada formulir, Anda dapat menggunakan sejumlah selector dan properti CSS untuk mengatur tampilan elemen-elemen formulir sesuai dengan desain yang diinginkan. Contoh penggunaan CSS dasar yang dapat Anda gunakan untuk memperindah tampilan formulir dapat dilihat pada Gambar 4.2.

Formulir dengan CSS



Gambar 4.2: Contoh Formulir Dengan CSS

Berikut adalah kode CSS dan HTML dasar yang dapat Anda gunakan untuk membuat tampilan formulir pada Gambar 2:

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulir dengan CSS</title>
  <style>
    /* Gaya untuk elemen form */
    form {
      max-width: 400px;
      margin: 0 auto;
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
      font-family: Arial, sans-serif;
    }
    /* Gaya untuk elemen input teks dan textarea */
    input[type="text"],
```

```
textarea {
  width: 100%;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 3px;
  font-size: 16px;
}
/* Gaya untuk elemen tombol Kirim */
input[type="submit"] {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 3px;
  cursor: pointer;
}
input[type="submit"]:hover {
  background-color: #45a049;
}
/* Gaya untuk label */
label {
  display: block;
  font-weight: bold;
  margin-bottom: 5px;
}
</style>
</head>
```

```
<body>
  <h1>Formulir dengan CSS</h1>
  <form action="/proses_formulir" method="post">
    <label for="nama">Nama:</label>
    <input type="text" id="nama" name="nama" required>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>
    <label for="pesan">Pesan:</label>
    <textarea id="pesan" name="pesan" rows="4"
required></textarea>
    <input type="submit" value="Kirim">
  </form>
</body>
</html>
```

Penjelasan:

1. CSS diletakkan dalam elemen `<style>` di dalam elemen `<head>` untuk mengatur tampilan formulir.
2. Properti seperti `max-width`, `margin`, `padding`, `border`, dan `border-radius` digunakan untuk mengatur tata letak dan bentuk formulir.
3. Properti `font-family`, `font-size`, dan `font-weight` digunakan untuk mengatur gaya teks di formulir.
4. Selector seperti `<input type="text">`, `<input type="submit">`, dan `<label>` digunakan untuk menggayakan elemen-elemen tertentu dalam formulir.
5. Penggunaan `pseudo-class: hover` pada tombol Kirim digunakan untuk memberikan efek saat tombol diklik.

Anda dapat mengganti nilai properti CSS sesuai dengan kebutuhan desain Anda. Selain itu, Anda juga dapat menambahkan lebih banyak properti dan selector untuk mengatur tampilan formulir dengan lebih rinci dan

menyesuaikan desain sesuai preferensi Anda. Untuk lebih detail, pseudo-elemen CSS lainnya akan dijelaskan pada Sub-bab 4.3.1 dan 4.3.2.

4.3.1 Menggunakan Pseudo-Element untuk Styling

Menggunakan pseudo-element di CSS memungkinkan Anda menambahkan elemen virtual ke dalam halaman dan memberikan gaya khusus pada elemen tersebut. Pseudo-element biasanya digunakan untuk mengubah tampilan elemen spesifik atau menambahkan dekorasi tertentu tanpa perlu menambahkan elemen HTML tambahan.

Di bawah ini adalah beberapa contoh penggunaan pseudo-element untuk styling:

1. Pseudo-Element::before dan::after

Pseudo-element::before dan::after memungkinkan Anda menambahkan konten sebelum atau setelah isi elemen yang sebenarnya. Anda juga dapat menggunakan pseudo-element ini untuk memberikan dekorasi atau ikon pada elemen.

Contoh untuk menambahkan simbol "✓" sebelum tombol Kirim:

```
input[type="submit"]::before {  
  content: "\2714"; /* Kode Unicode untuk simbol centang */  
  margin-right: 5px;  
}
```

2. Pseudo-Element::first-letter dan::first-line

Pseudo-element::first-letter memungkinkan Anda menerapkan gaya khusus pada huruf pertama dalam elemen, sedangkan::first-line mengatur gaya pada baris pertama dalam elemen.

Contoh untuk mengubah gaya huruf pertama pada paragraf:

```
p::first-letter {  
  font-size: 2em;  
  font-weight: bold;  
}
```

3. Pseudo-Element::selection

Pseudo-element::selection memungkinkan Anda mengubah gaya teks yang dipilih oleh pengguna.

Contoh untuk mengubah warna latar belakang teks yang dipilih menjadi biru dan warna teks menjadi putih:

```
::selection {  
  background-color: blue;  
  color: white;  
}
```

4. Pseudo-Element::placeholder

Pseudo-element::placeholder memungkinkan Anda meng gaya teks placeholder pada elemen input dan textarea.

Contoh untuk mengubah warna teks placeholder menjadi abu-abu:

```
input::placeholder,  
textarea::placeholder {  
  color: #999;  
}
```

Pseudo-element dituliskan dengan menggunakan "::" (dua titik dua kali) di depan namanya. Namun, beberapa pseudo-element tertentu seperti::before,::after, dan::placeholder juga bisa dituliskan dengan hanya satu ":" (satu titik dua kali). Meskipun demikian, disarankan untuk selalu menggunakan "::" agar sesuai dengan standar CSS3.

Pseudo-element adalah alat yang sangat berguna dalam CSS untuk menambahkan dekorasi dan styling tambahan tanpa perlu merubah struktur HTML atau menambahkan elemen tambahan. Dengan penggunaan yang bijaksana, Anda dapat mencapai desain yang menarik dan kreatif untuk halaman web Anda.

4.3.2 Membuat Tombol Submit yang Menarik

Untuk membuat tombol submit yang menarik, Anda dapat meng gaya tombol dengan menggunakan CSS untuk memberikan efek visual dan tampilan yang lebih menarik. Berikut adalah beberapa contoh gaya CSS yang dapat Anda gunakan untuk membuat tombol submit yang menarik:

1. Gaya Latar Belakang dan Warna

Anda dapat menggunakan properti background-color, color, dan border untuk mengatur latar belakang, warna teks, dan border tombol.

```
/* Gaya tombol submit */
```

```
input[type="submit"] {
  background-color: #4CAF50; /* Warna latar belakang */
  color: white; /* Warna teks */
  padding: 12px 20px; /* Padding tombol */
  border: none; /* Hapus border */
  border-radius: 5px; /* Radius sudut tombol */
  cursor: pointer; /* Ubah kursor menjadi tanda tangan
saat mengarahkan ke tombol */
}
```

2. Efek Hover dan Aktif

Anda dapat menambahkan efek saat pengguna mengarahkan kursor ke tombol (hover) atau saat tombol sedang ditekan (aktif).

```
/* Efek hover tombol */
input[type="submit"]:hover {
  background-color: #45a049; /* Warna latar belakang
berubah saat hover */
}
/* Efek tombol saat aktif (ditekan) */
input[type="submit"]:active {
  background-color: #3e8e41; /* Warna latar belakang
berubah saat tombol ditekan */
}
```

3. Gaya Shadow dan Transisi

Anda dapat menambahkan efek bayangan (shadow) dan transisi pada tombol saat ada perubahan status.

```
/* Tambahkan shadow saat hover */
input[type="submit"]:hover {
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}
/* Tambahkan efek transisi saat berubah status */
input[type="submit"] {
  transition: background-color 0.3s ease; /* Efek transisi
pada background-color */
}
```

4. Icon atau Simbo

Anda dapat menambahkan ikon atau simbol menggunakan pseudo-element::before atau menggunakan font icon seperti Font Awesome.

Contoh menggunakan pseudo-element::before:

```
/* Tambahkan simbol centang sebelum tombol */
input[type="submit"]::before {
  content: "\2714"; /* Kode Unicode untuk simbol centang */
  margin-right: 5px; /* Spasi antara simbol dan teks tombol */
}
```

5. Font dan Ukuran

Anda dapat menyesuaikan font dan ukuran teks tombol agar sesuai dengan desain Anda.

```
/* Gaya font dan ukuran teks tombol */
input[type="submit"] {
  font-family: Arial, sans-serif;
  font-size: 16px;
  font-weight: bold;
}
```

Anda dapat menggabungkan dan menyesuaikan properti-properti di atas untuk menciptakan tombol submit yang sesuai dengan desain situs web Anda. Pastikan untuk menguji tampilan tombol di berbagai perangkat dan browser untuk memastikan konsistensi tampilan.

4.4 Formulir dengan Beberapa Kolom

Membuat formulir dengan beberapa kolom memungkinkan Anda mengatur elemen-elemen formulir secara lebih terstruktur dalam beberapa baris atau kolom. Ini sangat bermanfaat ketika Anda ingin mengumpulkan beberapa jenis informasi sekaligus. Membuat formulir multi-kolom dengan CSS melibatkan beberapa aspek desain tata letak dan responsif agar formulir terlihat baik pada berbagai ukuran layar. Contoh pembuatan dapat dilihat pada Gambar 3.



The image shows a web form titled "Formulir Multi-Kolom dengan CSS". It is contained within a light gray border. On the left side, there are three labels: "Nama:", "Email:", and "Pesan:". To the right of each label is a corresponding input field. The "Nama:" and "Email:" fields are single-line text boxes, while the "Pesan:" field is a larger multi-line text area. At the bottom left of the form, there is a green button with the white text "Kirim".

Gambar 4.3: Contoh Formulir Dengan Beberapa Kolom

Berikut adalah kode CSS dan HTML dasar yang dapat Anda gunakan untuk membuat tampilan formulir pada Gambar 3:

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulir Multi-Kolom dengan CSS</title>
  <style>
    /* Gaya umum */
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }
    /* Gaya untuk kontainer formulir */
    .form-container {
      max-width: 800px;
      margin: 0 auto;
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
  </style>
</head>
</html>
```



```
/* Gaya untuk baris dalam formulir */
.form-row {
    display: flex;
    margin-bottom: 15px;
}
/* Gaya untuk label */
.form-label {
    flex: 1;
    font-weight: bold;
}
/* Gaya untuk input dan textarea */
.form-input {
    flex: 2;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
    font-size: 16px;
}
/* Gaya untuk tombol Kirim */
.submit-button {
    background-color: #4CAF50;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}
.submit-button:hover {
```

```
        background-color: #45a049;
    }
    /* Media query untuk tata letak responsif */
    @media (max-width: 600px) {
        .form-row {
            flex-direction: column;
        }
        .form-label,
        .form-input {
            flex: none;
            width: 100%;
        }
    }
</style>
</head>
<body>
    <h1>Formulir Multi-Kolom dengan CSS</h1>
    <div class="form-container">
        <form action="/proses_formulir" method="post">
            <div class="form-row">
                <label class="form-label" for="nama">Nama:</label>
                <input class="form-input" type="text" id="nama"
name="nama" required>
            </div>
            <div class="form-row">
                <label class="form-label"
for="email">Email:</label>
```

```
        <input class="form-input" type="email" id="email"
name="email" required>
    </div>
    <div class="form-row">
        <label class="form-label" for="pesan">Pesan:</label>
        <textarea class="form-input" id="pesan" name="pesan"
rows="4" required></textarea>
    </div>
    <input class="submit-button" type="submit"
value="Kirim">
</form>
</div>
</body>
</html>
```

Penjelasan:

1. `.form-container`: Ini adalah wadah utama untuk formulir, memberikan margin dan padding untuk mengatur tampilan. Maksimum lebar (`max-width`) diatur untuk memastikan formulir tidak terlalu lebar pada layar yang lebar.
2. `.form-row`: Ini adalah kelas untuk baris dalam formulir. Kami menggunakan `display: flex`; untuk memastikan elemen dalam baris berada dalam satu baris horizontal.
3. `label`: Flex item pertama dalam `.form-row`, yang memberi label pada elemen input atau `textarea`.
4. `input[type="text"]`, `textarea`: Flex item kedua dalam `.form-row`, yang memberikan elemen input atau `textarea` ruang yang lebih besar untuk mengisi informasi.
5. `input[type="submit"]`: Tombol Kirim memiliki gaya yang sama seperti contoh sebelumnya.

6. Pada media query (max-width: 600px), kami menggunakan tata letak flex berorientasi kolom untuk mengatasi tampilan yang lebih sempit, seperti tampilan pada perangkat seluler.
7. Kami mengatur lebar elemen label dan input menjadi 100% dalam tata letak berorientasi kolom untuk memastikan tampilan yang lebih baik pada tampilan seluler.

Dengan menggabungkan pendekatan ini, Anda dapat membuat formulir dengan beberapa kolom yang terorganisir dan mudah untuk diisi oleh pengguna. Anda dapat menyesuaikan gaya dan tata letak sesuai dengan desain yang diinginkan. Dengan pendekatan ini, Anda dapat membuat formulir multi-kolom yang responsif dan sesuai dengan berbagai ukuran layar.

Bab 5

Pengenalan CSS

5.1 Pengenalan Dasar CSS

CSS adalah singkatan dari Cascading Style Sheets. Secara bahasa, style page berarti mengalir ke bawah. CSS adalah bahasa scripting yang digunakan untuk menggambarkan tampilan dokumen markup. CSS selalu digunakan bersama dengan HTML, tetapi HTML tidak digunakan untuk mendesain halaman web, hanya untuk menampilkan konten (Anamisa and Mufarroha, 2020).

CSS (Cascading Style Sheets) adalah bahasa yang digunakan untuk mengubah tampilan web agar terlihat lebih menarik dan indah (Kode, 2023). CSS berperan dalam mendesain halaman web dengan menyesuaikan setiap elemen HTML agar terlihat menarik di browser (Abdulloh, 2022). Kusumawardani et al. (2023) mengatakan bahwa CSS adalah suatu bentuk pemrograman web yang dapat mengatur banyak elemen pada suatu halaman web agar seragam dan tentunya lebih terstruktur.

Dengan CSS kita dapat mengatur layout, warna, font, garis, background, animasi dan lainnya. Singkatnya, bahasa CSS sendiri digunakan untuk mendesain atau mengkonfigurasi tata letak halaman web. Seiring dengan perkembangan teknologi dan pemrograman, CSS tidak hanya digunakan dalam HTML, tetapi juga dapat digunakan untuk mendesain tampilan aplikasi Android. CSS digunakan dengan pengidentifikasi atau kelas.

Ada tiga cara untuk menyediakan referensi CSS:

1. Referensi ke file eksternal
2. Tulis CSS di elemen utama
3. CSS di dalam atribut style pada elemen

5.1.1 Sejarah dan Perkembangan CSS

Kode (2023), mengatakan bahwa sebelum adanya CSS belum ada bahasa yang digunakan untuk memformat dokumen (web). Kehadiran web terlihat membosankan dan tidak teratur. Ide asli memformat halaman di browser bukanlah hal baru. Pada tahun 1990, Tim Berners-Lee mengembangkan browser bernama NeXT Browser/Editor. Di browser ini kita bisa membuat style sheet. Namun, sintaks style sheet tidak dipublikasikan. Kemudian pada akhirnya, pada 10 Oktober 1994, Hakon Wium Lie menyusun proposal bahasa CSS dan mengirimkannya ke mailist [www-talk](#) dan [www-html](#).

November 1994 Wium Lie mempresentasikan rancangan proposal untuk CSS (Cascading Style Sheets) di konferensi Internet Chicago. Yang pertama menanggapi proposal ini adalah Bert Bos. Saat itu ia sedang mengembangkan web browser bernama Argo. Ia kemudian memutuskan untuk bergabung dengan Wium Lie untuk mengembangkan CSS.

April 1995 Bahasa skrip CSS diperkenalkan kembali pada World Wide Web Conference. Dalam hal ini, Wium Lie langsung memukul Bert Bos. Bert Bos memberikan ceramah tentang dukungan CSS untuk Argo dan Wium Lie mempresentasikan demo Browser Arena, yang dia modifikasi untuk mendukung penggunaan CSS.

Akhirnya, pada 17 Desember 1996, versi pertama CSS, CSS 1, dirilis. Mei 1998 Versi kedua CSS atau CSS 2 dirilis. Dukungan output media ditambahkan dalam rilis ini.

5.1.2 Perkembangan Versi CSS

(Anamisa and Mufarroha, 2020) dan (Kode, 2023) mengatakan bahwa ada beberapa versi CSS sejak rilis awal:

1. CSS 1 (1996): adalah versi pertama (17 Desember 1996)
CSS 1 mendukung pengaturan tampilan dalam hal:
 - a. Font

- b. Warna, teks, background, dan elemen lainnya
 - c. Text attributes, misalnya spasi antar baris, kata, dan huruf
 - d. Posisi teks, gambar, tabel, dan elemen lainnya
 - e. Margin, bording, dan padding
2. CSS 2 (1998): adalah versi kedua (Mei 1998)
CSS 2 dikembangkan untuk memenuhi kebutuhan akan format dokumen agar bisa ditampilkan di printer.
 3. CSS 2.1 (2011): dirilis pada 7 Juni 2011
 4. CSS 3 (2012): versi yang digunakan saat ini
CSS 3 dapat membuat animasi pada halaman web, di antaranya animasi warna dan animasi 3D. Fitur-fitur CSS 3 adalah:
 - a. Animasi
 - b. Efek teks
 - c. Efek pada kotak

Sintaks dasarnya sama untuk setiap versi. Hanya ada sedikit perbedaan pada fungsinya. Beberapa ditambahkan, yang lain dihapus.

5.1.3 Fungsi dan Kegunaan CSS

Anamisa and Mufarroha (2020), mengatakan bahwa CSS biasanya disimpan di dalam sebuah file berekstensi.css dan disematkan di dalam dokumen HTML untuk memberikan style pada halaman tersebut. Meski demikian, kita tetap dapat menambahkan sintaks CSS langsung di dalam dokumen HTML, di dalam tag `<style></style>` untuk mengakomodir penerapan style yang hanya digunakan di dalam dokumen tersebut. Kita juga dapat menyematkan sintaks CSS langsung di dalam atribut style pada suatu tag HTML untuk menerapkan style yang hanya digunakan pada tag HTML untuk menerapkan style yang hanya digunakan pada tag itu saja. Sama halnya dengan objek-objek lain yang ada dimuka buni ini.

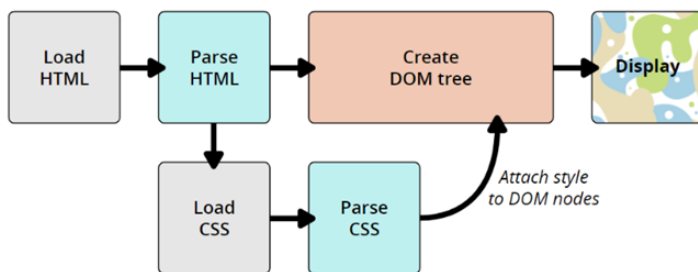
Adapun kelebihan dari CSS adalah sebagai berikut:

1. CSS sangat mudah untuk dipelajari dan digunakan
2. CSS memisahkan antara desain dan konten web/blog
3. Pengaturan desain dapat dilakukan seefisien mungkin

4. Karena satu css dapat dipakai beberapa kali, maka dapat menghemat penulisan kode.
5. Mempersingkat waktu kerja, baik saat membuat maupun saat memodifikasi halaman web/blog.

5.1.4 Cara Kerja CSS

Nayoan (2022), mengatakan bahwa CSS berfungsi saat browser memuat halaman web. Untuk menerapkan pengaturan rendering HTML yang telah ditentukan menggunakan kode CSS, prosesnya melibatkan beberapa langkah. Awalnya, browser akan memuat file HTML dan CSS (jika kode ditulis dalam CSS eksternal). Browser kemudian mengonversi keduanya menjadi Document Object Model (DOM). Ini adalah elemen yang mewakili file HTML dan CSS di memori perangkat pengunjung situs web. Setelah HTML dan CSS diubah menjadi DOM, browser menampilkan proses di mana browser menerapkan pengaturan dari kode CSS ke elemen HTML. Akibatnya, halaman web muncul di layar perangkat Anda.



Gambar 5.1: Cara Kerja CSS (Nayoan, 2022)

CSS memainkan peran besar dalam pembuatan halaman web karena tanpa CSS, tampilan dan nuansa situs web akan terlihat membosankan. Selain itu, dari sudut pandang desainer web, CSS akan sangat menghemat waktu karena dia tidak perlu mengetik konten yang sama (Puspa, 2022).

5.2 Cara Menulis CSS di HTML

Enterprise (2016) dan Kode (2023), mengatakan bahwa ada tiga cara menulis CSS di HTML:

1. Internal CSS: ditulis di dalam tag `<style>`;
2. Inline CSS: ditulis di atribut elemen HTML;
3. External CSS: ditulis di file CSS terpisah dengan HTML.

5.2.1 Internal CSS

Internal CSS adalah kode CSS yang ditulis di dalam tag `<style>`. CSS internal juga dikenal sebagai Embedded CSS.

`<style>` bisa ditulis di dalam tag `<head>`, bisa juga ditulis di dalam tag `<body>`. Tapi kebanyakan orang menulisnya dengan `<head>`. Tulis CSS di `<head>` hanya akan dibaca terlebih dahulu di `<body>`. Karena dibaca duluan, maka gaya yang akan digunakan adalah yang terakhir.

Contoh Internal CSS didefinisikan di dalam elemen `<style>`, di dalam bagian `<head>` halaman HTML:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Gambar 5.2: Contoh Internal CSS (w3schools, 2023)

5.2.2 Eksternal CSS

w3schools (2023), mengatakan bahwa CSS eksternal adalah kode CSS yang ditulis terpisah dari kode HTML. CSS eksternal ditulis dalam file khusus dengan ekstensi.CSS. Dengan lembar gaya eksternal, Anda dapat mengubah tampilan seluruh situs web hanya dengan mengubah satu file. Setiap halaman HTML harus menyertakan referensi ke file style sheet eksternal di dalam elemen `<link>`, di dalam bagian head.

Contoh Eksternal CSS ditentukan di dalam elemen `<link>`, di dalam bagian `<head>` halaman HTML:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Gambar 5.3: Contoh Eksternal CSS (w3schools, 2023)

Lembar gaya eksternal dapat ditulis dalam editor teks apa pun, dan harus disimpan dengan ekstensi.css. File.css eksternal tidak boleh berisi tag HTML apa pun.

5.2.3 Inline CSS

Inline CSS adalah kode CSS yang ditulis langsung pada atribut style dalam elemen HTML. Kode CSS ditulis langsung tanpa menggunakan kurung kurawal { ... }. Kemudian atribut tetap dipisahkan dengan titik koma. Inline CSS umumnya digunakan untuk menata elemen. Inline CSS jarang digunakan dalam proyek besar karena cukup sulit untuk mempertahankan kode jika Anda menggunakan terlalu banyak inline (Kode, 2023).

Gaya sebaris dapat digunakan untuk menerapkan gaya unik untuk satu elemen. Untuk menggunakan gaya sebaris, tambahkan atribut gaya ke elemen yang relevan. Atribut gaya dapat berisi properti CSS apa pun (w3schools, 2023).

Contoh Gaya inline didefinisikan dalam atribut "style" dari elemen yang relevan:

```
<!DOCTYPE html>
<html>
<body>

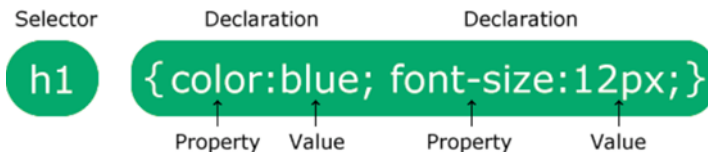
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

Gambar 5.4: Contoh Inline CSS (w3schools, 2023)

5.3 Sintaks Dasar CSS

(w3schools, 2023) mengatakan bahwa CSS punya sintaks atau tata cara penulisan kode yang harus diikuti. Aturan CSS terdiri dari pemilih dan blok deklarasi.



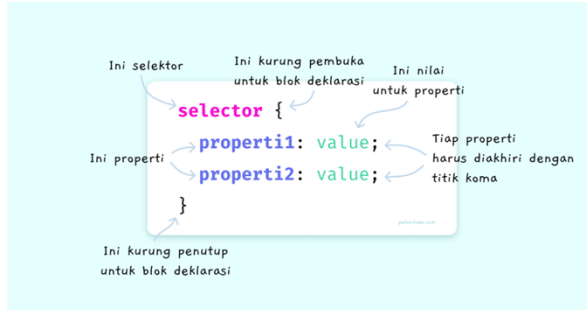
Gambar 5.5: CSS Syntax (w3schools, 2023)

Pemilih menunjuk ke elemen HTML yang ingin Anda gaya. Blok deklarasi berisi satu atau lebih deklarasi yang dipisahkan oleh titik koma. Setiap deklarasi menyertakan nama properti CSS dan nilai, yang dipisahkan oleh titik dua. Beberapa deklarasi CSS dipisahkan dengan titik koma, dan blok deklarasi diapit oleh kurung kurawal.

5.3.1 Struktur Dasar CSS

Kode (2023), mengatakan bahwa Struktur Dasar CSS terdiri dari tiga bagian:

1. Selektor
2. Blok Deklarasi
3. Properti dan nilainya



Gambar 5.6: Struktur Dasar CSS (Kode, 2023)

1. Selektor

Di CSS, selektor adalah pola yang digunakan untuk memilih elemen yang ingin Anda beri gaya.

2. Blok Deklarasi

Blok deklarasi adalah tempat kita menulis properti CSS atau properti yang akan ditugaskan ke elemen yang sudah selesai.

3. Properti dan Nilainya

Atribut adalah seperangkat aturan yang akan ditetapkan ke item yang dipilih.

5.3.2 Penulisan Komentar di CSS

Komentar sering digunakan untuk memberikan informasi tambahan dalam kode CSS. Mengomentari dalam CSS dilakukan dengan `/*` dan diakhiri dengan `*/`.

```

/* ini adalah komentar */

/*
  ini juga komentar
  yang ditulis lebih dari
  satu baris
*/

```

Gambar 5.7: Contoh Penulisan Komentar di CSS (Kode, 2023)

5.4 Memahami Selector di CSS

Kode (2023), mengatakan bahwa Selector adalah kata kunci dan simbol yang digunakan dalam CSS untuk memilih atau menyeleksi elemen HTML.

Ada 6 macam selektor di CSS:

1. Selektor Tag
Selektor Tag juga dikenal sebagai Type Selector. Selektor ini akan memilih item berdasarkan nama tag.
2. Selektor Class
Selektor Class adalah selektor yang memilih item berdasarkan nama kelas yang diberikan. Selektor Class memiliki titik di depannya.
3. Selektor ID
Selektor ID hampir seperti class. Perbedaannya adalah bahwa ID itu unik. Hanya dapat digunakan oleh satu elemen.
4. Selektor Atribut
Selektor Atribut adalah selektor yang memilih elemen berdasarkan atribut. Selektor ini hampir identik dengan pemilih tag.
5. Selektor Universal
Selektor Universal adalah selektor yang digunakan untuk memilih semua elemen dalam rentang tertentu. Selektor Universal sering digunakan untuk mengatur ulang CSS.
6. Pseudo Selektor
Pseudo Selektor adalah selektor untuk memilih elemen semu seperti status elemen, elemen sebelum dan sesudah, elemen ganjil, dan lain-lain.
Ada dua macam pseudo selektor:
 - a. Pseudo-class selektor untuk status elemen
 - b. Pseudo-element selektor untuk elemen semu dalam HTML

5.5 Menggunakan Warna di CSS

Warna adalah properti yang paling sering digunakan dalam CSS. Warna ditentukan menggunakan nama warna yang telah ditentukan sebelumnya, atau nilai RGB, HEX, HSL, RGBA, HSLA.

5.5.1 Mengetahui Properti Color

Properti color merupakan properti untuk memberikan warna pada teks.

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Gambar 5.8: Contoh Memberikan Warna pada Teks (w3schools, 2023)

5.5.2 Nilai yang Valid untuk Warna

Nilai warna pada properti `color` ditentukan dengan berbagai cara. Salah satunya adalah menuliskan nama warna dalam bahasa Inggris. Misalnya, warna teks untuk paragraf adalah merah, maka kita bisa dengan `color:red`. Selain menggunakan nama warna, ada cara lain yang valid seperti fungsi `rgb()`, kode heksa, dll.

1. Nama Warna

Nama warna sering digunakan karena lebih mudah diingat.

2. Kode Heksadesimal

Kode heksadesimal adalah kode yang menggunakan bilangan heksadesimal, dengan nilai dari 0 sampai f (16 dalam desimal).

Kode heksadesimal dimulai dengan tanda pagar (#), diikuti dengan kode heksadesimal. Kode ini dapat terdiri dari 3 atau 6 digit.

3. Fungsi `rgb()` dan `rgba()`

Fungsi `rgb()` adalah fungsi penghasil warna berdasarkan konsep warna RGB pada komputer.

4. Fungsi `hsl()` dan `hsla()`

Fungsi `hsl()` adalah konstruktor warna berdasarkan konsep warna HSL. HSL adalah singkatan dari Hue, Saturation, Lightness.

5. Gradien Warna

Gradien adalah kumpulan beberapa warna yang digabungkan bersama untuk membentuk gradien. Warna gradien dalam CSS biasanya dihasilkan menggunakan fungsi:

- a. `linear-gradient()` untuk gradien linear;
- b. `radial-gradient()` untuk gradien melingkar;
- c. `conic-gradient()` untuk memutar warna;

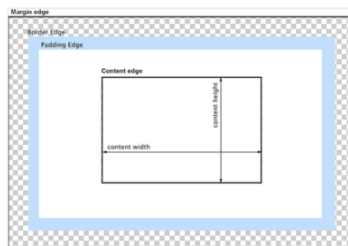
Setiap fungsi memiliki parameter yang hampir sama. Perbedaannya hanya pada setting awal. Fungsi `linear-gradient()` pertama-tama mengambil parameter orientasi, diikuti dengan warna yang akan dibaurkan. Parameter arah pada `linear-gradient()` bisa berupa nama arah seperti kiri, kanan, atas, bawah, maupun derajat seperti 90 derajat.

Bab 6

Model Kotak dan Tata Letak

6.1 Model Kotak

Saat meletakkan dokumen, mesin rendering browser mewakili setiap elemen sebagai kotak persegi panjang sesuai dengan model kotak dasar CSS standar. CSS menentukan ukuran, posisi, dan properti (warna, latar belakang, ukuran batas, dll.) dari kotak-kotak ini. Setiap elemen HTML ditampilkan dalam bentuk box (kotak). Jika diperhatikan, sebuah website umumnya terbagi menjadi beberapa area. Mulai dari menu di bagian atas, isi konten di bagian tengah, iklan di sisi kanan, sampai credit dan info kontak di bagian paling bawah. Setiap kotak terdiri dari empat bagian (atau area), ditentukan oleh tepinya masing-masing: tepi konten, tepi padding, tepi batas, dan tepi margin (Mozilla, 2023).



Gambar 6.1: Model Kotak

6.1.1 Area Konten

Area konten, yang dibatasi oleh tepi konten, berisi konten elemen yang "nyata", seperti teks, gambar, atau pemutar video. Dimensinya adalah lebar konten (atau lebar kotak konten) dan tinggi konten (atau tinggi kotak konten). Ini sering memiliki warna latar belakang atau gambar latar belakang.

Jika properti `box-sizing` diatur ke `content-box` (default) dan jika elemennya adalah elemen blok, ukuran area konten dapat ditentukan secara eksplisit dengan `width`, `min-width`, `max-width`, `height`, `min-height`, dan properti `max-height`.

6.1.2 Area Padding

Area padding, dibatasi oleh tepi padding, memperluas area konten untuk menyertakan padding elemen. Dimensinya adalah lebar kotak bantalan dan tinggi kotak bantalan. Ketebalan padding ditentukan oleh sifat `padding-top`, `padding-right`, `padding-bottom`, `padding-left`, dan shorthand padding

6.1.3 Area Border

Area Border, yang dibatasi oleh tepi batas, memperluas area padding untuk menyertakan batas elemen. Dimensinya adalah lebar kotak pembatas dan tinggi kotak pembatas. Ketebalan perbatasan ditentukan oleh properti `border-width` dan shorthand `border`. Jika properti `box-sizing` diatur ke `border-box`, ukuran area border dapat ditentukan secara eksplisit dengan properti `width`, `min-width`, `max-width`, `height`, `min-height`, dan `max-height`. Ketika ada latar belakang (warna latar belakang atau gambar latar belakang) yang diatur pada sebuah kotak, itu meluas ke tepi luar perbatasan (yaitu meluas di bawah perbatasan dalam urutan-z). Perilaku default ini dapat diubah dengan properti CSS `background-clip`.

6.1.4 Area Margin

Area margin, dibatasi oleh tepi margin, memperluas area perbatasan untuk menyertakan area kosong yang digunakan untuk memisahkan elemen dari tetangganya. Dimensinya adalah lebar kotak margin dan tinggi kotak margin. Ukuran area margin ditentukan oleh properti `margin-top`, `margin-right`, `margin-bottom`, `margin-left`, dan shorthand margin. Saat terjadi penurunan margin, area margin tidak ditentukan dengan jelas karena margin dibagi antar

kotak. Terakhir, perhatikan bahwa untuk elemen inline yang tidak diganti, jumlah ruang yang digunakan (kontribusi terhadap tinggi garis) ditentukan oleh properti line-height, meskipun border dan padding masih ditampilkan di sekitar konten.

Contoh di bawah ini akan diciptakan dua buah elemen HTML berupa <div> dan <p>. Elemen HTML berupa <div> merupakan elemen sebagai contoh konsep box model. Margin pada elemen ini sudah diatur dengan jarak 80px baik itu margin atas, bawah, kanan dan kiri. Padding diatur dengan jarak 40px untuk padding atas, bawah dan kiri. Sedangkan border diatur dalam keadaan solid dengan ketebalan 2px dan warna orange.

Elemen HTML yang kedua berupa elemen <p> yang berfungsi sebagai pembanding batas margin bawah pada elemen <div> (Saidal Faruq, 2022).

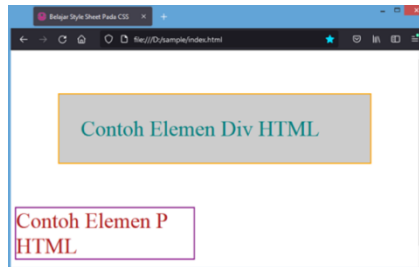
Script:

```
<!DOCTYPE html>
<html>
<head>
<title>Belajar Style Sheet Pada CSS</title>
<style>
  div{
    margin-top: 80px;
    margin-left: 80px;
    margin-right: 80px;
    margin-bottom: 80px;
    padding-top: 40px;
    padding-left: 40px;
    padding-bottom: 40px;
    border: solid 2px orange;
    font-size:40px;
    color: teal;
    background: #ccc;
```

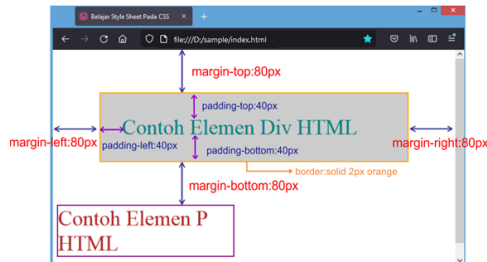
```

}
p{
  border: solid 2px purple;
  width:330px;
  font-size:40px;
  color:firebrick;
}
</style>
</head>
<body>
  <div>Contoh Elemen Div HTML</div>
  <p>Contoh Elemen P HTML</p>
</body>
</html>

```



Gambar 6.2: Contoh 1 Elemen HTML



Gambar 6.3: Contoh 2 Elemen HTML

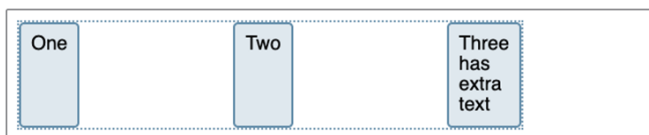
Pada contoh di atas, jika elemen HTML diciptakan sebagai elemen tunggal dan tidak ada elemen lainnya di sekelilingnya maka marginnya akan berbatasan dengan tepian layar

6.2 Tata Letak Kotak Fleksibel CSS

Modul tata letak kotak fleksibel CSS mendefinisikan model kotak CSS yang dioptimalkan untuk desain antarmuka pengguna, dan tata letak item dalam satu dimensi. Dalam model tata letak fleksibel, anak-anak wadah fleksibel dapat ditata ke segala arah, dan dapat "melenturkan" ukurannya, baik membesar untuk mengisi ruang yang tidak terpakai atau menyusut untuk menghindari meluapnya induk. Penyelarasan horizontal dan vertikal anak-anak dapat dengan mudah dimanipulasi.

Contoh dasar

Dalam contoh berikut wadah telah diatur ke `display: flex`, yang berarti bahwa tiga item anak menjadi item fleksibel. Nilai dari `justify-content` telah diatur `space-between` untuk menempatkan item secara merata pada sumbu utama. Jumlah ruang yang sama ditempatkan di antara setiap item dengan item kiri dan kanan rata dengan tepi wadah fleksibel. Anda juga dapat melihat bahwa item meregang pada sumbu silang, karena nilai default `align-items` menjadi `stretch`. Item meregang ke ketinggian wadah fleksibel, membuat masing-masing terlihat setinggi item tertinggi.



```
.box {  
    display: flex;  
    justify-content: space-between;  
}  
  
<div class="box">  
    <div>One</div>
```

```
<div>Two</div>
<div>Three
  <br>has
  <br>extra
  <br>text
</div>
</div>
```

Bab 7

Pengenalan Java Script

7.1 Apa itu Java Script ?

7.1.1 Sejarah dan Perkembangan JavaScript

Java Script Muthohir (2021) adalah bahasa script yang berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML. Jscript adalah bahasa script pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengekseskuan perintah-perintah disisi user, yang artinya disisi browser bukan disisi server web. Di mana objek tersebut dapat berupa suatu dokumen, URL, windows, frame form, button atau item lain yang semuanya itu mempunyai properti yang saling berhubungan dengannya dan masing-masing memiliki nama, lokasi, warna nilai dan atribut.

Jscript pertama kali diciptakan tahun 1995 oleh Brendan Eich, seorang karyawan Netscape. Awalnya, ia diminta untuk membuat bahasa scripting seperti Java yang dapat diterapkan di browser. Ia kemudian mendesain bahasa pemrograman baru hanya dalam waktu 10 hari dengan menggabungkan fitur-fitur yang terinspirasi dari Java, Scheme, dan Self. Versi pertama hanya terbatas untuk kalangan Netscape saja. Mulanya, bahasa pemrograman ini

dinamai Mocha, lalu sempat diubah menjadi LiveScript sebelum pada akhirnya berubah nama menjadi Jscript. Tak lama kemudian, Microsoft membuat bahasa baru yaitu Jscript yang digunakan pada browser miliknya (Internet Explorer). Pada dasarnya Jscript merupakan hasil adopsi dari Jscript.

Netscape lalu melakukan standarisasi dan di tahun 1997, Jscript berhasil diajukan ke ECMA Internasional dengan label ECMAScript. ECMAScript terus berkembang hingga tahun 1998, ECMAScript 2 diluncurkan disusul dengan ECMAScript 3 pada tahun 1999. ECMAScript mengalami banyak revisi dan peningkatan hingga menjadi bahasa pemrograman Jscript seperti sekarang yang andal untuk digunakan di berbagai model aplikasi. Di tahun 2005, pamor Jscript naik berkat adanya AJAX (Asynchronous JavaScript and XML). Library baru seperti JQuery dan MooTools dikembangkan untuk meminimalisir ketidakkonsistenan browser dan memudahkan untuk diterapkan pada design patterns. Hingga kini, Jscript menjadi bahasa skrip sisi klien yang banyak digunakan. Menurut data statistik, di tahun 2016 sebanyak 92% website menggunakan Jscript untuk menarik para pengunjung, termasuk situs-situs ternama seperti Google dan Facebook. Berikut ini tabel sejarah perkembangan Jscript untuk memudahkanmu memahaminya:

Tabel 7.1: Sejarah Perkembangan JavaScript

Tahun	ECMA	Browser
1995		Jscript ditemukan oleh Brendan Eich
1996		Netscape 2 dirilis dengan JavaScript 1.0
1997		Jscript diangkat menjadi ECMA standard (ECMA-262)
1997	ES1	ECMAScript 1 dirilis
1997	ES1	IE 4 menjadi browser pertama dengan ES1
1998	ES2	ECMAScript 2 dirilis
1998		Netscape 42 dirilis dengan Jscript 1.3
1999	ES2	IE 5 menjadi browser pertama dengan ES2
1999	ES3	ECMAScript 3 dirilis
2000	ES3	IE 5.5 menjadi browser pertama dengan ES3
2000		Netscape 62 dirilis dengan Jscript 1.5
2000		Firefox 1 dirilis dengan Jscript 1.5
2008	ES4	ECMAScript 4 discontinued
2009	ES5	ECMAScript 5 dirilis
2011	ES5	IE 9 menjadi browser pertama dengan ES5 *
2011	ES5	Firefox 4 dirilis dengan Jscript 1.8.5
2012	ES5	Dukungan penuh untuk ES5 di Safari 6
2012	ES5	Dukungan penuh untuk ES5 in IE 10

Tahun	ECMA	Browser
2012	ES5	Dukungan penuh untuk ES5 di Chrome 23
2013	ES5	Dukungan penuh untuk ES5 di Firefox 21
2013	ES5	Dukungan penuh untuk ES5 di Opera 15
2014	ES5	Dukungan penuh untuk ES5 di semua browser
2015	ES6	ECMAScript 6 dirilis
2016	ES6	Dukungan penuh untuk ES6 in Chrome 51
2016	ES6	Dukungan penuh untuk ES6 in Opera 38
2016	ES6	Dukungan penuh untuk ES6 in Edge 14
2016	ES6	Dukungan penuh untuk ES6 in Safari 10
2015	ES6	Dukungan penuh untuk ES6 in Firefox 52
2018	ES6	Dukungan penuh untuk berbagai browser

Kelebihan Jscript antara lain:

1. Mudah dipelajari

Jscript termasuk bahasa pemrograman yang mudah dipelajari. Bagi pemula, sintaks dan penulisan kodenya cukup ringkas hingga gampang untuk dimengerti. Sebagai perbandingan, berikut penulisan sintaks di Jscript dengan beberapa bahasa lainnya untuk menampilkan “Hello World!”

2. Fleksibel

Jscript adalah bahasa pemrograman yang fleksibel. Meski pemanfaatan utamanya pada sisi frontend, Jscript juga bisa dipakai untuk sisi server dengan adanya Node.js. Kamu bisa mengembangkan beragam jenis aplikasi seperti server, aplikasi mobile, dan desktop, game hingga console.

3. Interaktif dan Responsif

Jscript sangat cocok untuk membuat halaman web yang user-friendly. Saat ini semua produk digital baik aplikasi web ataupun mobile sangat mementingkan pengalaman penggunanya. Dengan Jscript, developer dapat merancang halaman web interaktif dengan beragam desain antarmuka yang kreatif.

4. Multiplatform Dan Cross Browser

Jscript bisa diimplementasikan untuk basis sistem operasi apa pun. Pengembangan frontednya bersifat lintas platform sehingga dapat digunakan untuk mengembangkan aplikasi yang berfungsi di

browser, desktop, bahkan di smartphone. Hampir semua browser juga didukung oleh Jscript sehingga dapat menampilkan video, animasi, dan media lain pada halaman web.

5. Ketersediaan framework dan library yang melimpah

Ketersediaan kerangka kerja dan library yang melimpah sangat membantu dalam menyediakan komponen bawaan agar pengembangan di Jscript menjadi lebih mudah.

7.1.2 Penerapan JavaScript dalam Pengembangan Web

Jscript pada awal perkembangannya berfungsi untuk membuat interaksi antara user dengan situs web menjadi lebih cepat tanpa harus menunggu pemrosesan di web server (Yusi Ardi Binarso, Eko Adi Sarwoko, 2012) (Pahlevi, Mulyani and Khoir, 2018) (Mariko, 2019) (Lay, 2017). Dalam perkembangan selanjutnya, Jscript tidak hanya berguna untuk validasi form, namun untuk berbagai keperluan yang lebih modern. Berbagai animasi untuk mempercantik halaman web, fitur chatting, efek-efek modern, games, semuanya bisa dibuat menggunakan JavaScript.

Akan tetapi karena sifatnya yang dijalankan di sisi client yakni di dalam web browser yang digunakan oleh pengunjung situs, user sepenuhnya dapat mengontrol eksekusi Jscript. Hampir semua web browser menyediakan fasilitas untuk mematikan Jscript, atau bahkan mengubah kode Jscript yang ada.

Selain itu penerapan Jscript dalam pengembangan web antara lain:

1. Membuat Website Interaktif

Fungsi utama Jscript bisa dibilang untuk membuat website terlihat menarik dengan konten-kontennya yang dinamis. Konten website yang bisa bergerak dengan otomatis tanpa perlu direload berkali-kali adalah salah satu pengaplikasian JavaScript.

2. Mengembangkan Aplikasi Mobile

Jscript juga digunakan untuk mengembangkan aplikasi di HP. Developer semakin dimudahkan membuat aplikasi mobile untuk dua sistem operasi sekaligus (Android dan iOS) berkat adanya framework Jscript khusus untuk aplikasi mobile

3. Menciptakan Game Berbasis Web

Fungsi Jscript yang lain adalah untuk membuat konten yang dinamis, dengan dengan menciptakan game berbasis web. Pengembangan game di Jscript didukung dengan framework khusus seperti Panda.js dan Playground.js.

4. Menjalankan Web Server

Jscript juga mampu dijalankan di sisi server dengan bantuan Node.js. Semenjak ada Node.js, Jscript digunakan di luar dari browser, misalnya untuk mengembangkan back-end (server), console, program desktop, mobile, game, dan lain sebagainya

7.2 Penyisipan JavaScript pada Halaman HTML

7.2.1 Menggunakan Script Tag

Untuk dapat memulai dan menjalankan program Jscript, ada dua perangkat lunak yang diperlukan yaitu browser dan teks editor, selain itu browser web yang akan digunakan harus mendukung Jscript.

Dua piranti pemrograman Jscript yaitu:

1. Teks Editor

Digunakan untuk menuliskan kode-kode Jscript, teks editor yang dapat digunakan antara lain notepad dan ultra edit. Teks editor adalah sebuah pengolah kata yang menghasilkan file dalam format ASCII murni

2. Web Browser

Digunakan untuk menampilkan halaman web yang mengandung kode-kode Jscript. Web browser yang digunakan harus mendukung Jscript. Browser yang dapat digunakan adalah internet explorer dan netdcape navigator. Selain itu browser web yang akan digunakan harus mendukung Jscript, dapat menggunakan Chrome, Firefox,

Internet Explorer, Opera ataupun browser lainnya yang mendukung Jscript. Jscript sebagai bahasa pemrograman yang berjalan di atas browser harus di tulis di dalam HTML.

Ada empat cara penulisan Jscript pada HTML

1. Tag <script>

Cara umum yang dipakai adalah menuliskannya dalam tag <script>.

Tag <script> bisa dibuat di dalam tag <head>, maupun di dalam tag <body>.

Menulis ke dalam output HTML menggunakan document.write ().

```
<html>
<h2> Penulisan dengan innerHTML</h2>
<p id="demo"></p>
<script>
Document.getElementById("demo").innerHTML="Belajar Web
Dengan Bahasa Java Script";
</script>
</html>
```

Hasil tampilan teks adalah sebagai berikut.



2. File Eksternal

Jika tidak ingin kode Jscript bercampur dengan HTML, bisa menuliskannya pada file terpisah. Caranya, adalah dengan menyimpannya pada sebuah file yang berekstensi.js, misalnya isinya sebagai berikut.

```
// file-eksternal.js
alert("Kode Javascript dari File Eksternal");
```

Kemudian menghubungkan file eksternal tersebut dengan file HTML, caranya gunakan tag <script> dengan atribut source untuk menentukan lokasi file Jscriptnya.

```
<html>
<title>Penulisan dengan external Javascript</title>
  <script src="file-eksternal.js">
  </script>
</head>
</html>
```

3. Atribut Event

```
<button onclick="alert('Ok Terima kasih!')">Klik donk!</button>
```

Selain event onclick ada juga event-event yang lain, seperti onsubmit, onload, ondoubleclick, onmouseover, onmouseout, dsb.

4. URL

Cara penulisan Jscript pada url, cara ini jarang digunakan, namun kita juga perlu mengetahuinya. Penulisan Jscript pada url menggunakan protokol Jscript. Pada bagian ini, akan diberikan contoh program untuk menampilkan teks sederhana ke layer monitor.

```
<script>
  document.write("<font size=8><center>");
  document.write("SELAMAT DATANG DI JAVASCRIPT <br>");
  document.write("Contoh          sederhana          menampilkan
  teks<br>");
  document.write("Dengan menggunakan JavaScript<br>");
</script>
```



7.2.2 Menyisipkan File JavaScript Eksternal

Cara untuk menginput kode Jscript ke dalam file HTML adalah dengan memindahkan kode Jscript ke dalam sebuah file terpisah, lalu diakses dari file HTML. Namun karena file Jscript ini berada di luar file HTML, disebut juga sebagai external Jscript. Cara ini lebih disarankan untuk kode Jscript yang panjang, dan dipakai oleh banyak halaman.

File Jscript external disimpan dalam ekstensi.js, seperti: kode.js, register.js, atau kodeku.js. Dari halaman HTML, kita memanggilnya menggunakan juga dengan tag `<script>`, tapi kali ini dengan tambahan atribut `src`. Atribut `src` berisi alamat dari file JavaScript tersebut, seperti berikut ini:

```
<script src="kode_javascript.js"></script>
```

Perhatikan bahwa tag `<script>` langsung ditutup dengan tag `</script>`. Di antara tag ini, tidak bisa ditulis kode internal JavaScript lain, tapi harus dipisah ke dalam tag tersendiri:

```
<script src="kode_javascript.js">
```

```
Tidak boleh ada kode lagi disini</script>
```

Sebagai contoh, akan dipindahkan perintah `alert("Hello World!!")` ke dalam file tersendiri. Silahkan buat file baru dengan nama `kode_javascript.js`, kemudian isi dengan perintah berikut:

```
alert("Hello World!!")
```

Contoh:

1

2

3

4

5

6

7

8

9

10

11

12 <!DOCTYPE html>

```
<html lang="id">
```

```
<head>
```

```
<title>Belajar JavaScript di Duniaikom</title>
<script src="kode_javascript.js"></script>
</head>
<body>
  <h1>Belajar JavaScript</h1>
  <p>Saya sedang belajar JavaScript di duniaikom.com</p>
  <p>Belajar Web Programming di Duniaikom.</p>
</body>
</html>
```

Dalam kode ini, perintah pemanggilan file kode_javascript.js ada di baris 5. Hasilnya, ketika di jalankan akan langsung tampil jendela alert seperti contoh di internal Jscript.

7.3 Variabel dan Tipe Data dalam JavaScript

7.3.1 Mendeklarasikan Variabel dengan Benar

Hampir setiap bahasa pemrograman ada yang namanya variable yang berguna untuk menyimpan data (Sabaruddin, 2019). Tanpa bantuan variabel sebuah program tidak bisa melakukan apa yang diinginkan. Pada Jscript pendeklarasian sebuah variabel sifatnya opsional, artinya Anda boleh mendeklarasikan atau tidak, Jika memberi nilai pada suatu variabel, maka dalam Jscript dianggap bahwa telah mendeklarasikan variabel tersebut.

Contoh:

```
Nama ="Fahra Ragita"
```

```
X = 2021
```

```
Y = 12122019
```

Untuk mendeklarasikan secara eksplisit, tulislah variabel tersebut dengan didahului kata kunci var.

Contoh

```
var nama;  
var nama="Fahra Ragita";  
var X = 2021;  
var Y;
```

Untuk mendeklarasikan beberapa variabel dalam satu baris, dapat dilakukan dengan menuliskan seperti berikut:

```
var A,B,C;
```

7.3.2 Memahami Berbagai Tipe Data (String, Number, Boolean, Array, Object)

Tidak seperti bahasa pemrograman lainnya, Jscript tidak mempunyai tipe data secara eksplisit (Sabaruddin, 2019). Hal ini dapat dilihat dari beberapa contoh yang telah disajikan sebelumnya dan mendeklarasikan variabel tapi tidak menentukan tipenya. Meskipun Jscript tidak mempunyai tipe data eksplisit, Jscript mempunyai tipe data Implisit.

Ada empat macam tipe data implisit yang dimiliki JavaScript yaitu:

1. Numerik, seperti 2351978, 2003948, 3.14, 100 dsb

Pada dasarnya JavaScript hanya mengenal dua macam tipe numerik, yaitu bilangan bulat dan bilangan real. Untuk bilangan bulat, Anda bisa merepresentasikan dengan basis desimal, oktal, atau heksadesimal.

Contoh:

```
var A = 100;  
var A = 0x2F;
```

Untuk pendeklarasian tipe bilangan real, Anda bisa menggunakan tanda titik atau notasi ilmiah (notasi E).

Contoh:

```
var a = 123.567  
var b = 1.234567E+3
```

2. String, seperti "Asep", "Mei", "Juli", "Kenapa Mundur", "JavaScript" dsb

Untuk mendeklarasikan tipe string dapat dilakukan dengan cara menuliskan string di antara tanda tunggal (‘) atau tanda petik ganda (”);

Contoh:

```
var A = ‘Ini pendeklarasian String’;  
var C = “Ini juga string”;
```

3. Boolean, hanya bernilai true atau false

Tipe Boolean hanya mempunyai nilai True atau false. Tipe ini biasanya digunakan untuk mengecek suatu kondisi atau keadaan.

Contoh

```
Var X = (Y>90);
```

Pada contoh ini menunjukkan bahwa jika Y lebih besar dari 90 maka X akan bernilai

True.

4. Null, yaitu variable yang tidak diinisialisasi

Tipe Null digunakan untuk merepresentasikan variabel yang tidak diberi nilai awal (inisialisasi)

```
var x;
```

5. Array, yaitu struktur data yang digunakan untuk menyimpan sekumpulan data dalam satu tempat. Setiap data dalam Array memiliki indeks, sehingga akan mudah memprosesnya serta indeks array selalu dimulai dari angka nol (0).

Syntax: var array_name = [item1, item2,...];

Contoh:

```
<script>
```

```
  Var
```

```
  kota=[ ‘Yogyakarta’, ‘Semarang’, ‘Jakarta’, ‘Surabaya’ ];
```

```
    document.write(kota[2]);
```

```
</script>
```

```
  Hasilnya: Jakarta
```

6. Object, yaitu sekumpulan list dari tipe data primitif (terkadang juga tipe data reference) yang menyimpan nilai dengan konsep berpasangan name-value. Tiap item (yang lebih dikenal dengan nama

variabel) disebut dengan property, dan function disebut dengan method.

Contoh:

```
var aContohObyek = {  
    namadepan: "Rizali",  
    namabelakang: "Asep Pratama"  
}
```

Berdasarkan contoh diatas, namadepan dan namabelakang termasuk property sedangkan valuenya adalah Rizali dan Asep Pratama.

7.4 Menggunakan Komentar pada Kode JavaScript

7.4.1 Jenis Komentar dalam JavaScript

Sama seperti bahasa pemrograman yang lainnya, Jscript juga menyediakan fasilitas untuk menuliskan komentar, komentar ini berguna bila nantinya Anda atau orang lain membaca suatu program. Pemberian komentar dalam Jscript dapat dilakukan dengan dua cara yaitu dengan menuliskan komentar setelah tanda garis miring dua kali.

Contoh:

```
//ini cara menuliskan komentar
```

atau

```
/*
```

```
ini juga cara menuliskan komentar apabila lebih
```

```
dari satu baris
```

```
*/
```

7.4.2 Penggunaan Komentar untuk Dokumentasi Kode

Ada beberapa jenis komentar dalam JavaScript yaitu single line comments atau komentar baris tunggal yang mengomentari satu baris atau sebagian dari satu baris dan multi line comments atau komentar multi baris yang mengomentari blok kode. Komentar multi baris ini lebih sering digunakan untuk dokumentasi formal. Pada JavaScript juga dikenal istilah Comment Out yang berarti menjadi bagian dari kode dan dikelilingi dengan simbol komentar JavaScript. Comment Out tidak dijalankan oleh browser yang berarti komentar ini tidak akan ditampilkan.

Berikut adalah contoh melakukan comment out pada single line comment di bawah ini:

```
//document.getElementById("text").innerHTML = "Selamat";  
document.getElementById("text2").innerHTML = "Belajar  
JScript";
```

Contoh berikut ini adalah ilustrasi dari comment out dari suatu blok kode:

```
/*document.getElementById("text").innerHTML = "SELAMAT";  
document.getElementById("text2").innerHTML = "Belajar";*/
```

Lebih jauh lagi, komentar JScript memungkinkan untuk mengomentari sebuah baris atau line, bagian dari baris tersebut, atau sebuah blok dari kode. Fitur ini dapat digunakan untuk melakukan

Comment Out adalah bagian dari kode, untuk itu sebaiknya jangan menghapus bagian tersebut untuk melihat apakah ada bug dalam baris tersebut.

Single line comments (komentar baris tunggal)

Komentar baris tunggal ini digunakan untuk mengomentari sebuah bagian dari suatu baris atau satu baris penuh dari kode yang ada di Jscript, dapat digunakan untuk menjelaskan code atau melakukan debugging juga bisa menggunakannya untuk melakukan comment out untuk mencegah eksekusi dari browser. Berikut ini adalah contoh penggunaan single line comment untuk menjelaskan baris dari kode di bawah ini.

```
//Ini adalah kode dalam javascript  
document.getElementById("item").innerHTML = "ini hanya  
contoh";
```

Pada contoh berikut ini, komentar ditulis dalam baris yang sama sebagai kode itu sendiri:

```
var n = 17;//deklarasi variabel n
var hasil = n + 23;//variabel hasil
```

Untuk komentar satu baris, harus menulis dua garis miring://. Apa pun yang terjadi setelah//hingga jeda baris diperlakukan sebagai komentar JavaScript dan tidak dieksekusi sebagai kode.

Multi line comments (komentar multi baris)

Komentar multi baris JavaScript memiliki tujuan yang sama dengan komentar baris tunggal. Blok komentar JavaScript dimulai dengan dua simbol:/* . Untuk mengakhiri komentar, gunakan simbol ini */lagi. Segala sesuatu di antara simbol-simbol ini akan diperlakukan sebagai komentar di JavaScript. Penggunaan komentar multiline JavaScript untuk meninggalkan komentar panjang atau mengomentari bagian kode yang lebih besar saat debugging. Contohnya sebagai berikut:

```
/* Contoh penggunaan multi
line comments dalam javascript */
document.getElementById("text").innerHTML = "Selamat";
document.getElementById("text2").innerHTML = "Belajar
JScript";
```

Bab 8

Variabel, Tipe Data, dan Operator

8.1 Variabel

Variabel merupakan identitas yang digunakan dalam menampung sebuah nilai yang digunakan untuk menyimpan sebuah informasi dan data yang kemudian akan di proses, terletak pada memori komputer (Sidik, 2011). Nilai yang dihasilkan akan ditampung pada satu slot di memory dan akan diberi nama yang merupakan identitas dari sebuah variabel. Adapun nilai yang dihasilkan oleh variabel seperti number (angka), objek, string (teks), array, dan sebagainya.



Gambar 8.1: Ilustrasi variabel

Pada gambar diatas menunjukkan bahwa gambar pertama yaitu variabel tanpa isi yang berarti bahwa meminta ruang kosong pada memori, sedangkan variabel dengan nilai yaitu mengisi data dalam ruang memori. Dalam membuat variabel pada javaScript yaitu menggunakan kata kunci var (Tagger, 2014) yang kemudian diikuti dengan nama variabel dan nilainya. Contohnya adalah sebagai berikut.

```
keyword nama-variabel;  
//atau  
keyword nama-variabel = value;
```

Gambar 8.2: penamaan variabel

Keterangan (sidik, 2018):

1. Keyword dapat diisi dengan var
2. Nama-variabel yaitu nama yang bisa ditentukan sesuai dari keinginan programmer, namun tetap harus menyesuaikan dengan aturan.
3. = merupakan operator dari assignment
4. value merupakan nilai yang akan dimasukkan kedalam sebuah variabel
5. : (semicolon) merupakan tanda yang digunakan untuk mengakhiri sebuah statement, namun di javaScript semicolon bersifat opsional.

```
var judul = "Pemula Belajar JavaScript";  
var visitorCount = 90242;  
var url = "https://www.pemula_belajar_javascript.com";
```

Gambar 8.3: contoh kasus variabel

Pada JavaScript terdapat beberapa aturan dalam penamaan variabel adalah sebagai berikut (Sidik, 2018):

1. Variabel case sensitive, perlu diperhatikan huruf besar dan kecil contohnya variabel Name akan berbeda dengan variabel name.
2. Diawali dengan sebuah huruf dan karakter dengan garis bawah underscore (`_`), atau tanda dollar (`$`)
3. Operasi pada ekpresi tidak boleh mengandung spasi, karakter dan tanda baca

4. Bukan kata kunci keyword pada javaScript yang sudah digunakan seperti let, var, function, class, if, for dan lain sebagainya.
5. Dapat berisi huruf, angka, underscore (_) atau tanda dollar (\$)

Adapun contoh dari penamaan variabel yang salah dan benar. Berikut variabel yang benar.

```
var namaDepan = "Syarifah";
var _namabelakang = "Fitra";
var nama234 = "Syarifah Fitra"
```

Gambar 8.4: contoh kasus variabel yang benar

Berikut variabel penamaan yang salah.

```
var nama?;
var 123nama = "Syarifah"
var class = "90242"
var nama lengkap = "Syarifah Fitra"
```

Gambar 8.5: contoh kasus variabel yang salah

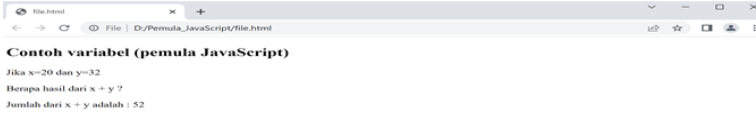
Pada dasarnya seorang progamer menggunakan gaya penulisan dengan menggunakan gaya camelCase, contoh yang dimaksud dengan gaya penulisan tersebut adalah namaDepan, namaBelakang. Pada penulisan sebuah variabel, seorang programmer biasanya jika dalam penulisan satu kata, maka menggunakan huruf kecil semua sedangkan jika terdiri dari dua kata atau lebih, huruf pertama dari kedua huruf menggunakan huruf besar dan seterusnya. Contoh deklarasi variabel tipe angka adalah sebagai berikut:

```
D:\> Pemula_Bootstrap > file.html ...
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5
6 <h2>Contoh variabel (pemula JavaScript)</h2>
7 <p>Jika x=20 dan y=32</p>
8 <p>Berapa hasil dari x + y ?</p>
9
10 <p id="hasil"></p>
11 <script>
12   var x = 20;
13   var y = 32;
14   var hasil = x + y;
15
16   document.getElementById("hasil").innerHTML = "Jumlah dari x + y adalah : " + hasil;
17 </script>
18 </body>
19 </html>
```

Gambar 8.6: Contoh kasus variabel tipe angka

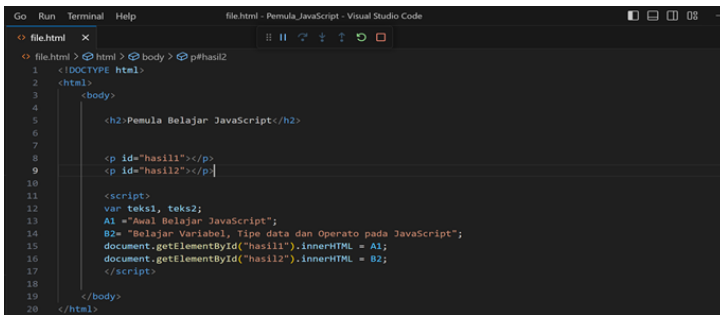
Pada baris penempatan variable merupakan contoh deklarasi yang berada pada baris 12,13 dan 14 pada JavaScript, baris 16 dengan kode + hasil yaitu dengan

memanggil variable dalam menampilkan sebuah variable dari variabel x dan y. Berikut hasil source code dari variabel tipe angka.



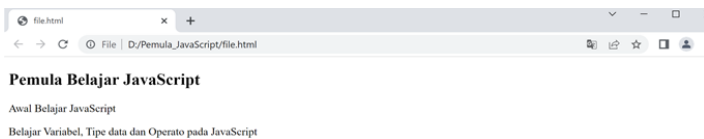
Gambar 8.7: Hasil variabel tipe angka

Adapun contoh variabel dengan tipe string adalah sebagai berikut.



Gambar 8.8: Contoh kasus variabel tipe string

Pada baris 12 merupakan deklarasi variabel, pemberian nilai pada variabel A1 dan A2 yang terletak pada baris 13 dan 14. Pemanggilan variabel baris 14 dan 15 merupakan hasil dari contoh kasus di mana untuk menghasilkan tampilannya berada pada baris 8 dan 9. Berikut hasil source code pada kasus variabel tipe string.



Gambar 8.9: Hasil source code variabel tipe string.

8.2 Tipe Data

Tipe data merupakan pengelompokan data berdasarkan apa yang dipahami oleh komputer, kepada compiler atau interpreter bagaimana programer ingin mengolah sebuah data. Konsep typeless programming language pada JavaScript pada tipe data dalam sebuah variabel kemudian ditentukan dari nilai yang disimpan. Pada JavaScript terdapat jenis-jenis tipe data yang dibedakan menjadi tipe data primitive dan non primitif. Pada tipe dat primitf adalah tipe data yang dapat menyimpan satu nilai dalam satu waktu di mana tipe data tersebut dianggap sama jika nilainya sama.

Adapun tipe data primitive adalah sebagai berikut.

1. String
2. Numerik
3. BigInt
4. Boolean
5. Undefined
6. Null
7. Symbol

Tipe data non primitif adalah menyimpan lebih dari satu nilai dalam satu waktu, di mana tipe data ini dianggap berbeda meskipun nilainya sama. Adapun Tipe data Non-Primitif adalah sebagai berikut.

1. Object
2. Array

8.2.1 Tipe data String

Pada Tipe data string yaitu tipe data yang mewakili atas data teks dan karakter (Nasir, 2011). Tipe data string dapat dimulai dengan petik tunggal maupun petik ganda namun setelah memulai tanda petik maka mengakhiri dengan tanpa petik pula, dengan memasukkan karakter diantara tanda petik tersebut, tipe data berupa huruf, tanda baca, angka dan rangkaian huruf dan angka. Data string juga dapat disebut dengan data rangkaian karakter. Adapun penggunaan tipe data ini adalah sebagai berikut.

```
"Syarifah Fitra Ramadhani"  
'Gorontalo'  
"345"  
"syarifah@gmail.com"
```

Gambar 8.10: Tipe data String

8.2.2 Tipe data Numerik

Tipe data ini adalah tipe data yang lebih dikenal dengan operasi matematika atau biasa disebut dengan proses perhitungan (Riza, 2022) seperti pembagian, perkalian, pengurangan dan penjumlahan, di mana bilangan yang dimaksud dapat berupa angka positif dan negatif yang biasa terdiri dari bilangan desimal, pecahan dan bilangan bulat. Adapun contoh dari bilangan numerik adalah sebagai berikut.

```
const bulatPositif = 10;  
const bulatNegatif = -10;  
const desimalPositif = 2.5;  
const desimalNegatif = -2.5;
```

Gambar 8.11: Tipe data Numerik

8.2.3 Tipe data BigInt

Pada tipe data ini dapat menyimpan bilangan bulat tanpa batasan nilai, di mana tipe ini baru dikenalkan pada ES2020. Namun pada tipe data setiap nilai harus diakhiri dengan huruf 'n', adapun contoh dari tipe data BigInt adalah sebagai berikut.

```
const A = 9999999999999999992n;  
console.log(A); // 455555555555553n
```

Gambar 8.12: Tipe data BigInt

8.2.4 Tipe data Boolean

Tipe data Boolean adalah tipe data yang bernilai true dan false (Rakhmadani, 2020), yang hanya menampilkan tipe data yang berniali benar atau salah. Jika nilai bernilai pada JavaScript bernilai 0 maka data tersebut masuk kekategori false atau salah dan sebaliknya. Pada dasarnya tipe data ini digunakan untuk menyatakan sebuah kondisi, di mana kondisi tersebut berada dalam proses

pengulangan maupun proses pemilihan. Adapun contoh dari tipe data Boolean adalah sebagai berikut.

```
var diaPintar = true;
var diaNakal = false;
```

Gambar 8.13: Tipe data Boolean

8.2.5 Tipe data Undefined

Pada tipe data ini adalah tipe data yang tidak diberi nilai di mana nilai yang diberikan tidak dideklarasikan tanpa inisialisasi. Pada kasus tipe data undefined hanya berlaku pada variabel `let` dan `var` (Putawa, 2022). Adapun contoh dalam tipe data ini adalah sebagai berikut.

```
var diaPintar = true;
var diaNakal = false;
```

Gambar 8.14: Tipe data Undefined

8.2.6 Tipe data Null

Pada tipe data null digunakan untuk mewakili ketidakhadiran yang disengaja dari sebuah nilai yang berada pada objek (Hendrawan, 2023). Pada dasarnya tipe data ini nanti akan memiliki sebuah variabel namun, sebelum penentuan variabel tersebut maka langkah yang dilakukan yaitu dengan menetapkan null ke variabel tanpa memiliki nilai apapun. Adapun contoh pada tipe data ini adalah sebagai berikut.

```
let percobaan = null;
console.log(percobaan); // null

percobaan = {
  nama: 'Syarifah',
  umur: 22,
  jurusan: 'Sistem Informasi',
  batch: '255'
};
percobaan.log(percobaan); // {nama: "Syarifah", umur: 22, jurusan: "Sistem Informasi", batch: "255"}
```

Gambar 8.15: Tipe data Null

8.2.7 Tipe Data Symbol

Tipe data symbol biasa disebut dengan nilai Symbol, tipe data ini diperkenalkan pada JavaScript ES2015. Nilai Symbol ini bersifat anonim, yaitu nilai symbol yang berjalan secara dinamis dapat menghasilkan nilai

yang unik dan tidak dapat dilihat. Adapun contoh tipe data symbol adalah sebagai berikut.

```
const x = Symbol();
const z = Symbol();

console.log(x === z); // false
console.log(x); // Symbol()
console.log(z); // Symbol()
```

Gambar 8.16: Tipe data Symbol

Tipe data Symbol biasanya sebagai nama kunci dikarenakan properti baru yang terdapat dalam sebuah objek agar tipe menempa properti objek yang sudah ada. Setelah membahas tipe data primitive maka selanjutnya akan dibahas tentang tipe data non-primitif. Pada dasarnya pada tipe data non primitif terdapat beberapa tipe data antara lain WeakSet, WeakMap, Set, Array, Object, dan Map. Namun, pada buku ini kita akan membahas 2 diantara beberapa tipe data non primitive yaitu tipe data object dan Array.

8.2.8 Tipe Data Object

Tipe data objek adalah sebuah tipe data kompleks yang dapat menyimpan kumulatif nilai dengan tipe data yang berbeda, yang berisikan kunci dan nilai yang lebih dikenal dengan sebutan key dan value. Dalam membuat object di JavaScript dapat menggunakan beberapa cara yaitu antara lain Object (Ilham, 2019). create(), Object(), dan dapat pula dengan menggunakan notasi literal. Namun pada implementasinya yang paling sering digunakan oleh programmer yaitu dengan menggunakan notasi literal. Pada saat membuat objek literal maka akan dibungkus dengan tanda kurung kurawal {}, dan jika property dalam sebuah system lebih dari satu property maka akan dipisahkan dalam bentuk koma. Adapun contoh tipe data Object adalah sebagai berikut.

```
let obj = {
  nama: 'Syarifah',
  umur: 22,
};
```

Gambar 8.17: Tipe data Object

Pada contoh percobaan diatas objek diberi nama obj, nama merupakan sebuah kunci sedangkan syarifah merupakan sebuah nilainya, dan juga begitupun dengan umur, yang kemudian diberi tanda kurung kurawal. Dalam proses mengakses nilai tertentu dapat menggunakan cara yaitu dengan menggunakan

nama objek kemudian diberi kunci, namun diantara objek dan kunci di beri tanda titik (`obj.umur`). Adapun contoh pada kasus tersebut adalah sebagai berikut.

```
let obj = {
  nama: 'Syarifah',
  umur: 22,
};

console.log(obj.nama); // Syarifah
```

Gambar 8.18: Tipe data Objek 2

Dalam mengubah sebuah nilai kunci objek maka diperlukan operator assignment = yang kemudian diikuti dengan nilai yang akan dimasukkan.

```
let obj = {
  nama: 'Syarifa',
  umur: 22,
};

obj.nama = 'fitra';

console.log(obj.nama); // fitra
```

Gambar 8.19: Tipe data Objek 3

Selain pada tipe data objek dapat mengubah sebuah nilai, namun tipe data ini juga dapat menambah kunci dan nilai baru. Adapun contoh dalam kasus tersebut adalah sebagai berikut

```
let obj = {
  nama: 'Syarifa',
  umur: 22,
};

obj.job = 'dosen';

console.log(obj); // {nama: "Syarifah", umur: 22, job: "dosen"}
```

Gambar 8.20: Tipe data Objek 4

8.2.9 Tipe Data Array

Tipe data Array adalah tipe data yang dapat digunakan tanpa property seperti objek, namun dapat menyimpan beberapa nilai. Nilai pertama atau indeks pertama dalam array memiliki indeks 0, kemudian nilai 1 dan seterusnya (Wijaya, 2015). Nilai array literal ditandai dengan tanda kurung siku [], dan jika nilai pada sebuah kasus melebihi satu nilai maka untuk memisahkannya dengan menggunakan tanda koma (,), dan dapat memasukkan array dan objek

dalam sebuah kasus dalam tipe data array. Adapun contoh kasus pada buku ini adalah sebagai berikut.

```
let arr = [
  'Syarifah',
  22,
  true,
  undefined,
  null,
  { nama: 'Syarifah' },
  [1, 2, 3, 4],
];
```

Gambar 8.21: Tipe data Array

Jika ingin mengakses elemen array dan mengubah elemen array dapat menggunakan cara sebagai berikut.

```
let arr = [
  'Syarifah',
  22,
  true,
  undefined,
  null,
  { nama: 'Syarifah' },
  [1, 2, 3, 4],
];

console.log(arr[0]); // Syarifah
console.log(arr[3]); //true
```

Gambar 8.22: Mengakses elemen array

```
let arr = [
  'Syarifah',
  22,
];
arr[1]=23;
console.log(arr); // ["Syarifah", 21]
```

Gambar 8.23: Mengubah elemen array

8.3 Operator

Operator merupakan alat untuk melakukan pemrosesan dan manipulasi data terhadap tipe data pada JavaScript yang merupakan tindakan, operasi, atau dalam proses perhitungan (Dermawan, 2022). Pada operator dikenal dengan sebutan operan yaitu di mana nilai atau variabel dalam sebuah operasi. Pada Operator dalam JavaScript ada beberapa jenis adalah sebagai berikut.

1. Operator Aritmatika
2. Operator Penugasan
3. Operator Logika
4. Operator Perbandingan

8.3.1 Operator Aritmatika

Pada operator ini menggunakan symbol dengan melakukan operasi matematika berupa pengurangan, pembagian, penjumlahan, perkalian dan lain sebagainya yang dilakukan dengan pemrosesan data berupa data numerik (Suryana, 2020). Pada operator ini jika terdapat kolom prioritas berupa tanda kurung () maka yang akan dikerjakan terlebih dahulu yaitu nilai prioritas pada nilai yang berada dalam kurung.

Tabel 8.1: Operator Aritmatika (Tarigan, 2020)

Operator	Deskripsi	Contoh	Hasil
+	Penjumlahan	$5 + 2$	7
-	Pengurangan	$5 - 2$	3
*	Perkalian	$5 * 2$	10
/	Pembagian	$5/2$	2.5
%	Modulus (Sisa bagi)	$5 \% 2$	1
**	Ekspensial (pemangkatan)	$5 ** 2$	25

Dalam operator aritmatika bukan hanya operator yang biasa digunakan pada tabel 8.1 namun ada juga operator yang digunakan dalam aritmatika adalah sebagai berikut.

Operator Increment

Pada increment (++) dibedakan menjadi 2 yaitu increment postfix (n++) mengembalikan nilai sebelum penjumlahan dan mengembalikan setelah penjumlahan disebut dengan increment prefix (++n). Adapun pada contoh kasus operator increment dapat dilihat pada gambar 8.23 (Harris, 2021).

```
// Increment Postfix
let a = 1;
a++;
console.log(a); // 2

// Increment Prefix
let b = 1;
++b;
console.log(b); // 2
```

Gambar 8.24: Increment Postfix dan Prefix

Sedangkan Decrement adalah (--) digunakan untuk mengembalikan nilai sebelum pengurangan yang disebut dengan postfix (n--) dan setelah pengurangan disebut dengan decrement prefix (--n) (Harris, 2021).

```
let z = 1;
z--;
console.log(z); // 0

let x = 1;
--x;
console.log(x); // 0
```

Gambar 8.25: Decrement Postfix dan Prefix

8.3.2 Operator Penugasan

Operator penugasan atau dikenal dengan symbol = digunakan untuk memberikan nilai terhadap variabel apapun. Beberapa operator dapat digabungkan dan dapat digunakan pada tipe data number kecuali += digunakan untuk penggabungan string (Harris, 2021).

Tabel 8.2: Operator Penugasan (Wibowo, 2017)

Operator	Contoh	Sama dengan
=	A = B	A = B
+=	A += B	A = A + B
-=	A -= B	A = A - B
*=	A *= B	A = A * B
/=	A /= B	A = A / B
%=	A %= B	A = A % B
**=	A **= B	A = A ** B

8.3.3 Operator Logika

Pada operator logika digunakan pernyataan seperti if. Logika && dan || yaitu menggabungkan nilai agar menghasilkan nilai Boolean yang sesuai dengan logika, sedangkan ! adalah negasi (Albar, 2021).

Tabel 8.3: Operator Logika (Albar, 2021)

Operator	Deskripsi
&&	And (dan)
	Or (atau)
!	Not (tidak)

Operator And menghasilkan false jika satu operan bernilai false. Adapun contoh kasus and dapat dilihat sebagai berikut.

```
const sudahTidur = true;
const sudahBelajar = true;

if (sudahTidur && sudahBelajar) {
  console.log('Selamat Anda sudah Tidur dan Belajar');
}
```

Gambar 8.26: Operator Logika And

8.3.4 Operator Perbandingan

Pada operator ini yaitu membandingkan dua buah nilai atau dari sebuah operan. Adapun operator perbandingan dapat dilihat pada tabel 8.4 adalah sebagai berikut (Wibowo, 2017).

Tabel 8.4: Operator Perbandingan (Wibowo, 2017)

Operator	Deskripsi
==	Nilai sama dengan
===	Nilai dan tipe data sama dengan
!=	Tidak sama dengan
!==	Nilai dan tipe data tidak sama dengan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar dari atau sama dengan
<=	Lebih kecil dari atau sama dengan

Bab 9

Aliran Kontrol dan Perulangan

Aliran kontrol atau dapat juga disebut dengan kondisional serta perulangan dalam pemrograman adalah cara untuk menambahkan logika pada program yang kita buat (Flanagan, 2020). Hal ini memungkinkan kode program yang ditulis untuk membuat keputusan berdasarkan suatu kondisi tertentu. Kondisional memungkinkan kode tertentu dijalankan hanya jika kondisi-kondisi tertentu terpenuhi. Sementara itu, perulangan akan mengulangi bagian kode yang sama selama kondisi tertentu masih terpenuhi. Gabungan dari kondisional dan perulangan ini dikenal sebagai struktur kontrol karena memungkinkan kita untuk mengendalikan kapan dan seberapa sering kode program tertentu dijalankan (Morgan, 2022). Di bab ini, akan dibahas tentang kondisional menggunakan pernyataan `if` dan `switch`, serta cara mengulangi kode dengan pernyataan `while` dan `for`. Setiap pembahasan disertai dengan contoh kode yang memudahkan untuk memahami konsep dari kondisional dan perulangan di pemrograman JavaScript.

9.1 Pengambilan Keputusan

Pernyataan bersyarat atau kondisional adalah sekumpulan perintah yang dijalankan jika kondisi tertentu bernilai benar atau `TRUE` (MDN Contributors, 2019). Javascript mendukung dua pernyataan kondisional: `if..else` dan `switch`.

9.1.1 if

Kita dapat menggunakan pernyataan `if(..)` untuk mengeksekusi pernyataan jika kondisi logika bernilai benar atau `TRUE`. Struktur pernyataan `if` merupakan struktur yang paling sederhana karena hanya melibatkan satu buah ekspresi yang akan dipreiksa. Pernyataan `if(..)` mengevaluasi kondisi dalam tanda kurung dan, jika hasilnya benar (`TRUE`), maka blok kode yang dituliskan dalam tanda blok `{..}` akan dieksekusi. Bentuk umum dari pernyataan `if` adalah sebagai berikut:

```
if (kondisi) {  
    statement;  
}
```

Di sini, kondisinya dapat berupa ekspresi apa pun yang dievaluasi menjadi benar atau salah. Jika kondisi bernilai benar, `statement` maka `statement` yang berada dalam blok `{ }` akan dieksekusi. Jika tidak, maka program akan berjalan pada baris kode selanjutnya untuk dieksekusi. Contoh implementasi pernyataan `if` dapat dilihat pada Gambar 9.1 berikut. Dapat dilihat pada contoh program pada Gambar 9.1, bahwa sebuah variabel bernama `speed` memiliki nilai 80 (baris 1). Kemudian pada baris ke-5, dilakukan pengecekan kondisi menggunakan pernyataan `if`, yaitu apabila nilai dari variabel `speed` lebih dari (`>`) 40, maka pernyataan pada baris ke-6 akan ditampilkan ke layar. Namun, apabila hasil pengecekan menghasilkan nilai salah (`FALSE`), maka `statement` pada baris ke-6 akan diabaikan (dilewatkan) atau tidak dieksekusi.



```
1 let speed = 80;  
2  
3 console.log('Kecepatan Anda saat ini: ${speed}');  
4  
5 if(speed > 40) {  
6   console.log('Kurangi kecepatan kendaraan Anda.');7 }  
8
```

Terminal output:

```
code node if.js  
Kecepatan Anda saat ini: 80  
Kurangi kecepatan kendaraan Anda.
```

Gambar 9.1: Contoh kode program dengan pernyataan `if`

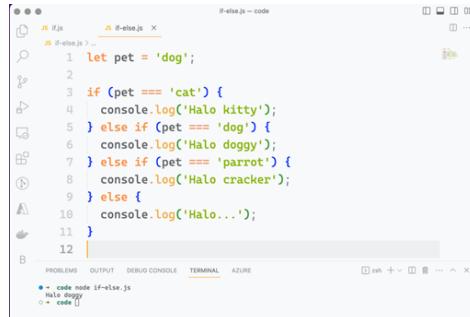
9.1.2 if..else

Pernyataan if juga dapat memiliki klausa else. Jika kondisi bernilai benar, pernyataan yang mengikuti kondisi tersebut dieksekusi. Jika kondisi tidak benar, pernyataan yang mengikuti kata else dieksekusi. Kedua pernyataan tersebut biasanya merupakan pernyataan majemuk (Matuszek, 2023).

Cara penulisan pernyataan if..else sebagai berikut:

```
if(kondisi) {  
    statement;  
} else {  
    statement;  
}
```

Contoh implementasi kode program untuk pernyataan if..else dapat dilihat pada Gambar 9.2. Pada contoh kode program dilakukan pengecekan untuk nilai dari variabel pet. Apabila pet memiliki nilai 'dog', maka pengecekan akan dilakukan mulai pada baris kode ke-3, karena hasil pengecekan bernilai FALSE (yaitu nilai dog tidak sama dengan cat), maka statement pada baris ke-4 tidak dieksekusi (dilewati). Kemudian pengecekan kembali dilakukan pada baris ke-6, dan saat pengecekan dilakukan menghasilkan nilai TRUE (karena kata dog memiliki nilai yang sama yaitu 'dog', yang merupakan kondisi yang akan dicek), sehingga statement pada baris ke-6 akan dieksekusi dan ditampilkan sebagai output dari program. Pada saat pengecekan menghasilkan nilai TRUE, maka kondisi lainnya (yaitu pada baris ke-8 tidak dieksekusi. Di baris ke-9 terdapat pernyataan else, pernyataan ini akan dieksekusi pada saat nilai dari variabel pet tidak menghasilkan nilai TRUE ketika dicek pada pernyataan if di baris ke-3, 5, dan 7. Apabila hal ini terjadi, maka statement "Halo.." akan dieksekusi dan ditampilkan sebagai output program.



```

1 let pet = 'dog';
2
3 if (pet === 'cat') {
4   console.log('Halo kitty');
5 } else if (pet === 'dog') {
6   console.log('Halo doggy');
7 } else if (pet === 'parrot') {
8   console.log('Halo cracker');
9 } else {
10  console.log('Halo...');
11 }
12

```

Terminal output: code node if-else.js, Halo doggy, == >>>

Gambar 9.2: Contoh kode program dengan pernyataan if..else

9.1.3 Ternary Operator: "?"

Pernyataan bersyarat juga dapat ditulis menggunakan operator ternary. Operator ternary mempersingkat pernyataan if..else menjadi satu ekspresi, yang sering kali berguna saat melakukan rendering bersyarat dengan ReactJS, Vue.js, atau Svelte (Minnick, 2023). Untuk menggunakan operator ternary, kondisi yang akan dicek ditulis sebelum tanda tanya (?), dan setelah tanda tanya dituliskan statement yang akan ditampilkan apabila hasil pengecekan kondisi bernilai TRUE, dan statement untuk pengecekan kondisi yang bernilai FALSE dituliskan setelah tanda titik dua. Sintaks penulisan dalam penggunaan operator ternary adalah sebagai berikut:

hasil = kondisi ? nilaiBenar: nilaiSalah



```

1 const jam = 10;
2 let pesan;
3
4 pesan = jam < 12 ? ('Selamat pagi!') : ('Selamat Datang');
5
6 console.log(pesan);
7

```

Terminal output: code node ternary.js, Selamat pagi!, == >>>

Gambar 9.3: Contoh kode program ternary operator

Gambar 9.3 memperlihatkan contoh implementasi kode program menggunakan operator ternary. Di gambar 9.3 operator ternary ditulis pada baris ke-4, di mana hasil pengecekan disimpan pada variabel pesan. Ketika dicek bahwa nilai variabel jam (pada baris ke-1 diberi nilai 10) kurang dari (<

12, jika bernilai TRUE maka string 'Selamat pagi!' akan menjadi nilai dari variabel pesan, sehingga ketika di baris ke-6 nilai dari variabel pesan ditampilkan akan menghasilkan output Selamat pagi!. Sedangkan, apabila hasil pengecekan kondisi oleh operator ternary pada baris ke-4 menghasilkan nilai FALSE, maka variabel pesan akan bernilai 'Selamat Datang', dan nilai inilah yang akan ditampilkan sebagai output program.

9.1.4 Switch

Pernyataan switch memungkinkan program untuk mengevaluasi sebuah ekspresi dan mencoba untuk mencocokkan nilai dari ekspresi tersebut dengan label case. Jika kecocokan ditemukan, program akan mengeksekusi pernyataan yang terkait.

JavaScript mengevaluasi pernyataan sakelar di atas sebagai berikut (MDN Contributors, 2019):

1. Program pertama-tama mencari klausa case dengan label yang sesuai dengan nilai ekspresi dan kemudian mentransfer kontrol ke klausa tersebut, mengeksekusi pernyataan terkait yang ada di dalam case tersebut.
2. Jika tidak ada label yang cocok ditemukan, program akan mencari klausa default:
 - a. Jika klausa default ditemukan, program akan mentransfer kontrol ke klausa tersebut dan mengeksekusi pernyataan terkait.
 - b. Jika tidak ada klausa default yang ditemukan, program akan melanjutkan eksekusi pada pernyataan yang mengikuti akhir switch.
 - c. Klausa default ditulis sebagai klausa terakhir, tetapi tidak harus demikian.

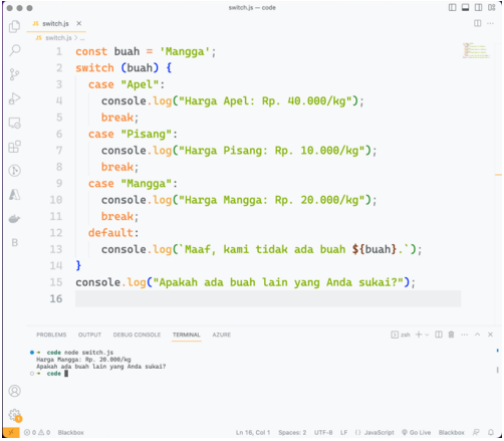
Sintaks dari pernyataan switch dapat dilihat pada Gambar 9.4 dan contoh implementasi kode program di Gambar 9.5.


```

1 switch(x) {
2   case 'value1': // if (x === 'value1')
3     ...
4     [break]
5
6   case 'value2': // if (x === 'value2')
7     ...
8     [break]
9
10  default:
11    ...
12    [break]
13 }

```

Gambar 9.4: Sintaks pernyataan switch (Javascript.info)



```

1 const buah = 'Mangga';
2 switch (buah) {
3   case "Apel":
4     console.log("Harga Apel: Rp. 40.000/kg");
5     break;
6   case "Pisang":
7     console.log("Harga Pisang: Rp. 10.000/kg");
8     break;
9   case "Mangga":
10    console.log("Harga Mangga: Rp. 20.000/kg");
11    break;
12  default:
13    console.log('Maaf, Kami tidak ada buah ${buah}.');
14 }
15 console.log("Apakah ada buah lain yang Anda sukai?");
16

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ACTIVE

```

• code node switch.js
  Harga Mangga: Rp. 20.000/kg
  Apakah ada buah lain yang Anda sukai?
• code

```

Ln 16, Col 1 Spaces: 2 UTF-8 LF JavaScript Go Live Blackbox

Gambar 9.5: Contoh kode program switch

Pada saat kode program di Gambar 9.5 dijalankan, nilai dari variabel buah (nilainya adalah Mangga) akan dievaluasi sesuai label yang ada di statement case (baris ke-2, 6, dan 9). Ketika ada label case yang bernilai TRUE atau sama ketika dievaluasi (pada contoh program diatas adalah case Mangga (baris ke-9), maka statement yang ada di dalam case tersebut akan dieksekusi dan program akan menampilkan output: Harga Mangga: Rp. 20.000/kg. Apabila tidak ada label case yang bernilai sama, maka statement yang ada pada klausa default yang akan dieksekusi sehingga program akan mengeluarkan output: Maaf, kami tidak ada buah.

9.2 Pengulangan

Perulangan adalah bentuk lain dari struktur kontrol dalam JavaScript yang memungkinkan kita menjalankan kode yang sama beberapa kali (Morgan, 2022). Ada lima jenis perulangan utama yang akan dibahas dalam sub-bab ini yaitu: `while`, `do..while`, `for`, `for..of`, dan `for..in`.

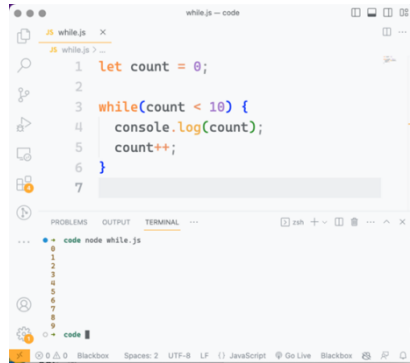
9.2.1 `while`

Perulangan `while` adalah jenis perulangan yang paling sederhana; pada dasarnya, perulangan `while` akan bekerja menjalankan kode yang sama selama beberapa kondisi bernilai benar. Dalam hal ini, perulangan ini mirip dengan pernyataan `if`, perbedaannya adalah perulangan `while` akan terus menjalankan kode selama kondisinya benar, sedangkan pernyataan `if` hanya akan menjalankannya paling banyak satu kali. Sering kali kita perlu menulis kode yang dijalankan beberapa kali, bukan hanya sekali.

Pada perulangan `while`, kondisi akan dicek pada bagian awal, sintaksnya seperti berikut:

```
while(kondisi) {  
    statement;  
}
```

Pertama, kondisi diuji; jika kondisi tersebut salah, maka tidak ada yang akan dilakukan, program akan berjalan ke baris berikutnya dan tidak menjalankan `statement` yang berada di dalam blok `while` (ditandai dengan `{.. }`). Sedangkan, jika kondisinya bernilai benar, maka pernyataan dalam blok `while` akan dijalankan, lalu keseluruhan perulangan (dimulai dengan pengujian ulang) dieksekusi kembali. Berikut ini adalah contoh perulangan `while` yang mencetak angka dari 0 hingga 9:

A screenshot of a code editor window titled 'while.js -- code'. The editor shows a JavaScript file named 'while.js' with the following code:

```
1 let count = 0;  
2  
3 while(count < 10) {  
4   console.log(count);  
5   count++;  
6 }  
7
```

The editor interface includes a search bar, a toolbar with icons for undo, redo, and other actions, and a bottom panel with tabs for 'PROBLEMS', 'OUTPUT', and 'TERMINAL'. The 'TERMINAL' tab is active, showing the command 'code node while.js' and its output: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'. The status bar at the bottom indicates 'Blackbox', 'Spaces: 2', 'UTF-8', 'LF', 'JavaScript', and 'Go Live'.

Gambar 9.6: Contoh implementasi perulangan while

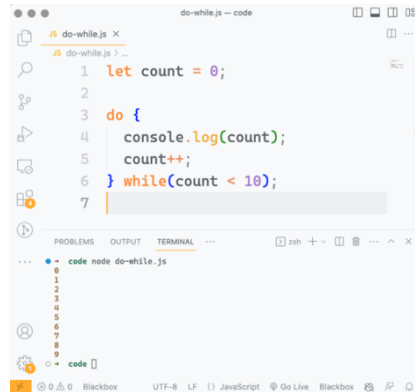
Seperti yang dapat Anda lihat, variabel `count` dimulai dari 0 dan bertambah setiap kali perulangan berjalan. Setelah perulangan dijalankan 10 kali, ekspresi atau pengecekan kondisi menjadi salah (yaitu, variabel `count` tidak lagi kurang dari 10), pernyataan `while` selesai, dan interpreter JavaScript melanjutkan ke pernyataan berikutnya dalam program.

9.2.2 `do..while`

Perulangan `do/while` sama seperti perulangan `while`, kecuali kondisi perulangan diuji di bagian akhir perulangan, bukan di awal. Ini berarti bahwa statement yang ada di bagian perulangan selalu dieksekusi setidaknya sekali. Sintaksnya adalah:

```
do {  
    statement;  
} while (kondisi);
```

Menggunakan contoh yang sebelumnya, apabila menggunakan perulangan `do..while`, maka kode program tersebut akan ditulis menjadi seperti berikut:



```
do-while.js - code
do-while.js
1 let count = 0;
2
3 do {
4   console.log(count);
5   count++;
6 } while(count < 10);
7

PROBLEMS OUTPUT TERMINAL ...
code node do-while.js
0
1
2
3
4
5
6
7
8
9
```

Gambar 9.7: Contoh implementasi perulangan do..while

Dibaris pertama dilakukan inisialisasi variabel bernama count yang diberi nilai awal 0. Kemudian perulangan dilakukan mulai pada baris ketiga, sehingga program akan memberikan output 0. Setelah itu pada baris kelima, nilai variabel count ditambahkan 1 (count++), sehingga nilainya menjadi 1. Pada baris keenam, dilakukan pengecekan apakah nilai variabel count (pada perulangan pertama ini, sekarang bernilai 1) kurang dari 10? Apabila menghasilkan nilai true (yang pada kondisi ini bernilai true), akan program akan melanjutkan untuk mengeksekusi kode baris keempat dan kelima. Ini terus dilakukan sampai pengecekan kondisi dibaris keenam menghasilkan nilai false (salah). Contohnya pada iterasi ke-10, di mana nilai dari variabel count adalah 10, sehingga pengecekan kondisi 10 kurang dari 10 menghasilkan nilai salah. Oleh karena nilai 10 sama dengan 10, bukan kurang dari 10.

9.2.3 for

Perulangan for adalah jenis perulangan lainnya. Seperti perulangan while, perulangan for terus mengulang selama kondisi bernilai benar. Pernyataan for menyediakan konstruksi perulangan yang sering kali lebih nyaman daripada pernyataan while. Pernyataan for menyederhanakan perulangan yang mengikuti pola umum.

Pernyataan for menggunakan tiga ekspresi dalam tanda kurung setelah kata kunci for untuk membuat perulangan (Minnick, 2023):

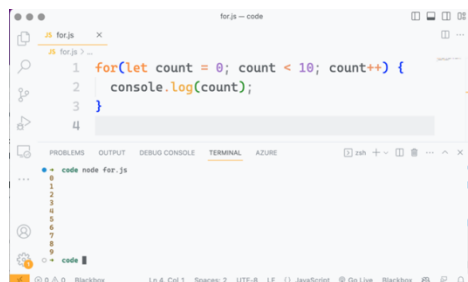
1. Inisialisasi: Nilai awal dari sebuah variabel. Variabel ini biasanya berupa penghitung.

2. Kondisi: Ekspresi Boolean yang menentukan apakah akan menjalankan pernyataan perulangan.
3. Ekspresi akhir: Ekspresi yang akan dievaluasi setelah setiap iterasi perulangan. Ekspresi ini biasanya digunakan untuk menambah penghitung.

Berikut sintaks dari perulangan for:

```
for(inisialisasi; kondisi; ekspresi) {  
    statement;  
}
```

Gambar 9.8 memperlihatkan contoh program yang sama di pembahasan perulangan while dan do..while yang dituliskan dengan perulangan for. Dapat dilihat pada baris ke-1, dibuat inisialisasi variabel bernama count dengan nilai awal 0. Kemudian pengecekan kondisi pada statement `count < 10`, apabila menghasilkan nilai true, maka kode program pada baris kedua akan dieksekusi, kemudian iterator akan menambahkan nilai dari variabel count dengan 1 (`count++`). Begitu seterusnya dilakukan sampai pengecekan kondisi menghasilkan nilai false, baru program akan berhenti.



Gambar 9.8: Contoh implementasi perulangan for

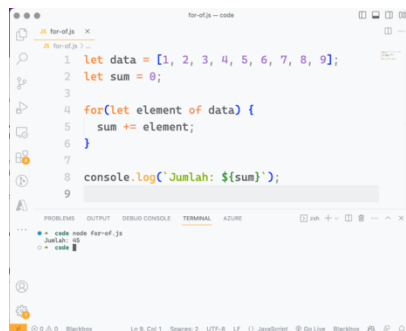
9.2.4 for..of

Dalam spesifikasi JavaScript ES6, didefinisikan pernyataan perulangan baru yaitu `for..of`. Jenis perulangan baru ini menggunakan kata kunci `for` tetapi merupakan jenis perulangan yang sama sekali berbeda dari perulangan `for` biasa. Perulangan `for..of` bekerja dengan objek yang dapat diulang. Contoh objek yang dapat diulang antara lain array, string, set, dan map (objek ini

merepresentasikan urutan atau kumpulan elemen yang dapat Anda ulangi atau lakukan iterasi dengan menggunakan perulangan `for..of`). Sintaks perulangan `for..of` sebagai berikut:

```
for(variabel of data) {  
    statement;  
}
```

Sepintas sintaksnya terlihat seperti perulangan `for` biasa, kata kunci `for` diikuti dengan tanda kurung yang berisi detail tentang apa yang harus dilakukan perulangan. Dalam hal ini, tanda kurung berisi deklarasi variabel (atau, untuk variabel yang sudah dideklarasikan, cukup nama variabel) diikuti dengan kata kunci `of` dan ekspresi yang dievaluasi ke objek yang dapat diulang, seperti array. Contoh kode program yang `for..of` dapat dilihat pada Gambar 9.9. Program ini melakukan perulangan dengan `for..of` pada sebuah array yang berisi nilai 1 sampai 9, kemudian menampung nilai data dalam array tersebut untuk dihitung jumlahnya (menggunakan variabel `sum`).



Gambar 9.9: Contoh Program Menggunakan Perulangan `for..of` pada sebuah array

Pada contoh kode program di Gambar 9.9, pada baris pertama dan kedua, di deklarasi dua buah variabel, yaitu variabel array bernama `data` yang memiliki 9 nilai (dari 1-9) dan variabel bernama `sum` dengan inisial nilai 0. Kemudian pada baris keempat, perulangan dilakukan dengan mendeklarasikan variabel bernama `element` (`statement let element`) untuk melakukan iterasi ke dalam isi dari array `data`. Dengan variabel `element` ini maka program dapat mengakses isi dari sebuah objek (dalam contoh ini adalah sebuah array). Kemudian di baris kelima isi dari objek array yang diiterasi ditambahkan nilainya ke dalam variabel `sum`. Hal ini terus berulang sampai isi dari array

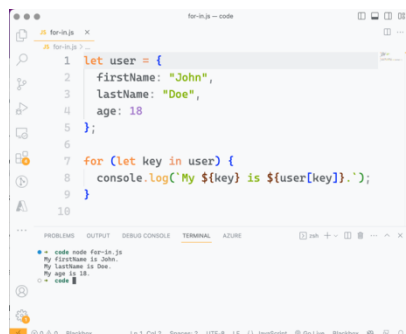
data telah selesai diiterasi (jika ada 10 data, maka hal ini diulangi sebanyak 10 kali). Baris kedelapan menampilkan hasil dari penjumlahan semua element dalam array data yang telah diiterasi dan menghasilkan output yaitu 45.

9.2.5 for..in

Perulangan for/in sangat mirip dengan perulangan for/of, dengan kata kunci of diubah menjadi in. Jika perulangan for/of membutuhkan objek yang dapat diulang setelah of, perulangan for/in dapat digunakan untuk semua objek setelah in. Perulangan for/of baru diperkenalkan di ES6, namun for/in telah menjadi bagian dari JavaScript sejak awal. Pernyataan for/in mengulang nama properti dari objek tertentu. Sintaks ini terlihat seperti ini:

```
for (variabel in objek) {  
    statement;  
}
```

Perulangan for..in mengulang properti sebuah objek dan properti yang diwarisi dari objek induknya. Karena array adalah salah jenis objek, maka for..in juga dapat digunakan untuk mengulang elemen-elemen dalam sebuah array. Contoh kode program perulangan for..in dapat dilihat pada Gambar 9.10.



Gambar 9.10: Contoh program dengan perulangan for..in

Pada contoh program di Gambar 9.10, perulangan for..in melakukan iterasi pada property dari sebuah objek dengan nama user. Pada baris 1-5 dilakukan deklarasi sebuah objek bernama user. Objek user memiliki tiga properti, yaitu firstName dengan nilai John, lastName dengan nilai Doe, dan age yang memiliki nilai 18. Kemudian pada baris ketujuh, untuk melakukan iterasi yang dapat mengakses nilai pada objek user, digunakanlah perulangan for..in.

Pertama-tama dibuat variabel bernama `key` (bisa juga menggunakan nama yang lain) melalui statement `let key`. Disusul dengan keyword `in` dan nama objek yang akan diiterasi yaitu `user`. Sehingga dengan penulisan kode lengkap pada baris ketujuh, program memiliki akses penuh ke property dan nilainya pada sebuah objek. Kemudian dibaris ke-delapan program menampilkan output berupa property objek beserta nilainya.

Bab 10

Fungsi dan Objek

10.1 Pengertian dan Pembuatan Fungsi

10.1.1 Mengenal Konsep Fungsi dalam Pemrograman

Konsep fungsi dalam pemrograman adalah cara untuk mengelompokkan serangkaian pernyataan atau instruksi menjadi sebuah blok kode yang dapat dipanggil dan dieksekusi secara terpisah. Fungsi memungkinkan pemrogram untuk memecah masalah besar menjadi bagian-bagian yang lebih kecil dan lebih terkelola. Ini adalah konsep yang mendasar dalam pemrograman dan digunakan di hampir semua bahasa pemrograman (Haverbeke, 2018).

Berikut adalah beberapa konsep penting yang terkait dengan fungsi dalam pemrograman:

1. Abstraksi

Fungsi memungkinkan Anda untuk mengisolasi detail-detail kompleks dari bagian lain dari kode. Ini membantu dalam memahami kode secara lebih umum tanpa harus terjebak dalam detail implementasi.

2. Penggunaan Ulang Kode

Dengan mendefinisikan fungsi, Anda dapat menulis kode yang dapat digunakan di berbagai bagian program Anda tanpa perlu menyalin dan menempelkan instruksi yang sama berulang-ulang.

3. Pemisahan Tanggung Jawab

Dengan memisahkan tugas-tugas ke dalam fungsi-fungsi terpisah, Anda dapat mengatur kode menjadi bagian-bagian yang lebih spesifik, mempermudah pemeliharaan dan pengembangan.

4. Pengurangan Duplikasi

Fungsi membantu mengurangi duplikasi kode, karena Anda hanya perlu merancang dan mengimplementasikan logika tertentu sekali, dan kemudian Anda dapat memanggilnya berulang kali.

5. Pengorganisasian

Fungsi membantu dalam mengatur logika dan alur kerja program dengan cara yang lebih terstruktur dan logis.

6. Parameter dan Argumen

Fungsi dapat menerima input melalui parameter. Parameter adalah variabel yang didefinisikan dalam tanda kurung saat mendefinisikan fungsi. Nilai-nilai ini disebut argumen saat fungsi dipanggil.

7. Nilai Kembalian

Fungsi dapat mengembalikan hasil menggunakan pernyataan `return`. Nilai ini dapat digunakan di tempat pemanggilan fungsi.

8. Fungsi Rekursif

Konsep rekursi melibatkan pemanggilan fungsi itu sendiri. Ini dapat digunakan untuk memecahkan masalah yang dapat dibagi menjadi kasus yang lebih kecil dan serupa.

9. Fungsi Anonim

Fungsi yang tidak memiliki nama disebut fungsi anonim. Ini sering digunakan dalam situasi seperti `callback` dalam pemrograman asinkron.

10. Skop

Setiap fungsi memiliki skop atau cakupan, yang menentukan di mana variabel dapat diakses. Variabel yang dideklarasikan dalam fungsi

hanya dapat diakses di dalam fungsi tersebut, kecuali jika variabel tersebut dinyatakan sebagai variabel global.

10.1.2 Mendeklarasikan Fungsi

Fungsi dideklarasikan dengan kata kunci `function`, diikuti oleh nama fungsi dan tanda kurung `()`. Isi dari fungsi ditempatkan di dalam blok kurung kurawal `{}`.

Contoh:

```
function greeting() {  
    console.log("Hello, world!");  
}
```

Fungsi dapat dipanggil dengan menggunakan nama fungsi diikuti oleh tanda kurung `()`. Pemanggilan ini akan menjalankan kode yang ada di dalam fungsi.

Contoh:

```
greeting();//Memanggil fungsi greeting
```

Fungsi dapat menerima nilai yang disebut argumen melalui tanda kurung saat dipanggil. Parameter adalah variabel yang digunakan dalam definisi fungsi untuk menerima argumen tersebut.

Contoh:

```
function greet(name) {  
    console.log("Hello, " + name + "!");  
}  
  
greet("Alice");//Output: Hello, Alice!
```

Fungsi dapat mengembalikan nilai menggunakan kata kunci `return`. Nilai ini dapat digunakan di tempat pemanggilan fungsi.

Contoh:

```
function add(a, b) {  
    return a + b;  
}
```

```
let result = add(5, 3);  
console.log(result);//Output: 8
```

10.2 Menggunakan Fungsi Bawaan JavaScript

10.2.1 Menggunakan Fungsi Math

Dalam JavaScript, objek `Math` adalah objek bawaan yang menyediakan berbagai fungsi matematika yang bermanfaat. Fungsi-fungsi ini dapat digunakan untuk melakukan operasi matematika yang umum, seperti menghitung akar kuadrat, menghasilkan angka acak, menghitung nilai absolut, dan lain sebagainya (Flanagan, 2020).

Berikut adalah beberapa contoh penggunaan fungsi dari objek `Math`:

1. `Math.abs()`: Mengembalikan nilai absolut dari suatu angka.
Contoh:

```
let num = -10;  
let absValue = Math.abs(num);  
console.log(absValue);//Output: 10
```
2. `Math.sqrt()`: Mengembalikan akar kuadrat dari suatu angka.
Contoh:

```
let number = 25;  
let squareRoot = Math.sqrt(number);  
console.log(squareRoot);//Output: 5
```
3. `Math.random()`: Menghasilkan angka acak antara 0 (inklusif) dan 1 (eksklusif).
Contoh:

```
let randomNum = Math.random();  
console.log(randomNum);//Output: Angka acak antara 0 dan 1
```
4. `Math.floor()` dan `Math.ceil()`: Mengubah angka menjadi bilangan bulat terdekat yang lebih rendah (floor) atau lebih tinggi (ceil).

Contoh:

```
let floatNum = 9.99;
let flooredNum = Math.floor(floatNum);
let ceiledNum = Math.ceil(floatNum);
console.log(flooredNum); //Output: 9
console.log(ceiledNum); //Output: 10
```

5. `Math.round()`: Mengembalikan pembulatan angka ke bilangan bulat terdekat.

Contoh:

```
let decimalNum = 7.5;
let roundedNum = Math.round(decimalNum);
console.log(roundedNum); //Output: 8
```

6. `Math.max()` dan `Math.min()`: Mengembalikan nilai maksimum atau minimum dari sejumlah angka yang diberikan sebagai argumen.

Contoh:

```
let maxValue = Math.max(5, 10, 2, 8);
let minValue = Math.min(5, 10, 2, 8);
console.log(maxValue); //Output: 10
console.log(minValue); //Output: 2
```

7. `Math.pow()`: Menghitung pangkat dari suatu angka.

Contoh:

```
let base = 2;
let exponent = 3;
let result = Math.pow(base, exponent);
console.log(result); //Output: 8 (2^3)
```

8. `Math.PI`: Mengembalikan nilai π (pi).

Contoh:

```
let circleArea = Math.PI * Math.pow(radius, 2);
```

10.2.2 Menggunakan Fungsi String

Fungsi string dalam JavaScript adalah metode yang dapat digunakan pada tipe data string. Metode-metode ini memungkinkan Anda untuk melakukan operasi atau manipulasi pada string, seperti mengubah huruf besar/kecil, memotong, mencari substring, dan banyak lagi.

Berikut adalah beberapa contoh penggunaan fungsi string di JavaScript (Flanagan, 2020):

1. `length`: Mengembalikan panjang dari string.

Contoh:

```
let text = "Hello, world!";
let length = text.length;
console.log(length); //Output: 13
```

2. `toUpperCase()` dan `toLowerCase()`: Mengubah semua karakter dalam string menjadi huruf besar atau huruf kecil.

Contoh:

```
let message = "Hello, World!";
let upperCaseMsg = message.toUpperCase();
let lowerCaseMsg = message.toLowerCase();
console.log(upperCaseMsg); //Output: "HELLO, WORLD!"
console.log(lowerCaseMsg); //Output: "hello, world!"
```

3. `charAt()`: Mengembalikan karakter pada posisi tertentu dalam string.

Contoh:

```
let word = "JavaScript";
let char = word.charAt(4);
console.log(char); //Output: "S"
```

4. `substring()` dan `slice()`: Mengambil potongan substring dari string.

Contoh:

```
let text = "Hello, world!";
let subText1 = text.substring(7, 12);
let subText2 = text.slice(0, 5);
console.log(subText1); //Output: "world"
console.log(subText2); //Output: "Hello"
```

5. `indexOf()` dan `lastIndexOf()`: Mencari indeks pertama atau terakhir dari substring dalam string.

Contoh:

```
let sentence = "I love programming, programming is fun!";
let firstIndex = sentence.indexOf("programming");
let lastIndex = sentence.lastIndexOf("programming");
console.log(firstIndex); //Output: 2
```

- ```
console.log(lastIndex);//Output: 22
```
6. `replace()`: Menggantikan substring tertentu dengan substring lain dalam string.  
Contoh:

```
let message = "Hello, world!";
let newMessage = message.replace("world", "Universe");
console.log(newMessage);//Output: "Hello, Universe!"
```
  7. `trim()`: Menghapus karakter-karakter spasi ekstra di awal dan akhir string.  
Contoh:

```
let paddedString = " Hello ";
let trimmedString = paddedString.trim();
console.log(trimmedString);//Output: "Hello"
```
  8. `split()`: Membagi string menjadi array berdasarkan pemisah tertentu.  
Contoh:

```
let csv = "apple,banana,orange";
let fruitsArray = csv.split(",");
console.log(fruitsArray);//Output: ["apple", "banana", "orange"]
```
  9. `concat()`: Menggabungkan beberapa string menjadi satu string baru.  
Contoh:

```
let firstName = "John";
let lastName = "Doe";
let fullName = firstName.concat(" ", lastName);
console.log(fullName);//Output: "John Doe"
```

### 10.2.3 Menggunakan Fungsi Date

Fungsi `Date` dalam JavaScript adalah objek bawaan yang digunakan untuk bekerja dengan tanggal dan waktu. Objek `Date` memungkinkan Anda untuk membuat, mengelola, dan memanipulasi tanggal, waktu, dan zona waktu. Anda dapat membuat objek `Date` baru dengan memanggil konstruktor `Date()` (Flanagan, 2020).



Contoh:

```
let currentDate = new Date();//Membuat objek Date saat ini
console.log(currentDate);//Output: Wed Aug 18 2023 15:30:00
GMT+0000 (Coordinated Universal Time)
```

Untuk Mendapatkan Komponen Tanggal dan Waktu, anda dapat menggunakan metode-metode seperti `getFullYear()`, `getMonth()`, `getDate()`, `getHours()`, `getMinutes()`, dan sejenisnya untuk mendapatkan komponen tanggal dan waktu dari objek `Date`.

Contoh:

```
let now = new Date();
let year = now.getFullYear();
let month = now.getMonth();
let day = now.getDate();
let hours = now.getHours();
let minutes = now.getMinutes();
console.log(year, month, day, hours, minutes);
//Output: Tahun, Bulan (0-11), Hari (1-31), Jam (0-23), Menit
(0-59)
```

Untuk mengatur tanggal dan waktu. anda dapat menggunakan metode-metode seperti `setFullYear()`, `setMonth()`, `setDate()`, `setHours()`, `setMinutes()`, dan sejenisnya untuk mengatur komponen tanggal dan waktu pada objek `Date`.

Contoh:

```
let birthday = new Date();
birthday.setFullYear(1990);
birthday.setMonth(5);
birthday.setDate(15);
console.log(birthday);//Output: Sat Jun 15 1990 15:30:00
GMT+0000 (Coordinated Universal Time)
```

Objek `Date` sendiri tidak memiliki metode bawaan untuk format tanggal. Namun, Anda bisa membuat fungsi sendiri atau menggunakan pustaka pihak ketiga untuk melakukan pemformatan tanggal yang diinginkan.

Contoh (menggunakan pustaka `date-fns`):

```
const dateFns = require('date-fns');
let currentDate = new Date();
let formattedDate = dateFns.format(currentDate, 'yyyy-MM-dd
HH:mm:ss');
console.log(formattedDate); //Output: "2023-08-18 15:30:00"
```

Anda dapat melakukan operasi matematika pada objek `Date` untuk menghitung selisih waktu, menambah atau mengurangi waktu, dan lain sebagainya.

Contoh:

```
let now = new Date();
let tomorrow = new Date(now);
tomorrow.setDate(now.getDate() + 1);
console.log(tomorrow); //Output: Thu Aug 19 2023 15:30:00
GMT+0000 (Coordinated Universal Time)
```

Objek `Date` memiliki banyak metode lain yang dapat Anda eksplorasi dalam dokumentasi JavaScript. Ingatlah bahwa pengelolaan waktu bisa menjadi kompleks, terutama ketika melibatkan zona waktu dan perhitungan yang akurat. Pustaka-pustaka pihak ketiga, seperti `date-fns`, `moment.js`, atau bahkan JavaScript modern dengan fitur-fitur baru seperti `Intl.DateTimeFormat`, dapat membantu dalam memanipulasi dan memformat tanggal dengan lebih efektif.

## 10.3 Objek Dalam JavaScript

### 10.3.1 Konsep Object Dalam JavaScript

Dalam JavaScript, konsep objek adalah salah satu aspek paling fundamental dan kuat. Objek memungkinkan Anda untuk mengelompokkan nilai-nilai terkait (properti) dan tindakan (metode) menjadi sebuah entitas tunggal. Objek

digunakan untuk merepresentasikan berbagai entitas dunia nyata dan konsep di dalam kode Anda, dan mereka membantu dalam mengorganisasi dan mengelola data dengan lebih terstruktur (Osmani, 2023).

Objek bisa digambarkan sebagai sesuatu yang memiliki properti dan nilai. Sebagai contoh, manusia. Manusia memiliki nama, pekerjaan, umur, dan lain sebagainya. Kemudian kita tahu bahwa nama orang ini adalah John, pekerjaannya adalah programmer, umurnya 24 tahun, dan lain-lain. Dalam hal ini, manusia adalah objek. Kemudian nama, pekerjaan, umur adalah properti objek. Terakhir, John adalah nilai dari properti nama, programmer adalah nilai dari properti pekerjaan, dan 24 tahun adalah nilai dari properti umur. Objek JavaScript dapat menampung properti dan nilai sebanyak apapun dengan tipe data yang beragam.

Berikut adalah beberapa poin penting tentang konsep objek dalam JavaScript:

1. Properti dan Metode

Objek dapat memiliki properti, yang adalah pasangan nama-nilai yang mewakili atribut dari objek tersebut. Objek juga dapat memiliki metode, yang adalah fungsi yang terkait dengan objek tersebut dan dapat dipanggil untuk melakukan tindakan tertentu.

2. Notasi Literal

Objek dapat dibuat dengan menggunakan notasi literal, yaitu dengan mengelompokkan pasangan nama-nilai di dalam tanda kurung kurawal {}.

3. Akses Properti dan Metode

Properti dan metode dalam objek dapat diakses dengan menggunakan operator titik (.) atau operator kurung siku ([]).

### 10.3.2 Membuat Object Dengan Object Literal dan Constructor

Cara mendeklarasikan objek dengan object literal adalah sebagai berikut:

```
let namaObjek = {
 namaProperti1: nilai1,
 namaProperti2: nilai2
};
```

Contoh:

```
let orang = {
 nama: 'john',
 umur: 24,
 pekerjaan: 'programmer'
};
```

Objek berciri khas datanya dibungkus dengan kurung kurawal {}, dan antar data diberi koma,.

Objek juga dapat dibuat menggunakan fungsi konstruktor, seperti `Object()`, `Array()`, atau konstruktor khusus yang Anda buat sendiri.

Contoh:

```
let namaObjek = new Object();
namaObjek.namaProperti1 = nilai1;
namaObjek.namaProperti2 = nilai2;
```

Contoh:

```
let orang = new Object();
orang.nama = 'john';
orang.umur = 24;
orang.pekerjaan = 'programmer';
```

## 10.4 Menerapkan Fungsi pada Object

### 10.4.1 Membuat Metode Pada Object

Metode pada objek dalam JavaScript adalah fungsi yang terkait dengan objek dan dapat dipanggil untuk melakukan tindakan tertentu terhadap objek tersebut. Metode memungkinkan Anda untuk menjalankan operasi atau manipulasi data yang terkait dengan objek, dan sering digunakan untuk menjaga kohesi dan enkapsulasi dalam kode Anda. Metode dapat ditambahkan ke objek saat objek dibuat atau setelahnya. Metode didefinisikan dalam objek

dengan cara menyatakan fungsi di dalam objek menggunakan sintaks `namaMetode: function() {.. }` (Freeman dan Robson, 2014).

Contoh:

```
let person = {
 firstName: "John",
 lastName: "Doe",
 sayHello: function() {
 console.log("Hello World");
 }
};
```

### 10.4.2 Penggunaan "this" Dalam Metode Object

Kata kunci `this` mengacu pada objek yang sedang memanggil metode tersebut. Ini memungkinkan Anda untuk mengakses properti dan metode dari objek itu sendiri.

Contoh:

```
let person = {
 firstName: "John",
 lastName: "Doe",
 sayHello: function() {
 console.log("Hello, " + this.firstName + "!");
 }
};
```

# Bab 11

## Pengenalan Bootstrap

### 11.1 Pendahuluan

Teknologi Web merupakan teknologi informasi yang berkembang sangat dinamis berdasarkan teknologi hardware dan software yang memiliki bahasa atau kode dalam berkomunikasi antara komputer satu dengan yang lainnya dan juga menyesuaikan dengan perkembangan dari peminat yang menggunakan teknologi tersebut. Aplikasi mobile merupakan salah satu perkembangan pada sisi teknologi hardware, sedangkan pada sisi software adalah perkembangan teknologi statis menuju web dinamis yaitu big data, service-oriented, dan cloud computing. Selain dari sisi hardware maupun software, terdapat juga dari sisi pengguna di mana dalam mengakses sebuah informasi mengarah pada multiple-gadget yaitu dapat mengakses seluruh informasi dari perangkat yang berbeda seperti tablet, smartphone dan desktop. Istilah dari kegiatan yang digunakan pengguna untuk memperoleh informasi dari perangkat yang berbeda dikenal dengan istilah perancangan web responsif (responsif web design atau RWD) (Adri, 2018).

Desain antar muka (User Interface –UI) merupakan cara program dan pengguna saling berinteraksi untuk memenuhi kebutuhan dari responsive web design. Antarmuka pengguna memberikan tampilan grafis antar komputer dan perangkat lunak yang dapat didengar, dilihat, disentuh dan dimengerti oleh manusia. Adapun contoh dari penerapan UI adalah *Automated Teller Machine*

(ATM), mouse komputer, remote control dan lain sebagainya (Wiwesa, 2021). Namun seiring perkembangan teknologi maka terjadi pergeseran akses dari content-oriented menjadi user-centered information menuju pada aplikasi terhadap pengalaman pengguna dalam berinteraksi dengan sistem yang berbasis web (user experience –UX).

Dalam mencapai kebutuhan dari UI dan UX maka dibutuhkan HTML dan CSS sebagai tata letak dari bagian elemen web dan interaksi pengguna dengan menggunakan JavaScript. Penggunaan elemen tersebut mengakibatkan pembebanan akses informasi dari sisi pengguna terhadap web browser. Solusi terhadap pembebanan akses informasi dapat dilakukan dengan pendekatan terhadap object-oriented web design. Salah satu framework yang bersifat responsive dan merupakan desain front end yang populer dikarenakan desain webnya dapat menyesuaikan tampilan webnya atau ukuran layar secara otomatis adalah Bootstrap (Josi, 2016).

Bootstrap merupakan framework yang memberikan kemudahan bagi penggunaannya dengan menggunakan beberapa bantuan interface berbasis CSS, HTML dan JavaScript sebagai kebutuhan dalam pengembangan tombol, tipografi, formulis, dan komponen antarmuka lainnya (Homaidi, 2016) (Nurmalasari, 2019). Selain itu framework bootstrap digunakan untuk menyusun program aplikasi yang bersifat client side dan front end (Riasinir, 2019).



**Gambar 11.1:** Logo Bootstrap 5 (<https://icons.getbootstrap.com>)

## 11.2 Sejarah Singkat Bootstrap

Pencipta dari framework bootstrap dibuat oleh programmer di Twitter pada tahun 2011, yang bernama Mark Otto dan Jacob Thornton. Lahirnya aplikasi ini dikarenakan Mark Otto dan Jacob Thornton berinisiatif untuk membuat

sebuah tools yang dapat digunakan untuk memudahkan pekerjaannya pada lingkungan internal Twitter dikarenakan pada ruang lingkup kerja mereka tidak adanya standarisasi yang menyulitkan kegiatannya tersebut (Alatas, 2013).

Bootstrap menjadi framework front-end yang paling banyak digunakan di publik dengan berevolusinya menjadi berbasis CSS yang menjadi framework yang lengkap dikarenakan berisikan JavaScript Plugin, Button, Icon dan Form sejak diluncurkannya pada bulan Agustus 2011 (Hidayah, 2018). Bootstrap versi 2.0 telah dirilis pada 31 Januari 2012, di mana fitur yang tersedia pada versi tersebut adalah telah dimasukkannya 12 kolom responsive grid system serta penambahan komponen baru lainnya sehingga peminat dari versi 2.0 mencapai 20.000 kali copy pada proyek Github. Pada 19 Agustus 2013 Bootstrap versi 3.0 telah rilis dengan penambahan beberapa fitur yaitu dengan mengadopsi flat design dan telah menggunakan konsep mobile-first.

Pada 18 Januari 2018, Bootstrap versi 4.0 telah hadir dengan menghadirkan konsep flexbox dalam layoutnya di mana sebelum menggunakan Flexbox, Bootstrap versi sebelumnya menggunakan sistem float, dan menggunakan Sass di mana sebelumnya menggunakan CSS preprocessor internal dari Less. Rilis terbaru pada 5 Mei 2021 yaitu merupakan Bootstrap 5 setelah melewati versi alpha dan beta sepanjang tahun 2020 yang diharapkan perkembangan versi Bootstrap dapat mempermudah dalam hal kinerja website lebih ringan dengan menghilangkannya jQuery yang digantikan dengan Vanilla JS.



**Gambar 11.2:** Perkembangan Framework Bootstrap



## 11.3 Kelebihan dan Kekurangan Bootstrap

Pada penjelasan terdahulu kita telah mengetahui pengertian dan sejarah dari Framework Bootstrap maka selanjutnya, yang perlu diketahui adalah kelebihan dan kekurangan dari Bootstrap.

Adapun beberapa kelebihan dari Bootstrap adalah sebagai berikut (Lubis, 2022) (Putra, 2020).

1. Mudah Digunakan

Tampilan website yang tersedia dari framework ini, lebih mudah untuk digunakan karena memudahkan pengguna dalam hal struktur atau kerangka kerja pada proses pengembangan tampilan website dan juga proses maintainancenya lebih terstruktur.

2. Responsive Grid

Adanya sistem grid yang terdiri dari baris dan kolom memudahkan programmer dalam menggunakan Framework ini dikarenakan tidak perlu membuatnya dari nol, memudahkan dalam membuat grid baru tanpa merubah grid yang telah ada. Sistem ini mempermudah dalam melakukan entri data. Pada sistem grid terdapat dua kelas berbasis desktop dan seluler yaitu fixed container (.container) merupakan layout paling dasar atau sebagai wadah yang menampung seluruh elemen grid dan ukuran website lebar namun layoutnya tidak penuh dan fluid container (.container-fluid) merupakan kebalikan dari fixed container yaitu tampilan websitenya dan layoutnya penuh.

3. Kompatibel dengan beberapa Browser

Dapat diakses oleh berbagai macam browser, sehingga website akan memberikan kemudahan bagi pengguna aplikasinya. Penggunaan website dengan Framework Bootstrap akan memudahkan pengguna tanpa ada masalah dalam proses pemakaian aplikasi.

4. Bootstrap image system

Dapat mengubah ukuran layar pengguna secara otomatis dengan menambahkan kelas .img-responsive, tidak hanya kelas dalam

membuat gambar secara otomatis namun Bootstrap juga menyediakan kelas untuk membantu memodifikasi bentuk gambar yaitu `.img-circle` dan `.img-rounded`.

#### 5. Dokumentasi Bootstrap yang lengkap

Pada fasilitas dokumentasi yang tersedia bagi developer pemula pada framework Bootstrap tersedia beberapa topik yang dapat diketahui yaitu antara lain:

- a. Konten – penyediaan source code Bootstrap
- b. Browser dan perangkat – tersedianya layanan yang dapat ditemukan untuk mempermudah dalam pencarian browser dan mobile yang didukung dan komponen berbasis seluler.
- c. JavaScript – memaparkan plugin JS yang berasal dari perkembangan JQuery
- d. Tema – terdapat variabel Sass untuk memudahkan dalam proses penyimpanan informasi sehingga pengguna lebih mudah dalam menggunakan aplikasi.
- e. Alat – bagian ini akan mengajarkan dalam membuat skrip npm Bootstrap.
- f. Aksesibilitas – tersedia batasan dan fitur yang disajikan dalam framework Bootstrap yaitu komponen, visibilitas, kontras warna, efek transisi, dan markup struktural.

Selain fitur yang disebutkan diatas pada dokumentasi Bootstrap terdapat juga layanan bagi pemula dalam membuat program dengan tersedianya contoh code yang dapat disalin dan juga dapat dikembangkan menjadi sebuah proyek. Meski dalam hal ini banyaknya kelebihan dari Framework ini namun, adapula kekurangan dari Bootstrap.



Adapun kekurangannya adalah sebagai berikut (Fauzy, 2010).

1. Gaya visual yang ditampilkan pada website bersifat monoton yang membuat tampilan akan menjadi kurang menarik.
2. Banyaknya fitur yang tersedia mengakibatkan ukuran file akan membengkak sehingga akan terjadi loading terhadap website namun untuk memecahkan masalah tersebut maka cukup dengan menggunakan versi file di `-minify`.

- Gaya Bootstrap yang cukup besar akan menimbulkan output pada HTML yang tidak dibutuhkna sehingga resource CPU akan terbuang-buang.

Setelah mengetahui kelebihan dan kekurangan dari Framework Bootstrap ini, maka selanjutnya adalah mengetahui tentang perbandingan dari generasi sebelumnya, di mana yang kita ketahui dari penjelasan sebelumnya bahwa bootstrap sudah mengalami pengembangan versi mulai dari tahun 2011 yaitu merupakan Bootstrap versi 1 dan kemudian dikembangkan sampai saat ini menjadi Bootstrap versi 5 yaitu pada tahun 2021. Namun pada buku ini, akan dibahas mengenai perbandingan Bootstrap 4 dan 5. Adapun perbandingannya dapat dilihat pada tabel berikut ini.

**Tabel 11.1:** Perbandingan Bootstrap 4 dan 5 (Sutiono, 2020).

| Hal Dasar         | Bootstrap 4                                                                                                           | Bootstrap 5                                                                                        |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| Logo              |                                     |                  |
| System Grid       | Memiliki 5 tier (xs, sm, md, lg, xl)                                                                                  | Memiliki 6 tier (xs, sm, md, lg, xl, xxl)                                                          |
| jQuery            | Memiliki fasilitas jQuery dan seluruh fitur yang terkait                                                              | Menggunakan VanilaJS dengan beberapa Plugin dan tidak menggunakan jQuery                           |
| Internet Explorer | Support IE 10 dan IE 11                                                                                               | Tidak lagi support IE 10 dan IE 11                                                                 |
| Form              | Terdapat tampilan yang berbeda pada setiap OS dan browser yang digunakan oleh pengguna pada Radio Button dan checkbox | Akan menampilkan form yang sama meskipun browser dan OS yang digunakan oleh pengguna berbeda       |
| Utilitas API      | Tidak dapat mengubah utilitas                                                                                         | Pada Versi ini akan memberikan kebebasan bagi pengguna dalam mengubah maupun membuat utilitas baru |

|                       |                                                                                                          |                                                                                                                                                |
|-----------------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Gutter</b>         | Menggunakan class <b>gutter</b> dengan ukuran font dalam px (pixel)                                      | Menggunakan class <b>g*</b> dengan ukuran font dalam rem                                                                                       |
| <b>Class vertikal</b> | Kolom dapat diposisikan default relatif                                                                  | Kolom tidak lagi pada posisi default relatif                                                                                                   |
| <b>Bootstrap Ikon</b> | Pada versi ini tidak memiliki ikon sendiri sehingga membutuhkan library font-awesom dari pihak ketiga    | Pada versi ini mempunyai SVG ikon sendiri                                                                                                      |
| <b>Jumbotron</b>      | Mendukung adanya Jumbotron                                                                               | Tidak mendukung adanya Jumbotron                                                                                                               |
| <b>Card</b>           | Card deck digunakan dengan width dan height yang seimbang                                                | Pada versi ini tidak menggunakan class card deck                                                                                               |
| <b>Navbar</b>         | Memiliki class dropdown berwarna putih secara default dan dropdown-menu dark pada properti inline-block. | Akan berubah menjadi dropdown warna hitam jika menggunakan class dropdown-menu-dark dikarenakan versi ini telah dihapus property inline-block. |

## 11.4 File Utama Bootstrap

File Utama yang yang terkompilasi pada Framework Bootstrap yang terdiri dari syntax yaitu: Bootstrap. css, Bootstrap.js dan Glyphicons. Untuk menjalankan plugin dan komponen JS maka dibutuhkan library JS atau biasa disebut dengan jQuery.

### 1. Bootstrap.css

Sebuah framework yang mengatur tata letak website, dengan fungsi ini maka developer dapat menciptakan tampilan yang tidak membutuhkan waktu yang lama untuk mengedit sesuai dengan kebutuhan dari developer (Anis, 2022).

### 2. Bootstrap.js

Pada Bootstrap terdiri dari file JavaScript yang merupakan bagian inti dari Bootstrap sebagai interaktivitas website. Salah satu yang dilakukan developer jika menulis beberapa syntax tanpa berulang kali dengan menggunakan jQuery. Adapun fungsi dari jQuery adalah

penggunaan properti CSS untuk animasi kustom; membuat widget; mengurangi data yang berasal dari lokasi lain secara dinamis.

### 3. Glyphicons

Pada bagian ini merupakan bagian yang penting dikarenakan untuk mempresentasikan data antarmuka pengguna, dalam hal ini glyphicons juga menyediakan ikon premium dan juga tersedia ikon tema secara gratis seperti Icons8, Falticon dan GlyphSearch.

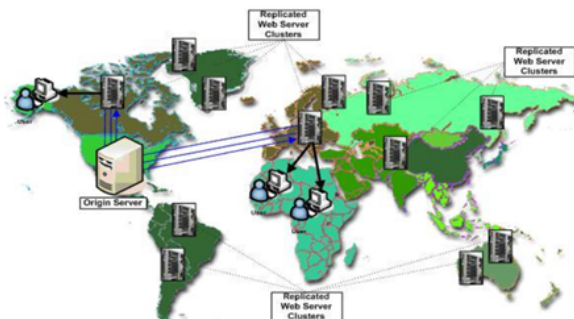
## 11.5 Cara Penggunaan Bootstrap

Dalam menggunakan Bootstrap 5 terdapat beberapa cara yang dapat dilakukan untuk menggunakan Bootstrap ini adalah sebagai berikut.

1. Menggunakan CDN;
2. Download secara manual.

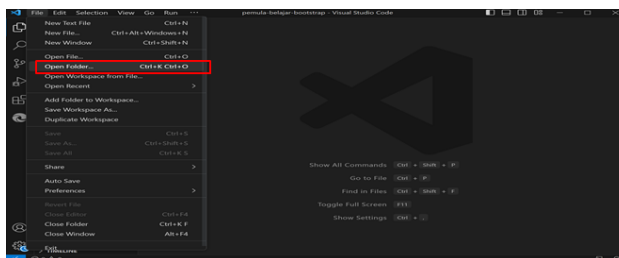
### 11.5.1 Menggunakan CDN

Menggunakan CDN merupakan cara yang digunakan pemula yang baru belajar dalam mengenal framework bootstrap ini, yaitu dengan mengcopy link CDN bootstrap, kemudian disimpan pada kode HTML. Namun, menggunakan cara seperti ini maka pengguna harus terhubung dengan jaringan internet. Sebelum melakukan langkah ini, penggunaan CDN terlebih dahulu harus mengetahui apa itu CDN. CDN merupakan singkatan dari Content Delivery Network, yaitu server yang tersebar diseluruh dunia yang mengantarkan konten secara cepat dan optimal berdasarkan pertimbangan koneksi dan geolokasi (Laksmiati, 2020).



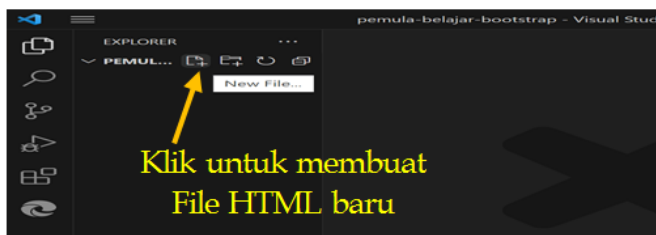
**Gambar 11.3:** Content Delivery Network (CDN) (Sitorus, 2022)

CDN adalah jaringan server yang proses pengiriman datanya dikirim melalui server CDN terdekat tanpa mengirim data pada server pusat, di mana setiap server pengganti memiliki duplikat data dari server utama. Buatlah folder baru dengan nama pemula-belajar-bootstrap, kemudian buka dengan visual studio Code. Klik File lalu pilih open folder.



**Gambar 11.4:** Membuka Folder

Setelah membuka folder yang telah dibuat dengan nama pemula-belajar-bootstrap, maka selanjutnya buatlah file HTML baru dengan nama belajar-bootstrap.html.



**Gambar 11.5:** Membuat HTML baru

Kemudian isi source code HTML belajar-bootstrap seperti pada gambar di bawah ini.

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <title>Belajar Bootstrap</title>
9
10 <!-- CSS Bootstrap -->
11 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-18m4498811"
12 />
13
14 <body>
15 <header class="bg-primary py-5">
16 <div class="container">
17 <h1 class="display-4 text-white">Belajar Bootstrap</h1>
18 </div>
19 </header>
20
21 <!-- JavaScript Bundle with Popper -->
22 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-k750014gtz2M01k1T
23 />
24 </body>
25 </html>

```

**Gambar 11.6:** Source Code pada HTML belajar-bootstrap

Pada Source code diatas menggunakan bootstrap dari CDN JSDeliver dengan class-class yang terdapat pada source code pada gambar 12.11 pada terdapat pada tag <header>. Kelebihan dalam menggunakan CDN adalah pengguna dapat menggunakan framework ini tanpa harus melakukan proses download, dengan menambahkan <head> dan <body> pada source code yang terdapat pada gambar diatas.

```

<!-- CSS Bootstrap -->
<link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.
css" rel="stylesheet" integrity="sha384-
18mE4kWBq781YhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy12QvZ6jIW3"
crossorigin="anonymous">

```

**Gambar 11.7:** <head> pada Source Code belajar-bootstrap

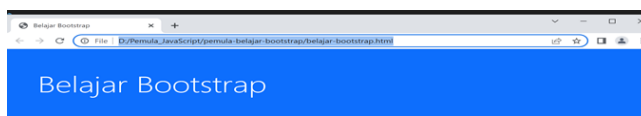
```

<!-- JavaScript Bundle with Popper -->
<script
src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle
.min.js" integrity="sha384-
ka7Sk06ln4gmtz2MlQnik11wXgYsOg+OMhUP+ILrH9sENB00LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

```

**Gambar 11.8:** <body> pada Source Code belajar-bootstrap

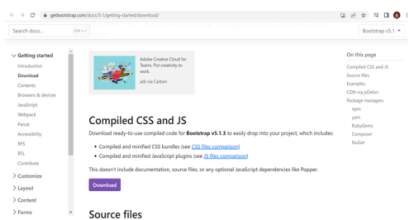
Hasil dari Source code belajar-bootstrap adalah sebagai berikut.



**Gambar 11.9:** Hasil Source code belajar-bootstrap

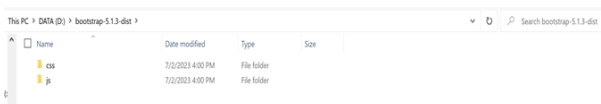
## 11.5.2 Download Secara Manual

Saat ini Bootstrap yang telah dirilis adalah versi 5.0, untuk mendownload Bootstrap v5.1.3 maka dapat dilakukan melalui website resmi pada <https://getbootstrap.com/docs/5.1/getting-started/download/>, kemudian pilih tombol download seperti pada gambar berikut.



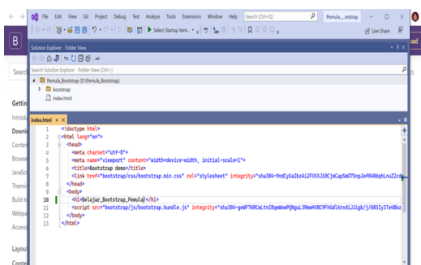
**Gambar 11.10:** Download Bootstrap

Setelah file telah berhasil terdownload menjadi file.zip, kemudian ekstrak file sehingga menjadi sebuah folder bootstrap dengan CSS dan JSS.



**Gambar 11.11:** Ekstrak File Bootstrap

Setelah file Bootstrap tersedia maka tahap selanjutnya untuk mengakses Framework ini dengan membuat file HTML. Pada lembar kerja yang akan dieksekusi diberi nama Pemula\_Bootstrap, kemudian langkah selanjutnya yaitu membuat file index.html di dalam folder yang telah tersedia yaitu Pemula\_Bootstrap. Dalam membuat index.html menggunakan aplikasi visual studi studi code. Berikut link dari aplikasi untuk memudahkan dalam membuat lembar kerja ini <https://code.visualstudio.com/>



**Gambar 11.12:** Source Code pada file index.html

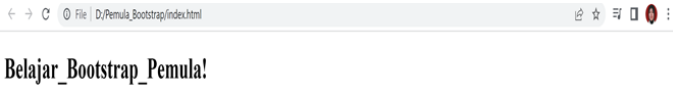


Pastikan dalam menulis lokasi file, karena jika tidak sesuai dengan tempat penyimpanan file maka bootstrap tidak akan terpanggil.

```
6 <title>bootstrap_nemoo</title>
7 <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9nOxyl9a1b2412F0V01j0Cj0nCap5m075np3eF8486qhLnu2Zcc
8 </head>
9 <body>
10 <h1>Belajar_Bootstrap_Pemula</h1>
11 <script src="bootstrap/js/bootstrap.bundle.js" integrity="sha384-gelw76RCat.tnZ8qMwPQNgul.3mmHVRBC9FtGdLkrxdi.13jgh/j/6851y3Te4Bkz
12 </body>
13 </html>
```

**Gambar 11.13:** Lokasi alamat file bootstrap

Kemudian setelah itu akses index.html melalui web browser:



**Gambar 11.14:** Tampilan kode CSS bootstrap

# Bab 12

## Sistem Grid Bootstrap

### 12.1 Memahami Sistem Grid pada Bootstrap

Bootstrap Grid system atau Sistem Grid Bootstrap adalah komponen inti dari kerangka kerja tersebut yang memungkinkan pengembang web untuk membuat tata letak responsif dengan mudah. Sistem Grid ini berbasis baris dan kolom, dan memungkinkan elemen-elemen pada halaman web untuk diatur dengan fleksibel sesuai dengan ukuran layar perangkat yang berbeda (Enterprise, 2016). Dengan menggunakan sistem Grid ini, dapat dengan mudah mengatur tata letak halaman web agar sesuai dengan berbagai ukuran layar, mulai dari perangkat mobile hingga desktop. Sistem ini terdiri dari 12 kolom dan 6 Breakpoint.

Satu kolom penuh panjangnya adalah 12. Jika kolom dibagi dua maka panjangnya akan menjadi 6. Jika dibagi lagi, maka panjang kolomnya akan semakin kecil, hingga yang paling kecil adalah 1. Ini artinya, kolom pada Grid hanya bisa dibagi sampai 12 saja. Selain untuk menentukan tata letak, Grid juga berfungsi sebagai guide line bagi desainer. Sehingga nanti akan membuat desain lebih mudah diterapkan ke dalam kode (Halim et al., 2021). Berikut adalah beberapa konsep penting dalam sistem Grid Bootstrap:

**Kelas Container:** Kelas “.container” adalah elemen induk yang membungkus semua konten pada halaman. Ada dua jenis.container dalam Bootstrap:.container: Mengatur lebar konten dan menambahkan margin pada sisi kiri dan kanan. Ideal untuk konten yang ingin diatur di tengah-tengah halaman..container-fluid: Mengisi seluruh lebar layar, sehingga konten menyesuaikan dengan ukuran layar.

**Kelas Row:** Kelas.row digunakan untuk membuat baris di dalam container. Baris ini kemudian berisi kolom-kolom. Setiap baris secara otomatis membentuk grup kolom dengan lebar maksimum yang sama.

**Kelas Column:** Kelas-kelas.col-\* digunakan untuk membuat kolom di dalam baris. Kolom ini dapat memiliki lebar yang berbeda sesuai dengan ukuran layar perangkat. Ada beberapa versi kelas ini berdasarkan ukuran layar:

.col-xs-\*: Untuk perangkat ekstra kecil (ekstra kecil layar smartphone).

.col-sm-\*: Untuk perangkat kecil (layar smartphone dan tablet dalam orientasi potret).

.col-md-\*: Untuk perangkat menengah (layar tablet dalam orientasi landscape dan layar desktop kecil).

.col-lg-\*: Untuk perangkat besar (layar desktop).

Dalam “\*”, Anda akan mengganti dengan angka 1 hingga 12 yang mewakili lebar kolom dalam sistem Grid.

**Offset dan Push/Pull:** Bootstrap juga menyediakan kelas-kelas untuk menggeser dan mengubah urutan kolom pada perangkat tertentu. ‘.col-md-offset-\*’ Menggeser kolom sejauh jumlah kolom yang ditentukan. ‘.col-md-push-\*’ Menggeser kolom ke arah kanan. ‘.col-md-pull-\*’ Menggeser kolom ke arah kiri.

```
<div class="container">
 <div class="row">
 <div class="col-md-6 col-sm-12">Kolom 1</div>
 <div class="col-md-6 col-sm-12">Kolom 2</div>
 </div>
</div>
```

**Gambar 12.1:** Penggunaan Sintaks “container”

Dalam contoh ini, pada layar perangkat menengah (md), kolom 1 dan kolom 2 masing-masing akan memiliki lebar setengah dari lebar container. Pada layar kecil (sm), kedua kolom akan menempati lebar penuh container.

## 12.2 Konsep dan Struktur Grid

Sistem Grid adalah sistem yang digunakan Bootstrap untuk mengatur tata letak (layout). Sistem ini terdiri dari 12 kolom dan 6 Breakpoint. Satu kolom penuh panjangnya adalah 12. Jika kolom dibagi dua maka panjangnya akan menjadi 6. Jika dibagi lagi, maka panjang kolomnya akan semakin kecil, hingga yang paling kecil adalah 1. Ini artinya, kolom pada Grid hanya bisa dibagi sampai 12 saja. Selain untuk menentukan tata letak, Grid juga berfungsi sebagai guide line bagi desainer. Sehingga nanti akan membuat desain lebih mudah diterapkan ke dalam kode (Anamisa and Mufarroha, 2022).



**Gambar 12.2:** Struktur Grid Bootstrap

Breakpoint adalah ukuran lebar yang menentukan tampilan responsif terhadap ukuran viewport perangkat tertentu. Saat ini Bootstrap memiliki 6 ukuran Breakpoint, yakni none, sm, md, lg, xl, dan xxl. nilai default Breakpoint adalah none. Ukuran ini adalah ukuran yang paling kecil, yakni sekitar 576px.

**Tabel 12.1:** Ukuran Breakpoint

No	Breakpoint	Class infix	Dimensi
1	X-Small	<i>None</i>	<576px
2	Small	sm	≥576px
3	Medium	md	≥768px

4	Large	lg	≥992px
5	Extra large	xl	≥1200px
6	Extra extra large	xxl	≥1400px

Berikut Ukuran Breakpoint sesuai dengan ukuran layar perangkat. Struktur dasar dari Grid Bootstrap sebagai berikut:

```

<div class="row">
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
</div>
<div class="row">
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
</div>
<div class="row">
 ...
</div>

```

**Gambar 12.3:** Membuat baris atau row

Untuk membuat sebuah baris gunakan sintax berikut ( `<div class="row">` ). Kemudian, tambahkan jumlah kolom yang diinginkan (tag dengan kelas `col-*-*` yang sesuai). Perhatikan bahwa angka-angka di dalam `col-*-*` harus selalu menghasilkan jumlah 12 untuk setiap baris. Di bawah ini beberapa contoh tata letak Grid dasar dari Bootstrap.

Untuk membuat 3 kolom yang besarnya sama, dalam 1 baris, maka digunakan sintaks berikut:

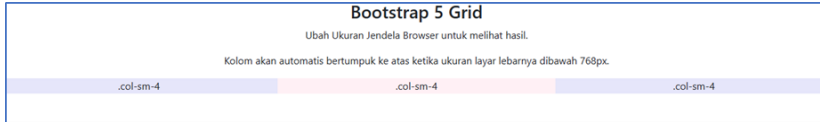
```

1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title Bootstrap demo title</title>
7 <link href="css/bootstrap.css" rel="stylesheet">
8 <script src="js/bootstrap.js"></script>
9 <script src="js/popper.min.js"></script>
10 </head>
11 <body>
12
13 <div class="text-center">
14
15 <h1>Bootstrap 5 Grid</h1>
16 <div class="container-fluid">
17
18 <p>Ubah ukuran jendela Browser untuk melihat hasil.</p>
19 <p>Kolom akan otomatis bertumpuk ke atas ketika ukuran layar lebarnya dibawah 768px.</p>
20 <div class="row">
21 <div class="col-sm-4" style="background-color: #lavender;"/></div>
22 <div class="col-sm-4" style="background-color: #lavender;"/></div>
23 <div class="col-sm-4" style="background-color: #lavender;"/></div>
24 </div>
25 </div>
26 </div>
27 </body>
28 </html>

```

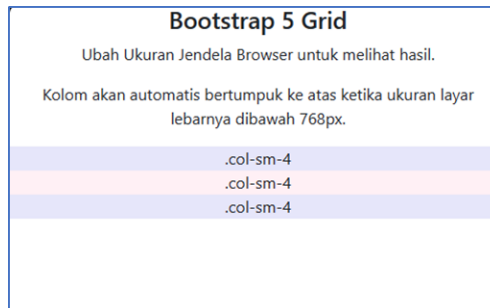
**Gambar 12.4:** Membuat baris atau row

Hasil untuk layar penuh sebagai berikut:



**Gambar 12.5:** Tampilan Layar Penuh 3 Kolom

Setelah mengubah ukuran layar dibawah 768px maka kolom akan saling menumpuk, seperti berikut:



**Gambar 12.6:** Tampilan Layar Smartphone 3 Kolom

Salah satu contoh lainnya, untuk membuat 2 kolom yang ukurannya berbeda dapat dilakukan dengan sintaks berikut:

```

1 <doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Bootstrap demo</title>
7 <link href="css/bootstrap.css" rel="stylesheet">
8 <script src="js/bootstrap.js"></script>
9 <script src="js/popper.min.js"></script>
10 </head>
11 <body>
12
13 <div class="text-center">
14 <h3>Bootstrap 5 Grid</h3>
15 <div class="container-fluid">
16
17 <p>Ubah Ukuran Jendela Browser untuk melihat hasil.</p>
18 <p>Kolom akan otomatis bertumpuk ke atas ketika ukuran layar lebarannya dibawah 768px.</p>
19 <div class="row">
20 <div class="col-sm-4" style="background-color: lavender;".col-sm-4</div>
21 <div class="col-sm-8" style="background-color: lavenderblush;".col-sm-8</div>
22 </div>
23 </div>
24 </body>
25 </html>

```

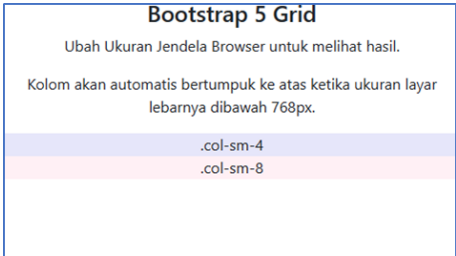
**Gambar 12.7:** Sintaks membuat 2 Kolom berbeda

Hasil dari sintaks dengan kolom pertama dengan ukuran 4 dan kolom kedua dengan ukuran 8. Hasilnya seperti pada gambar berikut:



**Gambar 12.8:** Tampilan 2 Kolom berbeda Layar Penuh

Dan apabila ukuran layar di ubah ke ukuran dibawah 768 px akan seperti berikut:



**Gambar 12.9:** Tampilan 2 Kolom berbeda Layar Smartphone

Untuk mengubah ukuran kolom misalnya 8 pada layar komputer dan tetap 12 pada layar smartphone maka dapat diberikan Breakpoint pada sintaks program.

```
<div class="col-12 col-lg-8">
 <!-- ukuran kolom ini 12 pada layar hp -->
 <!-- ukuran kolom ini 8 pada layar komputer -->
</div>
```

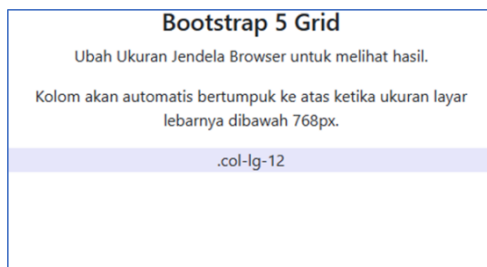
**Gambar 12.10:** Menggunakan Break point berbeda ukuran layar

Untuk tampilan pada layar komputer dekstop



**Gambar 12.11:** Menggunakan Break point berbeda ukuran layar Dekstop

Dan untuk tampilan pada smartphone akan penuk sebesar 12



**Gambar 12.12:** Menggunakan Break point berbeda ukuran layar Smartphone

## 12.3 Container dan Container Fluid

Container merupakan fondasi dasar dari blok layout. Container berfungsi untuk membungkus blok di dalamnya atau konten di dalam situs, sehingga terlihat rapi terhadap ukuran layar. Container juga memiliki Breakpoint. Ukuran kontainer akan 100% pada Breakpoint tertentu.

Misalnya, jika kita menggunakan class `container-md` maka lebar kontainer akan 100% pada layar *Extra Small dan Small* (Styawantoro and Komarudin, 2021).

### 1. Container

Dalam Bootstrap, "container" adalah elemen yang digunakan untuk mengatur konten di dalam tata letak yang terpusat dan memiliki lebar yang terbatas. Ini berarti bahwa konten di dalam elemen container akan ditempatkan di tengah-tengah halaman dan memiliki batas lebar tertentu, yang membuatnya tampak rapi dan mudah dibaca di layar yang lebih besar. Elemen container memiliki ukuran yang berbeda tergantung pada resolusi layar, karena Bootstrap menggunakan pendekatan responsif (Kholil and Akhsani, 2021).

### 2. Container Fluid

Di sisi lain, "container fluid" adalah elemen yang memenuhi seluruh lebar tampilan. Ini berarti kontennya akan diperluas sesuai dengan lebar tampilan yang tersedia, memberikan tampilan yang lebih luas dan memanfaatkan ruang layar yang lebih besar. Elemen container



fluid cocok digunakan ketika ingin konten mengisi lebar layar sepenuhnya tanpa batasan lebar tertentu.

Pemilihan antara container dan container fluid tergantung pada tata letak yang ingin di capai. Jika ingin konten yang terpusat dengan lebar yang terbatas, gunakan container. Namun, jika ingin tampilan yang lebih luas yang mengisi seluruh lebar layar, maka gunakan container fluid.

Contoh penggunaan:

```
<!-- Contoh penggunaan container -->
<div class="container">
 <!-- Konten di sini -->
</div>

<!-- Contoh penggunaan container fluid -->
<div class="container-fluid">
 <!-- Konten di sini -->
</div>
```

**Gambar 12.13:** Menggunakan Container dan Container-fluid

Hasil untuk penggunaan perintah “Container” tidak memenuhi layar



**Gambar 12.14:** Tampilan menggunakan Container

Hasil untuk penggunaan perintah “Container-fluid” lebar layar penuh



**Gambar 12.15:** Tampilan menggunakan Container-fluid

## 12.4 Struktur Dasar Grid dalam Bootstrap

### 12.4.1 Membuat Row dan Col

Class row dan col merupakan class untuk membuat baris dan kolom. Kedua class ini lah yang kita gunakan untuk membuat Grid. Class row berfungsi untuk membuat baris. Class ini menggunakan flex, namun Breakpoint tidak berlaku untuk class ini. Contoh penggunaannya seperti berikut:

```
<div class="row">

</div>
```

**Gambar 12.16:** Perintah membuat baris/row

Class col berfungsi untuk membuat kolom. Class ini harus dibungkus oleh class row agar menjadi Grid. Class col memiliki ukuran dan Breakpoint. Ukuran paling panjang adalah 12, dan ukuran paling pendek adalah 1. Format penulisannya sebagai berikut:

```
col-[breakpoint]-[ukuran]
```

**Gambar 12.17:** format membuat column

Dan contoh penggunaannya sebagai berikut:

```
<div class="row">
 <div class="col-lg-6 text-black" style="background-color: blueviolet;">col 1 row 1
 <!-- kolom 1 -->
</div>
<div class="col-lg-6 text-black" style="background-color: aqua;">col 2 row 1
 <!-- kolom 2 -->
</div>
</div>
<div class="row">
 <div class="col-lg-12 text-black" style="background-color: burlywood;">col 1 row 2
 <!-- kolom 3 -->
</div>
</div>
```

**Gambar 12.18:** Penggunaan column dan breakpoint serta ukuran

Hasilnya Sebagai berikut



**Gambar 12.19:** Tampilan membuat 2 column dan 2 row

## 12.4.2 Pengaturan Jumlah Kolom dan Lebar Kolom

Kolom dibangun di atas arsitektur kotak fleksibel Grid. Flexbox berarti kita memiliki opsi untuk mengubah kolom individual dan memodifikasi grup kolom pada baris (row). Kita dapat memilih bagaimana kolom dapat ditambahkan, menyusut, atau berubah. Saat membuat Grid layout, semua konten masuk dalam kolom. Hirarki Grid Bootstrap berpindah dari wadah ke baris ke kolom ke konten yang dibuat. Kita dapat menggabungkan konten dan kolom, tetapi perlu diperhatikan karena mungkin ada konsekuensi yang tidak diinginkan. Bootstrap menyertakan kelas yang telah ditentukan sebelumnya untuk membuat tata letak yang cepat dan responsif. Dengan enam Breakpoint dan beberapa kolom di setiap tingkat Grid, kami memiliki beberapa kelas yang telah dibuat untuk membuat tata letak yang diinginkan.

Jumlah kolom yang akan dibuat pada layout, sesuai dengan jumlah perintah `<div class="col">` Jika ingin membuat 2 kolom berarti perlu memasukkan 2 baris perintah `<div class="col">`

Sebagai Contoh:

```

15 <h3>Bootstrap 5 Grid</h3>
16 <div class="container-fluid">
17
18 <p>Ubah Ukuran Jendela Browser untuk melihat hasil.</p>
19 <p>Kolom akan otomatis bertumpuk ke atas ketika ukuran layar lebarnya dibawah 768px.</p>
20
21 <div class="row">
22 <div class="col-12 text-black" style="background-color: #blueviolet;">kolom 1 Baris 1
23 | <!-- kolom 1 -->
24 </div>
25 <div class="col-12 text-black" style="background-color: #aqua;">kolom 2 baris 1
26 | <!-- kolom 2 -->
27 </div>
28 </div>
29 <div class="row">
30 <div class="col-md-12 text-black" style="background-color: #lavender;"> Kolom 1 baris 2
31 | <!-- kolom 1 -->
32 </div>
33 </div>
34 </div>
35 </div>

```

**Gambar 12.20:** Sintaks membuat 2 column dan 2 row

Menghasilkan Tampilan sebagai berikut:



**Gambar 12.21:** Tampilan membuat column dan row

Format ukuran kolom dapat dituliskan setelah penulisan Breakpoint (`class="col-[Breakpoint]-[ukuran]"`) Ukuran Kolom penuh sebesar 12 dan ukuran terkecil adalah 1, kolom akan berpindah kebagian bawah pada layout jika jumlah keseluruhan ukuran melebihi 12.

```
<div class="row">
 <div class="col-lg-3 text-black" style="background-color: #blueviolet;">kolom 1 Baris 1
 <!-- kolom 1 -->
</div>
 <div class="col-lg-3 text-black" style="background-color: #aqua;">kolom 2 baris 1
 <!-- kolom 2 -->
</div>
 <div class="col-lg-3 text-black" style="background-color: #burlywood;">kolom 3 baris 1
 <!-- kolom 3 -->
</div>
 <div class="col-lg-3 text-black" style="background-color: #cadetblue;">kolom 4 baris 1
 <!-- kolom 4 -->
</div>
</div>
```

**Gambar 12.22:** Sintaks membuat 1 row dengan 4 col

Hasilnya Sebagai berikut dikarenakan jumlah ukuran ke 4 baris `<div class="col-lg-3">` adalah 12:



**Gambar 12.23:** Tampilan membuat 1 row dengan 4 col

## 12.5 Mengatur Tampilan Responsif dengan Grid System

### 12.5.1 Menggunakan Breakpoint untuk Perangkat Berbeda

Breakpoint dalam konteks Bootstrap merujuk pada titik-titik tertentu dalam lebar tampilan layar di mana tata letak atau desain elemen-elemen pada halaman web dapat berubah untuk memastikan responsivitas yang lebih baik. Bootstrap adalah kerangka kerja (framework) front-end yang populer yang digunakan untuk mengembangkan tata letak dan desain responsif pada situs web dan aplikasi web (Sunarya and Bahit, 2020). Dalam Bootstrap, Breakpoint adalah poin tertentu di dalam Grid responsif yang menandai perubahan dalam perilaku tata letak. Secara umum, Breakpoint digunakan untuk mengatur bagaimana tampilan situs web akan beradaptasi dengan lebar layar yang berbeda, seperti pada perangkat desktop, tablet, dan ponsel.

Bootstrap menyediakan beberapa Breakpoint bawaan yang dapat digunakan:

1. 'xs' (Extra Small): Lebar layar kurang dari 576px.
2. 'sm' (Small): Lebar layar antara 576px dan 768px.
3. 'md' (Medium): Lebar layar antara 768px dan 992px.
4. 'lg' (Large): Lebar layar antara 992px dan 1200px.
5. 'xl' (Extra Large): Lebar layar lebih dari 1200px.
6. 'xxl' (Extra extra Large): Lebar layar lebih dari 1400px.

Setiap Breakpoint dipilih untuk menampung kontainer dengan baik, yang lebarnya adalah kelipatan 12. Breakpoint juga mewakili subkelompok ukuran perangkat standar, dan dimensi area pandang mereka tidak secara khusus menargetkan setiap kasus penggunaan atau perangkat. Rentang tersebut memberikan fondasi yang baik dan konsisten untuk dibangun di hampir semua perangkat. Setiap Breakpoint memiliki pengaturan Grid yang berbeda. Misalnya, Anda dapat mengubah jumlah kolom dalam baris (row) Grid, menyembunyikan atau menampilkan elemen tertentu, atau mengatur ukuran elemen berdasarkan Breakpoint yang digunakan.

Contoh penggunaan Breakpoint dalam Bootstrap bisa seperti ini:

```
<div class='container'>
 <div class="row">
 <div class="col-md-6 col-lg-4">
 <!-- Isi konten -->
 </div>
 <div class="col-md-6 col-lg-4">
 <!-- Isi konten -->
 </div>
 <div class="col-md-12 col-lg-4">
 <!-- Isi konten -->
 </div>
 </div>
</div>
```

**Gambar 12.24:** Sintaks breakpoint medium (mg) dan large (lg)

Dalam contoh di atas, pada layar dengan lebar lebih besar atau sama dengan 992px (ukuran lg dan xl), semua kolom akan memiliki lebar yang sama (4 kolom dalam 1 baris). Namun, ketika lebar layar kurang dari 992px (ukuran md, sm, dan xs), kolom pertama dan kedua akan mengambil setengah lebar (6 kolom dalam 1 baris), sementara kolom ketiga akan mengambil seluruh lebar (12 kolom dalam 1 baris). Penggunaan Breakpoint ini membantu meningkatkan responsivitas situs web Anda terhadap berbagai ukuran layar, memastikan pengalaman pengguna yang baik pada berbagai perangkat.

## 12.5.2 Menyusun Tata Letak Responsif dengan Grid System

Tata letak responsif dengan Grid system adalah pendekatan yang umum digunakan dalam desain web untuk memastikan bahwa halaman web Anda dapat menyesuaikan diri dengan berbagai ukuran layar, mulai dari perangkat desktop hingga ponsel pintar. Salah satu framework yang populer untuk mencapai ini adalah CSS Grid.

Berikut adalah langkah-langkah umum untuk menyusun tata letak responsif menggunakan CSS Grid:

1. Rencanakan Tata Letak Anda:

Tentukan bagaimana Anda ingin tata letak halaman web Anda diatur. Pertimbangkan bagian-bagian penting seperti header, konten utama,

sidebar, dan footer. Tentukan juga berapa banyak kolom dan baris Grid yang diperlukan.

2. Struktur HTML:

Buat struktur HTML dasar yang mencerminkan tata letak yang Anda inginkan. Gunakan elemen semantik seperti `<header>`, `<main>`, `<aside>`, dan `<footer>` untuk mengelompokkan konten.

3. CSS Grid Container:

Pada elemen yang akan menjadi wadah untuk tata letak Grid, terapkan properti `display: Grid;`. Ini akan mengubah elemen menjadi Grid container yang siap untuk menampung Grid items (kotak-kotak yang membentuk tata letak).

4. Definisikan Grid Template:

Gunakan properti `Grid-template-columns` dan `Grid-template-rows` untuk mendefinisikan jumlah dan ukuran kolom serta baris dalam Grid. Anda dapat menggunakan unit seperti `fr` (fraksi dari ruang yang tersedia) atau `px` (pixel).

5. Menempatkan Konten:

Gunakan properti `Grid-column` dan `Grid-row` pada Grid items untuk menentukan posisi mereka dalam Grid. Anda juga dapat menggunakan properti seperti `Grid-column-start`, `Grid-column-end`, `Grid-row-start`, dan `Grid-row-end` untuk menentukan rentang sel yang diisi oleh item.

6. Menyesuaikan Ukuran pada Layar Berbeda:

Agar tata letak responsif, gunakan media queries. Anda dapat mengatur ulang ukuran kolom dan baris, atau bahkan mengubah tata letak sepenuhnya, sesuai dengan lebar layar yang berbeda.

Contoh kode CSS untuk mengatur tata letak responsif dengan CSS Grid.

```
1 .container {
2 display: grid;
3 grid-template-columns: 1fr 2fr; /* Contoh: dua kolom dengan lebar fraksi */
4 grid-template-rows: auto;
5 gap: 20px; /* Jarak antar grid items */
6 }
7
8 .grid-item {
9 /* Menempatkan item pada grid */
10 grid-column: span 1; /* Item akan mencakup 1 kolom */
11 grid-row: auto; /* Item akan menyesuaikan tinggi kontennya */
12 }
13
14 /* Responsif pada lebar layar kurang dari 768px */
15 @media (max-width: 768px) {
16 .container {
17 grid-template-columns: 1fr; /* Satu kolom pada layar lebih kecil */
18 }
19 }
```

**Gambar 12.25:** Sintaks CSS letak responsif

## 12.6 Kustomisasi Grid System

### 12.6.1 Menggunakan Offset dan Push-Pull

Di dalam kerangka kerja Bootstrap, Anda dapat menggunakan fitur push dan pull untuk mengatur posisi kolom dalam tata letak responsif. Fitur ini memungkinkan Anda untuk menggeser atau memindahkan kolom-kolom dalam susunan baris (rows) yang menggunakan sistem Grid Bootstrap.

Berikut penjelasan lebih lanjut tentang "push kolom" dan "pull kolom" dalam Bootstrap:

1. Push Kolom (Push Column):

Push kolom dalam Bootstrap mengacu pada memindahkan posisi kolom ke kanan dengan memberikan jarak antara kolom yang satu dengan yang lain. Anda dapat menggunakan kelas "push-\*" untuk menciptakan jarak tersebut.

2. Pull Kolom (Pull Column):

Pull kolom dalam Bootstrap mengacu pada memindahkan posisi kolom ke kiri dengan memberikan jarak di sebelah kiri kolom tersebut. Anda dapat menggunakan kelas "pull-\*" untuk menciptakan jarak tersebut.



Contoh penggunaan push dan pull kolom:

```

<body>
<div class="text-center">
<h3>Bootstrap 5 Grid</h3>
<div class="container">
<p>Ubah Ukuran Jendela Browser untuk melihat hasil.</p>
<p>Kolom akan otomatis bertumpuk ke atas ketika ukuran layar lebarnya dibawah 768px.</p>

<div class="row">
<div class="col-sm-4 col-sm-push-8" style="background-color: #lavender;"> .col-sm-4 .col-sm-push-8</div>
<div class="col-sm-8 col-sm-pull-4" style="background-color: #lavenderblush;"> .col-sm-8 .col-sm-pull-4</div>
</div>
</div>
</body>

```

**Gambar 12.26:** Sintaks push dan pull column

Pada contoh di atas, kolom pertama akan dipindahkan ke kiri sejauh empat kolom.

Kedua fitur ini, baik push maupun pull kolom, sangat berguna untuk mendapatkan tata letak yang diinginkan atau mengatur posisi konten dengan lebih presisi dalam tata letak Grid Bootstrap. Perhatikan bahwa contoh-contoh di atas menggunakan Breakpoint "md" (medium) untuk responsif, tetapi Anda dapat menyesuaikan kelas-kelas tersebut sesuai dengan desain Anda dan Breakpoint yang Anda tentukan (Kholil and Akhsani, 2021).

## 12.6.2 Mengatur Margin dan Padding dalam Grid

mengatur margin dan padding sangat penting untuk mengatur tata letak elemen dalam Grid atau layout halaman. Margin adalah ruang di luar batas elemen, sementara padding adalah ruang di dalam batas elemen. Ini memungkinkan untuk mengendalikan jarak antara elemen-elemen tersebut. Jika ingin menggunakan Grid dalam CSS, untuk dapat mengatur margin dan padding pada elemen-elemen dalam Grid dengan cara berikut (Rozi, 2015):

### Mengatur Margin

Margin dapat diatur menggunakan properti CSS 'margin'. Untuk dapat menggunakannya dengan nilai dalam piksel, persen, atau menggunakan kata kunci seperti 'auto':

```

.elem {
margin: 10px; /* margin sekitar elemen sebesar 10px */
margin-top: 20px; /* margin atas sebesar 20px */
margin-left: 15px; /* margin kiri sebesar 15px */
margin-right: 15px; /* margin kanan sebesar 15px */
margin-bottom: 30px; /* margin bawah sebesar 30px */
}

```

**Gambar 12.27:** Class pengaturan margin

## Mengatur Padding

Padding juga dapat diatur menggunakan properti CSS 'padding'. Cara penggunaannya mirip dengan margin:

```
.elem {
 padding: 10px; /* padding dalam elemen sebesar 10px */
 padding-top: 20px; /* padding atas sebesar 20px */
 padding-left: 15px; /* padding kiri sebesar 15px */
 padding-right: 15px; /* padding kanan sebesar 15px */
 padding-bottom: 30px; /* padding bawah sebesar 30px */
}
```

**Gambar 12.28:** Class pengaturan padding

Misalkan Anda memiliki struktur Grid yang terdiri dari beberapa elemen dalam kasus ini membuat 2 Elemen:

```
<div class="grid">
 <div class="elem">Elemen 1</div>
 <div class="elem">Elemen 2</div>
</div>
```

**Gambar 12.29:** Class pengaturan padding

Anda dapat mengatur margin dan padding elemen-elemen tersebut seperti ini:

```
.grid {
 display: grid;
 grid-template-columns: 1fr 1fr;
 gap: 20px;
}

.elem {
 padding: 10px;
 margin: 10px;
 background-color: lightgray;
}
```

**Gambar 12.30:** Class pengaturan padding

Dalam contoh ini, setiap elemen akan memiliki padding dan margin sebesar 10px. Selain itu, ada jarak (gap) sebesar 20px antara elemen-elemen dalam Grid.



# Bab 13

## Komponen Bootstrap

### 13.1 Komponen Bootstrap

Pada bab ini akan dibahas mengenai komponen pada framework bootstrap. Adapun komponen-komponen yang berada pada bootstrap adalah sebagai berikut: Carousel untuk slideshow, Navigasi bar (Navbar) yang memiliki tampilan yang interaktif (Suprayogi, 2019), Button Groups, button groups, pagination, membuat label dan badgets, jumbotron untuk menampilkan pesan, card, form dan footer di bootstrap 5.

#### 13.1.1 Menggunakan Carousel untuk Slideshow

Carousel merupakan proses penampilan gambar yang diberikan sebuah efek slide sehingga biasa disebut dengan istilah slideshow. Cara ini biasa digunakan pada halaman utama website sebagai salah satu cara untuk mempromosikan sebuah produk yang ingin dipasarkan dengan membuat sebuah slideshow dengan menampilkan design yang menarik, upaya ini dilakukan untuk menarik perhatian pelanggan dalam berkunjung pada halaman website yang telah disediakan. Dalam membuat carousel pada halaman website yang pertama dilakukan adalah menyediakan gambar yang akan ditampilkan pada website. Berikut adalah source code dalam membuat carousel dengan bootstrap (Kaban, 2017).

```

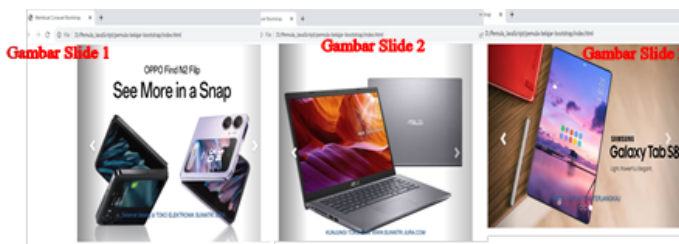
le Edit Selection View Go Run Terminal Help index.html - pemula-belajar-bootstrap - Visual Studio Co
 > belajar-bootstrap > index.html 4 x
 > index.html > html > body > div.col-md-8.col-md-offset-2
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <link rel="stylesheet" href="bootstrap.css">
6 <title>Membuat Corousel Bootstrap</title>
7 <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
8 <script type="text/javascript" src="js/jquery.js"></script>
9 <script type="text/javascript" src="js/bootstrap.js"></script>
10 </head>
11 <body>
12
13 <div class="col-md-8 col-md-offset-2">|
14
15 <div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
16 <!-- Indicators -->
17 <ol class="carousel-indicators">
18 <li data-target="#carousel-example-generic" data-slide-to="0" class="active">
19 <li data-target="#carousel-example-generic" data-slide-to="1">
20 <li data-target="#carousel-example-generic" data-slide-to="2">
21
22
23 <!-- Wrapper for slides -->
24 <div class="carousel-inner" role="listbox">
25
26 <div class="item active">
27
28 <div class="carousel-caption"> <div class="text-primary">
29 | Selamat datang di TOKO ELEKTRONIK SUWATRI JURA
30 </div>
31 </div>
32 </div>
33 <div class="item">
34
35 <div class="carousel-caption"> <div class="text-primary">
36 | KUNJUNGI TOKO KAMI WWW.SUWATRI JURA.COM
37 </div>
38 </div>
39 </div>
40 <div class="item">
41
42 <div class="carousel-caption"> <div class="text-primary">
43 | HARGA MURAH DAN TERJANGKAU
44 </div>
45 </div>
46 </div>
47 </div>
48
49 <!-- Controls -->
50
51
52 Previous
53
54
55
56 Next
57
58 </div>
59
60 </div>
61
62 <script src="js/jquery-1.12.4.min.js"></script>
63 <script src="js/jquery.easing.1.3.js"></script>
64 <!-- Include all compiled plugins (below), or include individual files as needed -->
65 <script src="js/bootstrap.min.js"></script>
66 <script src="js/script.js"></script>
67
68 </body>
69 </html>

```

Gambar 13.1: Source code membuat slideshow

1. Pada listing program diatas, pada baris 15 sampai 58 yang menentukan dalam membuat carousel. Pada baris 15 kita harus membuat class carousel beserta id nya, yang merupakan contoh dari carousel. Dalam membuat slide pada kasus ini dimulai dari 0 sampai dengan 2 sehingga akan tampil slide berjumlah 3 tampilan data
2. Item-item yang akan dijadikan slider terdapat pada baris 25 yaitu class carousel-inner dengan role listbox
3. Dalam membuat slider pertama terdapat pada baris 26 sampai 32. Baris 26 disertakan active untuk membuat class item. Gambar pertama yang akan ditampilkan pada baris 27 dengan nama `img/image111.jpg`.
4. Baris 28 sampai 32 berisi class carousel-caption, di mana slider pertama terdapat pada baris 28 setelah itu dibuatlah content yang akan dimasukkan pada slider. Baris 33 sampai 39 merupakan slider kedua sedangkan baris yang menunjukkan slider ketiga berada pada baris 40 sampai 47.

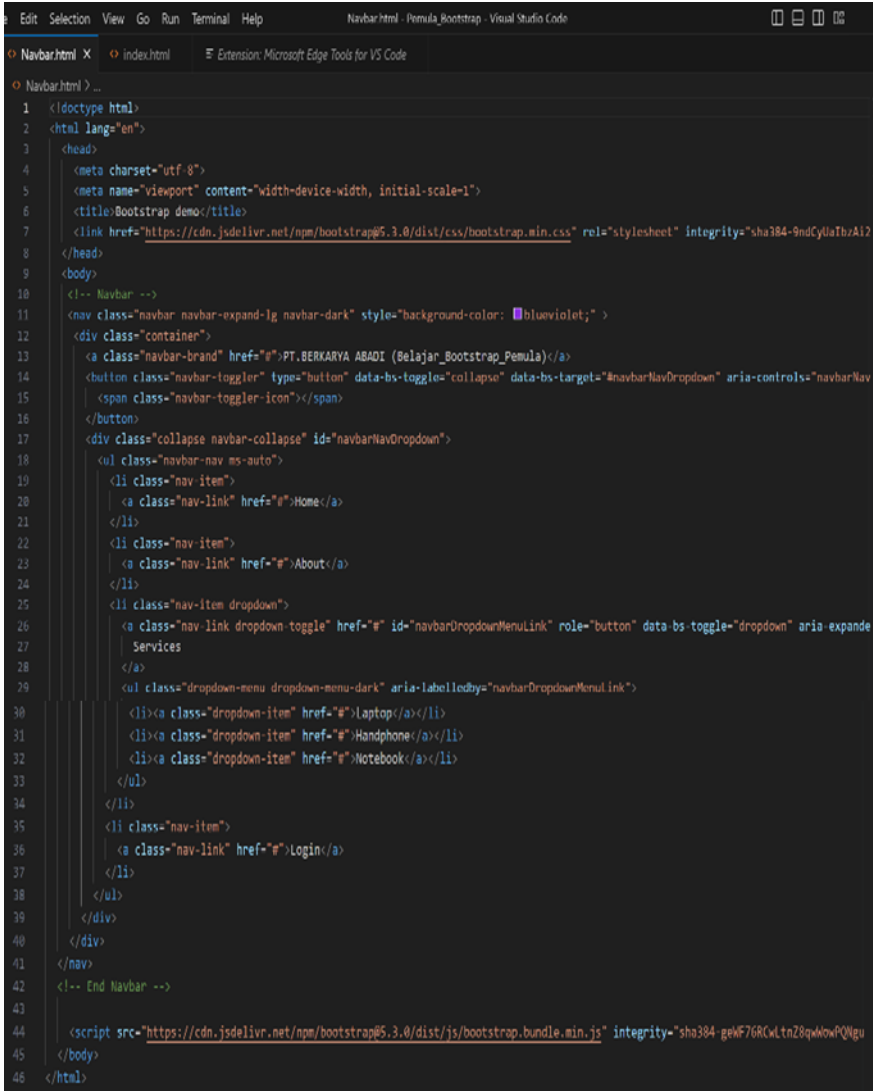
Hasil dari source code membuat slideshow di Bootstrap 5 dari framework bootstrap adalah sebagai berikut.



**Gambar 13.2:** Tampilan slideshow pada halaman website

### 13.1.2 Membuat Navigasi yang interaktif (Navbar)

Navbar merupakan menu utama yang berada pada header website yang berupa komponen dalam website yang berasal dari singkatan navigation bar. Komponen navbar ini biasa berisi nama website ataupun tempat untuk meletakkan logo dari perusahaan (Janah, 2023).



```

1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Bootstrap demo</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyliaIbA12
8 </head>
9 <body>
10 <!-- Navbar -->
11 <nav class="navbar navbar-expand-lg navbar-dark" style="background-color: #blueviolet;" >
12 <div class="container">
13 PT.BERKARYA ABADI (Belajar_Bootstrap_Pemula)
14 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav
15
16 </button>
17 <div class="collapse navbar-collapse" id="navbarNavDropdown">
18 <ul class="navbar-nav ms-auto">
19 <li class="nav-item">
20 Home
21
22 <li class="nav-item">
23 About
24
25 <li class="nav-item dropdown">
26 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-expande
27 Services
28
29 <ul class="dropdown-menu dropdown-menu-dark" aria-labelledby="navbarDropdownMenuLink">
30 Laptop
31 Handphone
32 Notebook
33
34
35 <li class="nav-item">
36 Login
37
38
39 </div>
40 </div>
41 </nav>
42 <!-- End Navbar -->
43
44 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-gwF76RCwltN28qWwPQlgu
45 </body>
46 </html>

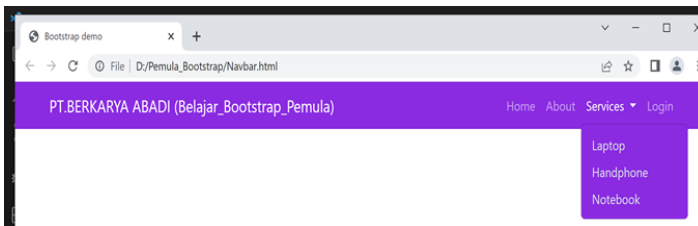
```

**Gambar 13.3:** Source Code Navbar

1. Untuk memanggil css bootstrap dapat dilihat pada baris 7 pada listing program diatas. Dalam membuat navbar dimulai dari baris 11 sampai 41 dengan class `<nav class="navbar-expand-lg navbar-dark">` dan menggunakan `style=background-color blueviolet` yaitu berwarna ungu.

2. Baris 12 membuat sebuah class container di mana tampilan yang diharapkan adalah posisi list menu atau lebar halaman akan fixed width.
3. Baris 14 sampai 16 untuk menampilkan button atau icon menu. Sebagai aksi dari icon menu pada class collapse navbar-collapse terdapat pada baris 17.
4. Baris 29 sampai 33 adalah merupakan dropdown menu yang terdapat pada tampilan list menu pada class dropdown-menu.

Hasil dari source code navbar dari framework bootstrap.



**Gambar 13.4:** Tampilan Navbar

### 13.1.3 Button Groups

Komponen ini digunakan untuk menumpuk beberapa tombol secara bersama pada satu baris, cara ini digunakan dalam menempatkan item seperti tombol alignment secara bersamaan. Adapun penggunaan button group dapat dilihat pada tabel 13.1.

**Tabel 13.1:** Penggunaan button group (Janah, 2023)

Kelas	Deskripsi	Contoh code
.btn-group	Pada kelas ini merupakan dasar penggunaan yang digunakan pada button group yaitu dengan class <b>btn</b> yang berada dalam <b>btn-group</b> .	<ol style="list-style-type: none"> <li>1. <code>&lt;div class="btn-group"&gt;</code></li> <li>2. <code>&lt;button type="button" class="btn btn-default"&gt;Button1&lt;/button&gt;</code></li> <li>3. <code>&lt;button type="button" class="btn btn-default"&gt;Button2&lt;/button&gt;</code></li> </ol>



		4. <code>&lt;/div&gt;</code>
<code>.btn-toolbar</code>	Berfungsi untuk menggabungkan set <code>&lt;div class="btn-group"&gt;</code> ke dalam <code>&lt;div class="btn-toolbar"&gt;</code> pada komponen kompleks	<ol style="list-style-type: none"> <li>1. <code>&lt;div class="btn-toolbar" role="toolbar"&gt;</code></li> <li>2. <code>&lt;div class="btn-group"&gt;..&lt;/div&gt;</code></li> <li>3. <code>&lt;div class="btn-group"&gt;..&lt;/div&gt;</code></li> <li>4. <code>&lt;/div&gt;</code></li> </ol>
<code>.btn-group-lg</code> , <code>.btn-group-sm</code> , <code>.btn-group-xs</code> ,	Penggunaan kelas ini untuk tombol namun, tidak akan mengubah ukuran pada tombol tersebut yang diterapkan pada button groups.	<ol style="list-style-type: none"> <li>1. <code>&lt;div class="btn-group btn-group-lg"&gt;..&lt;/div&gt;</code></li> <li>2. <code>&lt;div class="btn-group btn-group-sm"&gt;..&lt;/div&gt;</code></li> <li>3. <code>&lt;div class="btn-group btn-group-xs"&gt;..&lt;/div&gt;</code></li> </ol>
<code>.btn-group-vertical</code>	Kelas ini digunakan untuk tombol secara vertikal	<ol style="list-style-type: none"> <li>1. <code>&lt;div class="btn-group-vertical"&gt;</code></li> <li>2. <code>..</code></li> <li>3. <code>&lt;/div&gt;</code></li> </ol>

### 13.1.4 Pagination

Teknik yang digunakan untuk membantu pengguna aplikasi dalam mencari data tanpa menampilkan seluruh data, di mana teknik ini membantu jika data yang tersedia melebihi ratusan data dan mempermudah user agar tidak melakukan scroll terlalu jauh. Penggunaan pagination akan mempermudah dalam proses pencarian data dengan menampilkan halaman pada website secara berurutan (Alatas, 2013).

**Tabel 13.2:** Kelas paginaton (Janah, 2023)

Kelas	Deskripsi	Contoh code
.pagination	Menampilkan halaman dengan menggunakan kelas ini.	<pre>&lt;ul class="pagination"&gt;   &lt;li&gt;&lt;a href="#"&gt;&amp;lquo;&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;&lt;a href="#"&gt;1&lt;/a&gt;&lt;/li&gt;   .... &lt;/ul&gt;</pre>
.disable, .active	Untuk menunjukkan halaman saat ini dapat menggunakan .active dan untuk menyesuaikan dengan link dengan menggunakan kelas.disable	<pre>&lt;ul class="pagination"&gt;   &lt;li class="disabled"&gt;&lt;a href="#"&gt;&amp;lquo;&lt;/a&gt;&lt;/li&gt;   &lt;li class"active"&gt;&lt;a href="#"&gt;1&lt;span class="sr-only"&gt;(current)&lt;/span&gt;&lt;/a&gt;&lt;/li&gt;   .... &lt;/ul&gt;</pre>
.pagination- lg .pagination- sm	Kelas ini digunakan untuk membuat tampilan halaman dengan ukuran yang berbeda	<pre>&lt;ul class="pagination pagination- lg"&gt;..&lt;/ul&gt; &lt;ul class="paginaton"&gt;..&lt;/ul&gt; &lt;ul class="pagination pagination- sm"&gt;..&lt;/ul&gt;</pre>

Berdasarkan tabel diatas, maka penulis mencoba memasukkan kelas pagination ini kedalam source code. Pada kasus ini digunakan menggunakan tiga halaman yaitu terdiri dari halaman 1, 2 dan 3. Selain menunjukkan halaman, juga menampilkan keterangan previous dan next yang bisa digunakan untuk memindahkan halaman pada halaman berikutnya maupun halaman sebelumnya.

```

65 <script src="js/bootstrap.min.js"></script>
66 <script src="js/script.js"></script>
67 <ul class="pagination pagination-lg">
68 <li class="page-item">Previous
69 <li class="page-item">1
70 <li class="page-item">2
71 <li class="page-item">3
72 <li class="page-item">Next
73
74 </body>
75 </html>
```

**Gambar 13.5:** Source code pagination

Setelah penjelasan tentang membuat halaman pada website maka hasil dari source code pada gambar 13.5 adalah sebagai berikut.



**Gambar 13.6:** Tampilan pagination

### 13.1.5 Label dan Badges

Pada label digunakan untuk menampilkan pada halaman di website berupa tips, jumlah dan markup lainnya dengan menggunakan class pada tabel. Adapun class yang digunakan pada tabel adalah label-primary, label-success, label-info, label-default, label-warning dan label danger (Enterprise, 2016). Pada kasus ini samsung menggunakan class pada label-default. Adapun source code yang digunakan adalah sebagai berikut.

```
Samsung
Xiomay
Oppo
IPhone
DeLuna
Hiuwai
```

**Gambar 13.7:** Souce code label



**Gambar 13.8:** Tampilan source code label

Label mirip dengan badget namun perbedaan utamanya adalah digunakan untuk menampilkan informasi yang belum dibaca dengan tampilan pada bagian sudut lebih bulat.

```
Komentar produk anda 150
```



**Gambar 13.9:** Tampilan source code dan hasil dari badget

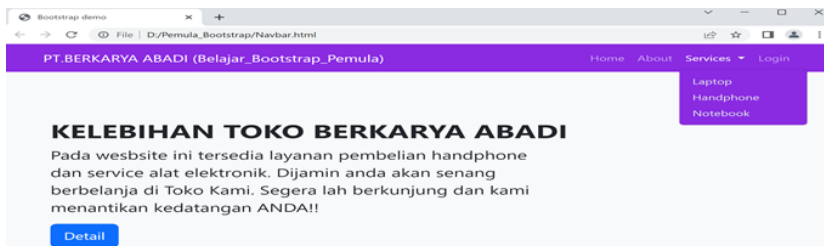
### 13.1.6 Jumbotron untuk menampilkan pesan

Komponen yang menampilkan informasi pada highlight di halaman website, komponen ini biasa digunakan oleh pemilik website untuk menampilkan informasi penting sehingga pengunjung dapat lebih fokus untuk memperhatikan pesan yang ditampilkan salah satu contohnya seperti promosi pada sebuah shop online (Kaban, 2017). Berikut adalah source code untuk membuat jumbotron.

```
43 <!--Jumbotron-->
44 <div class="p-5 mb-4 bg-light rounded-3">
45 <div class="container-fluid py-5">
46 <h1 class="display-10 fw-bold">KELEBIHAN TOKO BERKARYA ABADI</h1>
47 <p class="col-md-8 fs-4">Pada website ini tersedia layanan pembelian handphone dan service alat elektronik. Dijamin anda akan senang
48 <button class="btn btn-primary btn-lg" type="button">Detail</button>
49 </div>
50 </div>
51 <!--End Jumbotron-->
```

**Gambar 13.10:** Source Code Jumbotron

Hasil dari source code jumbotron dari framework bootstrap.



**Gambar 13.11:** Tampilan jumbotron

### 13.1.7 Card

Komponen yang berisi elemen yang menampilkan konten seperti video, teks dan gambar dengan menampilkan fitur yang memudahkan bagi developer untuk merancang tampilan website agar lebih responsif, menarik dan fleksibel (Faisal, 2020).

```

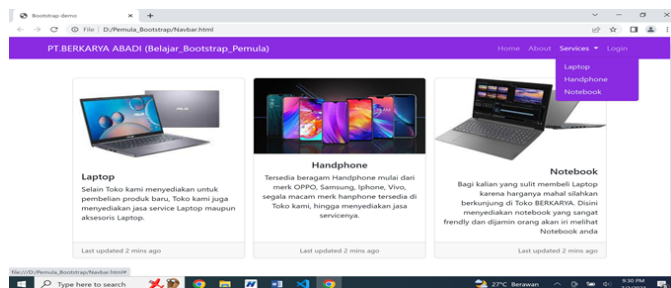
43 </-- card -->
44 <section id="card">
45 <div class="container">
46 <div class="row row-cols-1 p-5 row-cols-md-3 g-3">
47 <div class="col mb-5">
48 <div class="card h-100">
49
50 <div class="card-body">
51 <h5 class="card-title">Laptop</h5>
52 <p class="card-text">Selain Toko kami menyediakan untuk pembelian produk baru, Toko kami juga menyediakan jasa service Laptop</p>
53 </div>
54 <div class="card-footer">
55 <small class="text-muted">Last updated 2 mins ago</small>
56 </div>
57 </div>
58 </div>
59 <div class="col mb-5">
60 <div class="card h-100 text-center">
61
62 <div class="card-body">
63 <h5 class="card-title">Handphone</h5>
64 <p class="card-text">Tersedia beragam Handphone mulai dari merk OPPO, Samsung, Iphone, Vivo, segala macam merk hanphone terse</p>
65 </div>
66 <div class="card-footer">
67 <small class="text-muted">Last updated 2 mins ago</small>
68 </div>
69 </div>
70 </div>
71 <div class="col mb-5">
72 <div class="card h-100 text-end">
73
74 <div class="card-body">
75 <h5 class="card-title">Notebook</h5>
76 <p class="card-text">Bagi kalian yang sulit membeli Laptop karena harganya mahal silahkan berkunjung di Toko BERKARYA. Disini</p>
77 </div>
78 <div class="card-footer">
79 <small class="text-muted">Last updated 2 mins ago</small>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </section>
86 </-- End card -->

```

**Gambar 13.12:** Souce Code membuat Card

1. Pada listing program card diatas terdapat isi content yang terbagi atas body dan footer yang terdapat pada baris 50 dan 54. Pada area body dan footer menggunakan elemen <div>.
2. Pada baris untuk elihat gambar yang disisipkan pada card yang akan dibuat dengan menggunakan tiga card yaitu dapat dilihat pada baris 49 merupakan tampilan card untuk menunjukkan card 1 dan card ke 2 pada baris 61 dan card ke 3 berada pada baris ke 73.
3. Untuk memberikan informai terkaid isi card yang berada pada setiap card dalam kasus ini dengan menggunakan <p class="card-text"> kemudian menuliskan informasi yang akan di tampilkan pada halaman website.

Hasil dari source code card dari framework bootstrap.



Gambar 13.13: Tampilan Card

### 13.1.8 Form

Bagian ini merupakan elemen untuk menyempurnakan tampilan website dengan adanya form layout pada halaman web dan aplikasi. Form pada bootstrap akan menyederhanakan dalam pembuatan form alignment seperti input field, text area, label, tombol dan selectboxes. Adapun form layout terdiri dari 3 jenis pada bootstrap yaitu form vertical yang merupakan layout yang diterapkan pada form control tanpa menambah kelas dasar; form horisontal merupakan layout aligned yang disejajarkan dengan label dan form control yang berdampingan dengan kelas grid bootstrap; dan form sebaris (Huda, 2020).

```

> Navbar.html X index.html Extension: Microsoft Edge Tools for VS
0 Navbar.html > html > body > section#contact
86 <!-- cara -->
87 <!--Form-->
88 <section id="contact">
89 <div class="container">
90 <div class="mb-3 row text-center">
91 <h1>KOMENTAR</h1>
92 </div>
93
94 <div class="row justify-content-center">
95 <div class="col-md-6">
96 <form>
97 <div class="mb-3">
98 <label for="exampleInputEmail1" class="form-label">Email</label>
99 <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
100 </div>
101 <div class="mb-3">
102 <label for="exampleFormControlTextarea1" class="form-label">Komentar</label>
103 <textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
104 </div>

```

```

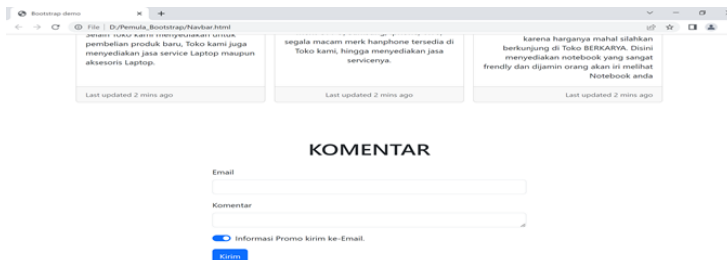
105 <div class="mb-3 form-check form-switch">
106 <input class="form-check-input" type="checkbox" role="switch" id="flexSwitchCheckDefault">
107 <label class="form-check-label" for="flexSwitchCheckDefault">Informasi Promo kirim ke-Email.</label>
108 </div>
109
110
111 <div class="mb-3">
112 <button type="submit" class="btn btn-primary">Kirim.</button>
113 </div>
114
115 </form>
116 </div>
117 </div>
118
119 </div>
120 </section>
121 <!--end services-->
122
123 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-gwW769Culcn28qWov9qNu"
124 </body>
125 </html>

```

**Gambar 13.14:** Souce Code membuat form

Pada baris 88 sampai pada baris 117 merupakan gambaran untuk membuat form inputan yang akan dilakukan pada kasus ini, di mana dalam kasus ini menggunakan label class="form-label" yang terdapat 4 data yang akan ditampilkan yaitu berupa data email, komentar, informasi promo kirim ke-email dan kirim.

Hasil dari source code form dari framework bootstrap.



**Gambar 13.15:** Tampilan form

### 13.1.9 Footer di Bootstrap 5

Bagian ini akan mengimplementasikan informasi akhir yang biasa digunakan oleh developer berupa informasi alamat dari perusahaan ataupun sosial media yang dapat diperoleh informasinya yang pada studi kasus alamat tertetera

menggunakan text-end untuk membuat text menjadi rata kanan, dan grid yang digunakan yaitu col-xxl pada ultra wide screen dan col-sm grid pada layar kecil (Suhemi, 2023). Adapun source code yang digunakan pada studi kasus ini adalah sebagai berikut.

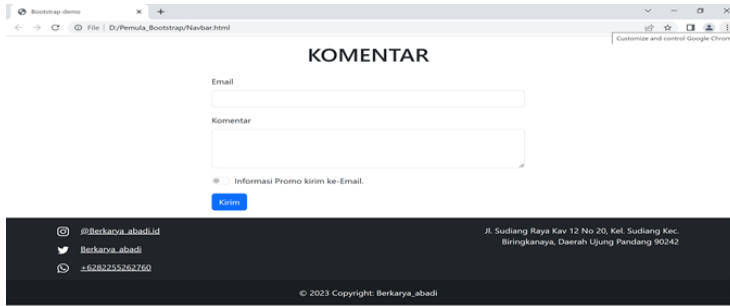
```
Navbar.html X index.html Extension: Microsoft Edge Tools for VS Code
</services-->
</footer-->
123 <footer class="bg-dark text-white mb-3">
124 <div class="container">
125 <div class="row p-3">
126 <div class="col-sm-8 col-xxl-9">
127 <div class="mb-3">
128 <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor" class="bi bi-instagram"
129 viewBox="0 0 16 16">
130 <path
131 d="M16 8.064 14.349 16 8.064 14.349 0 0 8.064 14.349 16 8.064" />
132 </svg>
133 @Berkarya_abadi.id.
134 </div>
135 </div>
136 <div class="mb-3">
137 <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor" class="bi bi-twitter"
138 viewBox="0 0 16 16">
139 <path
140 d="M5.026 15c6.038 0 9.341-5.033 9.341-9.341 0-.14-.002-.282-.006-.422A6.685 6.685 0 0 1 16 3.542a6.685 6.685 0 0 1-1-1.889S15.53 1.649 12.147 0c-.931-.94-1.924-.94-2.927 0 0 0-.42 2.76C12.147 0 9.341 5.033 9.341 9.341 0 15 5.026 15" />
141 </svg>
142 Berkarya_abadi.
143 </div>
144 </div>
145 <div>
146 <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor" class="bi bi-whatsapp"
147 viewBox="0 0 16 16">
148 <path
149 d="M13.061 2.326A7.854 7.854 0 0 0 9.341 7.994c0 .068.068.13.13.188c.224.176.457.324.688.445c.232.12.464.218.696.294c.232.076.464.13.696.176c.232.048.464.096.696.144c.232.048.464.096.696.144c.232.048.464.096.696.144c.232.048.464.096.696.144c.232.048.464.096.696.144" />
150 </svg>
151 +6282255262760.
152 </div>
153 </div>
154 </div>
155 <div class="col-sm-4 col-xxl-3 text-end mb-3">
156 Jl. Sudiang Raya Kav 12 No 20, Kel. Sudiang Kec. Biringkanaya,
157 Daerah Ujung Pandang
158 90242
159 </div>
160 </div>
161 </div>
162 <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">
163 © 2023 Copyright: Berkarya_abadi
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
</footer-->
```

Gambar 13.16: Source Code membuat footer

Pada listing diatas pada baris 123 merupakan class bg-dark dan text-white yang memberikan tampilan warna latar belakang gelap dengan tulisan berwarna putih pada footer. Penggunaan class container untuk menampilkan konten footer.



Hasil dari source code footer di Bootstrap 5 dari framework bootstrap.



**Gambar 13.17:** Tampilan footer

### 13.1.10 Mengatur Tampilan Gambar

Pada Bootstrap tersedia class untuk mengatur tampilan gambar dan bentuknya, yaitu dalam bentuk circle, rounded, dan thumbnail maupun bentuk yang ditampilkan dalam bentuk responsif dengan mengikuti resolusi layar pengguna.

Pengaturan gambar dalam bootstrap terdiri dari beberapa class-class. Adapun class-class pada bootstrap yang digunakan adalah sebagai berikut (Kaban, 2017) (Hannan, 2023).

1. **Img-rounded**

Class ini digunakan untuk membuat tampilan dalam bootstrap berbentuk bulat atau melengkung pada keempat sudutnya.

2. **Img-circle**

Pada class ini digunakan tampilan berupa bentuk seperti lingkaran.

3. **Img-thumbnail**

Pada class ini digunakan untuk membuat tampilan seperti bingkai dengan sebuah jarak pemisah

4. **Img-responsive**

Pada kelas ini untuk menampilkan gambar mengikuti lebar dari resolusi pengguna dikarenakan tampilan bersifat responsif.

Adapun cara mengatur gambar dapat dilihat pada source code berikut ini.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
6 <meta http-equiv="x-ua-compatible" content="ie=edge" />
7 <title>Materi Bootstrap 5</title>
8 <!-- Font Awesome -->
9 <link
10 rel="stylesheet"
11 href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
12 />
13 <!-- MDB -->
14 <link rel="stylesheet" href="css/mdb.min.css" />
15 </head>
16 <body>
17 <!-- Start your project here-->
18 <div class="container bg-light p-4 mt-4 overflow-hidden">
19 <h2 class="text-center"><div class="text-success">Mengatur Tampilan Gambar </div></h2>
20 <p class="text-justify text-center">Tampilan Gambar 1 Thumbnail,
21 Gambar 2 tentang circle dan Gambar 3 adalah rounded.</p>
22 <div class="container">
23 <div class="row">
24 <div class="col-12 col-sm-4 mb-2">
25
27 <p class=""><div class="text-danger"> Gambar 1</div></p>
28 </div><!--/first column-->
29 <div class="col-12 col-sm-4 mb-2">
30
32 <p class=""><div class="text-danger"> Gambar 2</div></p>
33 </div><!--/second column-->
34 <div class="col-12 col-sm-4 mb-2">
35
37 <p class=""><div class="text-danger"> Gambar 3</div></p>
38 </div><!--/third column-->
39 </div> <!--/row-->
40 </div><!--/container-->
41
42 <!-- End your project here-->
43
44 <!-- MDB -->
45 <script type="text/javascript" src="js/mdb.min.js"></script>
46 <!-- Custom scripts -->
47 <script type="text/javascript"></script>
48 </body>
49 </html>

```

**Gambar 13.18:** Source code mengatur tampilan gambar

Pengaturan tentang tampilan gambar terdapat pada baris 22 sampai 40, untuk menampilkan gambar seperti terlihat pada gambar dibawah ini yang terbagi atas tiga gambar yaitu gambar 1 terdapat pada baris 25 dengan tampilan gambar dengan menggunakan `img-thumbnail` yaitu sebuah gambar yang akan menampilkan gambar bingkai, sedangkan gambar 2 terdapat pada baris 30 yang menampilkan gambar berbentuk `circle` yaitu berbentuk lingkaran dan gambar ke 3 terdapat pada baris 35 yaitu `rounded` yaitu menampilkan gambar dalam bentuk melengkung pada keempat sisinya.

Berdasarkan source code pada kasus diatas maka hasil yang diperoleh dapat dilihat pada gambar 13.19 berikut ini.



**Gambar 13.19:** Tampilan gambar thumbnail, circle dan rounded

# Daftar Pustaka

- Abdulloh, R. (2022) 7 Materi Pemrograman Web Untuk Pemula 1: HTML, CSS, & MariaDB. Edited by eEp (as.prastito@gmail.com). Jakarta: PT. Elex Media Komputindo.
- Adams, C. et al. (2007) The Art & Science of CSS, Design.
- Addy Osmani (2023) "Learning JavaScript Design Patterns", O'Reilly Media
- Adri, M. (2018) "Bootstrap 4: Designing Awesome Responsive Website," Yogyakarta: ANDI.
- Ahmad Muhardian, n.d. Belajar HTML #07: Cara Membuat Link untuk Menghubungkan Halaman Web. 28 Juni 2020.
- Ahmad Muhardian, n.d. Belajar HTML #13: Cara Menambahkan Video pada Web. 22 Juli 2020.
- Ahmad Muhardian, n.d. Belajar HTML #14: Cara Menambahkan Audio pada HTML. 27 Juli 2020.
- Alatas, H. (2013) "Responsive Web Design dengan PHP & Bootstrap," Yogyakarta: Lokomedia.
- Albar, A. A. (2021) "Operator Logika," Available at: [https://rstudio-pubs-static.s3.amazonaws.com/825480\\_4cdbc1925e3a441bb63b68ab219745fe.html](https://rstudio-pubs-static.s3.amazonaws.com/825480_4cdbc1925e3a441bb63b68ab219745fe.html) (Accessed: July 13, 2023).
- Anamisa, D. R. and Mufarroha, F. A. (2022) Dasar Pemrograman WEB Teori dan Implementasi: HTML, CSS, Javascript, Bootstrap, CodeIgniter. Media Nusa Creative (MNC Publishing).
- Anis, Y., Purwatiningsy., Retnowati., Fajrina, E. A. N. (2022) "Penerapan Framework Bootstrap Dalam Sistem Informasi Rekam Medis Data

- Posyandu dengan Metode Waterfall," *Jurnal Sistem Komputer dan Informatika*, 4(2), hal. 310-318.
- Banks, M., & Card, O. S. (2008). *On the Way to the Web The Secret History of the Internet and Its Founders*.
- Bowers, M., Synodinos, D. and Sumner, V. (2011) *Pro HTML5 and CSS3 Design Patterns, Pro HTML5 and CSS3 Design Patterns*. Available at: <https://doi.org/10.1007/978-1-4302-3781-5>.
- Brooks, D.R. (2011) *Guide to HTML, JavaScript and PHP, Guide to HTML, JavaScript and PHP*. Available at: <https://doi.org/10.1007/978-0-85729-449-4>.
- David Flanagan (2020) "JavaScript: The Definitive Guide", O'Reilly Media
- Dermawan, D. A., Mashuri, C., Permadi, G. S., Gunawan, D. A., Widiasih, D. (2022) "Membuat Game Berbasis Website Menggunakan Bahasa JavaScript dan PHP
- Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
- Enterprise, J. (2016) *Pengenalan HTML dan CSS*. Jakarta: PT. Elex Media Komputindo.
- Eric T. Freeman dan Elisabeth Robson (2014) "Head First JavaScript Programming", O'Reilly Media
- Faisal, M. R., Abadi, F. (2020) "Pemrograman Web Dasar I: Belajar HTML 5," Banjarbaru, Kalimantan Selatan: Scripta Cendekia.
- Fauzy, A & Supandi, E. D. (2010) "Memprediksi Interval Reliabilitas Produk Dengan Metode Bootstrap Persentil," *Jurnal ILMU DASAR*, 11(2), hal. 120-123.
- Flanagan, D. (2020). *JavaScript: The Definitive Guide, Seventh Edition*. O'Reilly Media.
- Foundation Website Creation with CSS, XHTML, and JavaScript (2008) *Foundation Website Creation with CSS, XHTML, and JavaScript*. Available at: <https://doi.org/10.1007/978-1-4302-0992-8>.

- Halim, A. et al. (2021) 'Pelatihan Dasar Pembuatan Website Sederhana Menggunakan Framework Bootstrap', *Praxis: Jurnal Pengabdian Kepada Masyarakat*, 1(1), pp. 134–139.
- Hannan, ABD. (2023) "Tutorial Framework Bootstrap Bahasa Indonesia," Available at: [https://www.academia.edu/13821490/TUTORIAL\\_FRAMEWORK\\_BOOTSTRAP\\_BAHASA\\_INDONESIA](https://www.academia.edu/13821490/TUTORIAL_FRAMEWORK_BOOTSTRAP_BAHASA_INDONESIA) (Accessed: August 15, 2023)
- Harris, M. (2021) "Mengetahui Operator Bahasa Pemrograman: Pengertian, Sifat dan Jenisnya," Available at: <https://www.gramedia.com/literasi/operatoradalah/#:~:text=Operator%20increment%20digunakan%20untuk%20menambah,mengurangkan%20variabel%20sebanyak%20satu%20angka/> (Accessed: July 12, 2023)
- Hendrawan, I. R., Antara, P., Indraswari, A. D., Widihasani, A. F., dan Utami, E. (2023) "Analisis Aspek Kualitas Skema Basis Data Pada Sistem Informasi Tour dan Travel," *JIKO (Jurnal Informatika dan Komputer)*, 7(1), hal. 37-45.
- Hidayah, A., Aulia, A. S., Bherta, R., Indirawati, D. (2018) "Membangun Website Sekolah Luar Biasa (SLB) Martapura Oku Timur Dengan Menggunakan PHP dan MYSQL," *JTIM: Jurnal Teknik Informatika Mahakarya*, 1(2), hal. 11-19.
- Homaidi, A. (2016) "Sistem Informasi Akademik AMIK Ibrahimy Berbasis Web," *Jurnal Ilmia Informatika*, 1(1), hal. 17-23.
- Huda, M. (2020) "Bootstrap 4:Belajar CRUD Menggunakan PHP dan MySQL," Malang:AE Publishing.
- Ilham, N. A., Naziro. (2019) "Implementasi Konsep Pemrograman Berorientasi Objek Pada Aplikasi Sistem Parkir Menggunakan Bahasa Pemrograman Java," *Jurnal Edukasi Elektro*, 3(2), hal. 63-69.
- Jamsa, K., King, K., & Anderson, A. (2002). *HTML & Web Design Tips & Techniques*. McGraw Hill LLC.
- Janah, N. (2023) "Komponen bootstrap," Available at: [https://www.academia.edu/12285144/komponen\\_bootstrap](https://www.academia.edu/12285144/komponen_bootstrap) (Accessed: July 30, 2023)

- Javascript.info. (n.d.). The Modern JavaScript Tutorial. [online] Available at: <https://www.javascript.info> (Accessed 11 Juni 2023).
- Josi, A. (2016) "Implementasi Framework Bootstrap pada Website STMIK Prabumulih," *Jurnal Manajemen dan Informatika Komputer Pelita Nusantara (MANTIK)*, 20(2), hal. 1-4.
- Jubilee Enterprise. (2018). *HTML, PHP, dan MySQL untuk Pemula*. Elex Media Komputindo.
- Kaban, R. (2017) "Design Website Responsive Dengan Bootstrap," Available at: <http://itgeek.id/download> (Accessed: August 15, 2023)
- Kholil, M. and Akhsani, R. (2021) *DESAIN DAN PEMROGRAMAN WEB: HTML, CSS, PHP, MYSQL, & BOOTSTRAP*. Klik Media.
- Kode, P. (2023) *Tutorial Dasar CSS Untuk Pemula*, <https://www.petanikode.com/tutorial/css/>.
- Kusumawardani, D. M. et al. (2023) *Web Dasar Menggunakan HTML, CSS, JS, PHP dan Studi Kasus*. Edited by Efitra. Jambi: PT. Sonpedia Publishing Indonesia.
- Laksmiati, D. (2020) "Implementasi Content Delivery Network (CDN) Untuk Optimasi Kecepatan Akses Website," *Jurnal AKRAB JUARA*, 5(1), hal. 49-56.
- Lay, M.E. (2017) 'E-Commerce Gitar Akustik Dan Sparepart Kota Malang Menggunakan Metode Customer To Customer', *Jurnal Mahasiswa Teknik Informatika*, 1(2), pp. 1-7.
- Lubis, A. B. U., Sembiring, B. O., Irwan, D. (2022) "Penerapan Framework Bootstrap Dalam Pengembangan Sistem Informasi Isi Ulang Parfum Pada LE Plus Parfume Berbasis Web," *Djtechno: Journal of Information Technology Research*, 3(1), hal 66-75.
- Marijn Haverbeke (2018) "Eloquent JavaScript, A Modern Introduction to Programming," No Starch Press.
- Mariko, S. (2019) 'Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus', *Jurnal Inovasi Teknologi Pendidikan*, 6(1), pp. 80-91. Available at: <https://doi.org/10.21831/jitp.v6i1.22280>.

- Mastra, K. N. L., & Dharmawan, R. F. (2018). Tinjauan User Interface Design Pada Website E-Commerce Laku6. *Narada*, 5(1), 83–94.
- Matuszek, D. (2023). *Quick JavaScript*. CRC Press.
- MDN Contributors. (2019). [online] JavaScript. MDN Web Docs. Available at: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- Melorse, J., Perroy, R. and Careas, S. (2015) HTML, XHTML, and CSS, Statewide Agricultural Land Use Baseline 2015.
- Meyer, E.A. (2004) 'Cascading Style Sheets: The Definitive Guide', Style (DeKalb, IL), 104(March).
- Minnick, C. (2023). *JavaScript All-in-One for Dummies*. John Wiley & Sons.
- Morgan, N. (2022). *JavaScript Crash Course*. No Starch Press.
- Mozilla (2023) Introduction to the CSS basic box model - CSS: Cascading Style Sheets | MDN. Tersedia pada: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_box\\_model/Introduction\\_to\\_the\\_CSS\\_box\\_model](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_box_model/Introduction_to_the_CSS_box_model) (Diakses: 18 Agustus 2023).
- Murhadian, A., 2020. Belajar HTML #08: Cara Menampilkan Gambar di HTML. [online] Available at: <https://www.petanikode.com/html-img/>.
- Muthohir, M. (2021) *Mudah Membuat Web Bagi Pemula*. Yayasan Prima Agus Teknik.
- Muttaqin, dkk (2023). *Pengantar Teknologi Digital*. Yayasan Kita Menulis.
- Nasir, M., Akbar, M. (2011) "Perancangan Website SMP Negeri 2 Lamasi," *Jurnal Ilmiah d'ComputarE*, vol. 1, hal. 42-49
- Nayoan, A. (2022) Apa itu CSS? Pengertian, Fungsi, dan Contohnya, <https://www.niagahoster.co.id/blog/pengertian-css/>.
- Nurmalasari, N., Anna, A., Arissusandi, R. (2019) "Rancang Bangun Sistem Informasi Akutansi Laporan Laba Rugi Berbasis Web Pada PT United Tractors Pontianak," *Evolusi : Jurnal Sains dan Manajemen*, 7(2), hal. 6-14.
- Pahlevi, O., Mulyani, A. and Khoir, M. (2018) 'SISTEM INFORMASI INVENTORI BARANG MENGGUNAKAN METODE OBJECT



- ORIENTED DI PT. LIVAZA TEKNOLOGI INDONESIA JAKARTA', Jurnal PROSISKO, 5(1). Available at: <https://livaza.com/>.
- PawPrints Learning Technologies. (2014). *Beginning PHP & MySQL Development: Code Your Own Dynamic Website Today*. Lulu.com.
- Prasetiadi, A. E. (2011). *Web 3.0: Teknologi Web Masa Depan*. 1(3), 1–6.
- Puspa, R. (2022) Apa itu CSS? Pengertian, Kelebihan, dan Jenis CSS! Available at: <https://academy.alterra.id/blog/apa-itu-css/> (Accessed: 16 August 2023).
- Putawa, R. A. (2022) "Makna Filosofis Ketiadaan dan Relevansinya dengan Tipe Data Undefined Pada JavaScript," *Jurnal Filsafat Indonesia*. 5(1), hal. 80-86.
- Putra, M. Y. (2020) "Responsive Web Design Menggunakan Bootstrap Dalam Merancang Layout Website," *Information System For Educator And Professionals*, 5(1), hal. 61-70.
- Queirós, R. (2018) 'CSS Preprocessing: Tools and automation techniques', *Information* (Switzerland), 9(1). Available at: <https://doi.org/10.3390/info9010017>.
- Rakhmadani, D. P., Putra, Y. D., Dianti, I. L. (2020) "Transformasi Digital Pada Bisnis UMKM Dengan Penerapan DBMS," *Jurnal EKSEKUTIF*, 17(2), hal. 258-279.
- Riasinir, T. J., Widyasari. (2019) "Pemanfaatan Framework Bootstrap Dalam Merancang Website Responsif Untuk Toko D2 Adventure," *Jurnal ENTER*, 2, hal. 346-355.
- Riza, A. A., Saputro, D. R. S. (2022) "Clustering Data Numerik Menggunakan Algoritme X-Means," *Seminar Nasional Matematika, Geometri, Statistika dan Kompilasi. SeNa- MaGeStiK 2022*.
- Rozi, Z. A. (2015) *Bootstrap Design Framework*. Elex Media Komputindo.
- Sabaruddin, R. (2019) 'Modul Web Programming III', pp. 1–63.
- Saidal Faruq (2022) Konsep Box Model Pada CSS. Tersedia pada: <https://www.saidalfaruq.net/css/konsep-box-model-pada-css-l6u2av0x> (Diakses: 18 Agustus 2023).

- Saputra, A. (2019). Buku Sakti HTML, CSS & Javascript: Pemrograman Web Itu Gampang. Anak Hebat Indonesia.
- Setiawan, D. (2018) Buku Sakti Pemrograman Web: HTML, CSS, PHP, MYSQL & Javascript. Start Up.
- Sharma, A. (2018) 'Introduction to HTML ( Hyper Text Markup Language ) - A Review Paper', International Journal of Science and Research (IJSR), 7(5).
- Sidik, B. (2011) "JavaScript," Informatika Bandung : Bandung.
- Sidik, B. (2018) "Pemrograman JavaScript untuk aplikasi Web," Informatika Bandung: Bandung.
- Sitorus, S. P., Hasibuan, E. R., Rohani. (2022) "Analysis Performance of Content Delivery Network by Used Rateless Code Method," Sinkron: Jurnal dan Penelitian Teknik Informatika, 7(4), hal. 2348-2359.
- Styawantoro, I. and Komarudin, A. (2021) Pemrograman Berbasis WEB HTML, PHP 7, MySQLi, Dan Bootstrap 4. Penerbit Lakeisha.
- Suhemi, M. A. (2023) "Menambahkan Footer di HTML Menggunakan JavaScript dan Bootstrap 5," Available at: <https://blogs.powercode.id/menambahkan-footer-di-html-menggunakan-javascript-dan-bootstrap5/> (Accessed: August 15, 2023)
- Sunarya, M. H. and Bahit, M. (2020) Pemrograman Internet. Deepublish.
- Suprayogi, B., Rahmanesa, A. (2019) "Penerapan Framework Bootstrap Dalam Sistem Informasi Pendidikan SMA Negeri 1 Pacet Cianjur Jawa Barat," TEMATIK: Jurnal Teknologi Informasi dan Komunikasi, 6(2), hal. 119-127
- Surayana, T. (2020) " Operator," Available at: <https://repository.unikom.ac.id/61917/1/Bab%203%20Operator.pdf/> (Accessed: July, 12, 2023).
- Suryana, T., n.d. Membuat Tautan atau Hyperlink. [online] Available at: <https://www.google.com>.
- Sutiono. (2020) " Perbandingan Bootstrap 4 vs Bootstrap 5," Available at: [https://dosenit.com/bootstrap/perbandingan-bootstrap-4-vs-bootstrap-5/#Bootstrap\\_4\\_vs\\_5](https://dosenit.com/bootstrap/perbandingan-bootstrap-4-vs-bootstrap-5/#Bootstrap_4_vs_5) (Accessed: July 19, 2023).

- Tagger, B., Sulistiyono, M. (2017) "Modul Pemrograman Web. Dimulai dari penggunaan HTML sampai JavaScript," See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/316664126>
- Tarigan, H. G. (2020) "Rancang Bangun Detektor Kadar PH Bahan Berbasis Arduino Di Laboratorium Kebun Percobaan dan Peternakan Universitas Pembangunan Panca Budi Medan," Skripsi Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
- Tutang. (2020). HTML Bagi Pemula. Datakom Litas Buana.
- W3School, n.d. HTML Links - Hyperlinks.
- w3schools (2023) CSS Tutorial, <https://www.w3schools.com/css/>.
- w3schools (2023) HTML List and Table. Available at: [https://www.w3schools.com/html/html\\_tables.asp](https://www.w3schools.com/html/html_tables.asp) (Accessed: 7 August 2023).
- W3schools, n.d. HTML Audio. [online] Available at: [https://www.w3schools.com/html/html5\\_audio.asp](https://www.w3schools.com/html/html5_audio.asp).
- W3Schools, n.d. HTML Images. [online] Available at: [https://www.w3schools.com/html/html\\_images.asp](https://www.w3schools.com/html/html_images.asp).
- Wibowo, Alfian. (2017) "Operator Penugasan (Assignment) dan Penggunaannya Pada Pemrograman Bahasa C. Available at: <https://www.wapwibowo.com/2017/09/operator-penugasanassignment-dan.html> (Accessed: July 13, 2023).
- Wibowo, Alfian. (2017) "Operator Perbandingan (Relasional) Pada Pemrograman Bahasa C," Available at: <https://www.wapwibowo.com/2017/10/operator-perbandingan-relasional-pada.html> (Accessed: July 19, 2023).
- Wijaya, A., Feter, N. (2015) "Aplikasi Simulasi Pengurutan Data Menggunakan Algoritma Heap Short," Jurnal Pseudocode, II(2), hal. 81-88.
- Winarno, E. and Zaki, A. (2015) Desain Web Responsif Dengan Html 5 & Css 3. Elex Media Komputindo.
- Wiwesa, N. R. (2021) "User Interface dan User Experience Untuk Mengelola Kepuasan Pelanggan," Jurnal Sosial Humaniora Terapan, 3(2), hal. 17-31.

---

Yusi Ardi Binarso, Eko Adi Sarwoko, dan N.B. (2012) 'PEMBANGUNAN SISTEM INFORMASI ALUMNI BERBASIS WEB PADA PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS DIPONEGORO'.



# Biodata Penulis



**Muhammad Fairuzabadi** lahir di Bulukumba, Sulawesi Selatan tahun 1974. Alumni S1 Program Sarjana Ilmu Komputer Fakultas MIPA Universitas Kristen Immanuel (UKRIM) Yogyakarta Tahun 1998 dan S2 Program Magister Ilmu Komputer Universitas Gadjah Mada Tahun 2006. Bekerja sebagai dosen pada Program Sarjana Informatika Fakultas Sains dan Teknologi Universitas PGRI Yogyakarta. Selain itu, sebagai tenaga ahli dalam berbagai proyek pengembangan sistem informasi, master plan dan blue print TIK diintansi Pemerintah

Daerah, Kementerian dan Lembaga. Diantaranya: Masterplan Jogja Smart Province, Cetak Biru Sistem Informasi Terintegrasi PDLKWS Kementerian Kehutanan dan Lingkungan Hidup, dan Rencana Induk Sistem Pemerintahan Berbasis Elektronik (SPBE) Pemerintah Kota Metro Lampung.



**Oktoverano Hendrik Lengkong** lahir di Jakarta, 12 Oktober 1983. Menempuh pendidikan di Universitas Trisakti Magister Desain (Desain Komunikasi Visual) dan juga menyelesaikan pendidikan pada konsentrasi magister management jurusan marketing. Pada tahun 2020 di Universitas Klabat. Pada tahun 2018-2019 dipercayakan sebagai sekretaris regional Prov. Sulawesi Utara dalam Asosiasi Pilot Drone Indonesia. Aktif dalam penelitian dengan topik Multimedia, Social Media, Internet of Things, Virtual Reality,

UI/UX, Digital Marketing, Graphic Design. Dalam aktivitas penelitian pernah mendapatkan pengakuan dalam bentuk penghargaan yang diterbitkan oleh Direktorat Jenderal Pendidikan Tinggi mendapatkan penghargaan untuk penelitian terbaik. Aktif juga sebagai IT Consultant terlebih khusus dalam

perancangan web, serta Desain Grafis dan Sosial Media Analisis. Dengan aktifitas keseharian saat ini sebagai dosen tetap di Universitas Klabat – Manado.



**Ferawaty** lahir di Medan. Beliau merupakan dosen di Universitas Pelita Harapan, Prodi Informatika. Beliau pernah menulis buku dengan judul Mengelola Sumber Daya Sistem Komputer dengan Bahasa Assembly bersama penulis 2 dan juga telah menerbitkan buku Algoritma dan Pemrograman Tingkat Dasar berbahasa Python.



**Ida Wahyuni** menerima gelar Sarjana dalam bidang Teknik Informatika dari Institut Teknologi dan Bisnis Asia, Indonesia pada tahun 2014 dan gelar Master bidang Ilmu Komputer dari Universitas Brawijaya, Indonesia pada tahun 2017. Saat ini ia sedang menyelesaikan Program Doktorat di Department of Computer Science & Information Engineering (CSIE), National Central University (NCU), Taiwan dengan topik disertasi tentang "Industrial Optimization on Multistage Manufacturing Processes (MMPs) using Machine Learning".

Ia adalah dosen pada Fakultas Teknologi dan Desain, Institut Teknologi dan Bisnis Asia, Indonesia. Minat penelitiannya meliputi machine learning, artificial intelligent, deep learning, industrial optimization, dan smart manufacturing.

Telah menerbitkan 23 publikasi pada jurnal dan konferensi baik nasional maupun internasional, beberapa buku referensi, dan 2 buku tentang kuliah di Taiwan. Selain itu, juga aktif menjadi author, menulis, dan membuat konten di webiste [www.indowhiz.com](http://www.indowhiz.com) serta Instagram dan YouTube @indowhiz.

E-mail: [ida.wahyuni8@gmail.com](mailto:ida.wahyuni8@gmail.com), [idawahyuni@asia.ac.id](mailto:idawahyuni@asia.ac.id)

Instagram: @idawhyuni



**Indah Clara Sari** lahir di Pematang Siantar, pada 1 Oktober 1993. Ia tercatat sebagai lulusan Universitas Sumatera Utara. Wanita yang kerap disapa Lala ini adalah anak dari pasangan Sarianto (Ayah) dan Suriani (Ibu). Ia juga sudah menikah dengan Abdi Nugraha (Suami) dan telah memiliki seorang anak laki-laki yang bernama Alfarizi Ziandra Nugraha. Indah Clara Sari adalah seorang pengajar, Dosen tetap di Politeknik Wilmar Bisnis Indonesia. Pada Tahun 2022, Indah Clara Sari berhasil meraih penghargaan sebagai "Dosen Berprestasi" Tahun Akademik 2021/2022 Program Studi Teknologi Rekayasa

Perangkat Lunak di Politeknik Wilmar Bisnis Indonesia.



**Dr. Janner Simarmata, S.T., M.Kom.** (C.SP., C.BMC., C.DMP., C.PI., C.PKIR., C.SF., C.PDM., C.SEM., C.COM., C.SI., C.SY., C.STMI INT'L, CBPA., C.WI.)

Sarjana Teknik Informatika dari STMIK Bandung, Magister Ilmu Komputer dari Universitas Gadjah Mada (UGM) dan Doktor Pendidikan Teknologi Kejuruan (PTK) diperoleh dari Universitas Pendidikan Indonesia (UPI) Bandung bidang kajian

Blended Learning. Menulis buku sejak tahun 2005. Dosen di Pendidikan Teknologi Informatika dan Komputer (PTIK) Fakultas Teknik Universitas Negeri Medan. Sejak tahun 2005 hingga saat ini telah menulis buku sebanyak 240 judul buku.



**Setia Wardani** lahir di Banjarnegara, pada 28 September 1984. Ia tercatat sebagai lulusan Teknik Informatika IST-AKPRIND Yogyakarta pada tahun 2007 dan lulusan Magister Teknik Informatika STMIK AMIKOM Yogyakarta tahun 2011. Wanita yang kerap disapa Setia ini adalah dosen tetap pada Program Studi Informatika Universitas PGRI Yogyakarta sejak tahun 2012.





**Syarifah Fitrah Ramadhani** telah menyelesaikan Program Pendidikan Strata Satu (S1) jurusan Pendidikan Teknik Informatika dan Komputer di Universitas Negeri Makassar pada tahun 2015 dan Melanjutkan ke Program Pendidikan Strata Dua (S2) jurusan Sistem Komputer di STMIK Handayani Makassar pada tahun 2017. Penulis memulai karier di dunia pendidikan pada tahun 2018 sebagai Dosen tetap Program Studi Sistem Informasi pada Kampus STMIK Adhi Guna Palu.

Telah Mengampu mata kuliah Riset Operasi, E-Business, Manajemen Proyek Perangkat Lunak, Teori Bahasa dan Automata, Sistem Pendukung Keputusan, Administrasi Sistem Operasi, Analisis Perancangan Sistem, Artificial Intelligence (Kecerdasan Buatan), dan Aplikasi Komputer.

Selama ini terlibat aktif sebagai Dosen Pembimbing dan Penguji Mahasiswa Program Studi Sistem Informasi dan Teknik Informatika.

Keahlian penulis adalah menguasai software Engineering dan Mobile Computing, dan aktif dalam melakukan riset yang kemudian di publikasikan dalam bentuk sebuah jurnal.

E-Mail: [syarifahfitrah@gmail.com](mailto:syarifahfitrah@gmail.com) , [syarifahfitrahramadhani@gmail.com](mailto:syarifahfitrahramadhani@gmail.com)



**Reymon Rotikan** lahir pada tanggal 19 September 1987 di Parigi, Sulawesi Tengah. Lulus dari Universitas Klabat program studi Sistem Informasi dan melanjutkan pendidikan Magister di Yuan Ze University, Taiwan R.O.C di program studi Computer Science. Saat ini mengajar di Fakultas Ilmu Komputer, Universitas Klabat.



**Stenly Ibrahim Adam** lahir di Serang, pada 15 September 1987. Menyelesaikan program Master di National Taiwan University of Science and Technolgy (NTUST), Taipei Taiwan. Aktif dalam penelitian dengan topik mengenai teknologi mobile dan web. Saat ini sebagai dosen di Fakultas Ilmu Komputer Universitas Klabat, Airmadidi Minahasa Utara. Mata kuliah yang sering diajarkan adalah Mobile Programming, Front-end Web Development dan Back-end Web Development.



**Pujianti Wahyuningsih.** Menyelesaikan strata satu di jurusan sistem informasi di STMIK AKBA pada tahun 2015 dan strata dua jurusan ilmu komputer di STMIK Handayani Makassar pada tahun 2017.

Memulai jenjang karier didunia pendidikan sebagai dosen dengan diterbitkan surat keputusan pengangkatan dosen tetap pada Yayasan Pendidikan Handayani di STMIK Handayani pada tanggal 12 November 2015, menjadi kepala Biro Administrasi Akademis Kemahasiswaan (BAAK) pada tahun 2016-2019 di STMIK handayani Makassar, Ketua Jurusan Teknik Informatika pada tahun 2022 sampai sekarang di Universitas Handayani Makassar. Penulis aktif melakukan research setiap tahun yang kemudian mempublish hasil penelitian dalam bentuk jurnal yang terkait dalam bidang ilmu komputer.



**Billy Eden William Asrul.** Pengajar pada Universitas Handayani Makassar, Fakultas Ilmu Komputer, Program Studi Teknik Informatika. Sedang Menempuh Pendidikan S3 di Universitas Hasanuddin dengan Topik Penelitian Learning analytics. Sebelumnya mengikuti Pendidikan Program S1 di STMIK Handayani dan S2 di Universitas Hasanuddin.

Mengampu mata kuliah Pemrograman Berorientasi Obyek, Pemrograman Aplikasi Mobile, Logika Algoritma dan Pemrograman Visual. Selama ini terlibat aktif sebagai Pengurus Ikatan Ahli Informatika Indonesia (IAII) dan Sebagai Pengurus Badan Kejuruan Informatika (BKI) Persatuan Insinyur Indonesia (PII).

Telah menulis Buku referensi dengan judul Sistem Pendukung Keputusan Dengan Metode Double Exponential Smoothing Penerbit Cipta Media Nusantara.

E-mail: bewagabriel.bg@gmail.com , billy@handayani.ac.id



**Suwatri Jura, S.Kom., M.Kom** Lulus program Studi Strata satu pada Program Studi Teknik Informatika di STMIK Handayani dan melanjutkan program magister pada bidang ilmu komputer di STMIK Handayani. Saat ini penulis sebagai Dosen Tetap Yayasan pada Universitas Handayani Makassar dengan mengajarkan beberapa mata kuliah bidang Teknik Informatika.

# DESAIN WEB

## Menguasai **HTML**, **CSS**, **JavaScript**, dan **Bootstrap** untuk Pemula

Selamat datang dalam buku ini yang akan memperkenalkan Anda pada dunia desain web menggunakan HTML, CSS, JavaScript, dan Bootstrap. Bagi pemula yang ingin menguasai dasar-dasar pembuatan halaman web yang menarik dan responsif, buku ini merupakan panduan yang tepat.

Buku ini berisi:

BAGIAN 1: HTML (Hypertext Markup Language):

Bab 1 Pengenalan Desain Web dan HTML

Bab 2 Daftar dan Tabel

Bab 3 Hyperlink, Gambar dan Media

Bab 4 Formulir dan Elemen Input

BAGIAN 2: CSS (Cascading Style Sheets):

Bab 5 Pengenalan CSS

Bab 6 Model Kotak dan Tata Letak

BAGIAN 3: JavaScript:

Bab 7 Pengenalan Java Script

Bab 8 Variabel, Tipe Data, dan Operator

Bab 9 Aliran Kontrol dan Perulangan

Bab 10 Fungsi dan Objek

BAGIAN 4 Bootstrap:

Bab 11 Pengenalan Bootstrap

Bab 12 Sistem Grid Bootstrap

Bab 13 Komponen Bootstrap



YAYASAN KITA MENULIS  
press@kitamenulis.id  
www.kitamenulis.id

