

# SISTEM OPERASI

Marti Widya Sari



2018

# **SISTEM OPERASI**

**Oleh**  
**Marti Widya Sari**

**Maret, 2018**

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan karunia serta rahmatNya sehingga penulis dapat menyelesaikan buku ajar mata kuliah Sistem Operasi ini dengan baik. Maksud dan tujuan dari pembuatan buku ajar ini adalah untuk memenuhi keperluan proses belajar mengajar mata kuliah Sistem Operasi serta untuk pengayaan materi bagi mahasiswa. Buku ini berisi tentang materi-materi mata kuliah Sistem Operasi selama satu semester, serta soal-soal latihan yang dapat mendukung proses pembelajaran.

Mengingat masih terbatasnya pengetahuan, kemampuan dan pengalaman, penulis menyadari masih banyak kekurangan dalam penyusunan buku ajar ini. Untuk itu, saran dan kritik yang bersifat membangun sangat Penulis harapkan demi perbaikan penulisan buku ajar ini. Semoga buku ajar ini dapat memberikan manfaat bagi kemajuan ilmu pengetahuan.

Yogyakarta, Maret 2018

Penulis

## DAFTAR ISI

HALAMAN JUDUL	
KATA PENGANTAR .....	ii
Bab 1 .....	1
Pengenalan Sistem Operasi.....	1
A. Pengertian Sistem Operasi .....	1
B. Komponen-komponen Komputer .....	1
C. Layanan Sistem Operasi.....	7
D. Jenis-jenis Sistem Operasi .....	8
Bab 2 .....	10
Konsep Dasar dan Sejarah Sistem Operasi .....	10
A. Konsep Dasar .....	10
B. Tujuan Sistem Operasi:.....	10
C. Jenis Sistem Operasi .....	10
D. Perkembangan Generasi Sistem Operasi .....	11
E. Faktor Sistem Operasi .....	12
Bab 3 .....	14
Skema Dasar dan Arsitektur Komputer .....	14
A. Perangkat Keras ( <i>Hardware</i> ) .....	14
B. Perangkat Lunak ( <i>Software</i> ) .....	16
Bab 4 .....	17
Struktur Penyimpanan Massal .....	17
A. Magnetic Disc .....	17
B. Teknologi Hardisk .....	21
C. Disk Attachment .....	22
Bab 5 .....	26
Pengaksesan Lintas Disk.....	26
A. Waktu Akses .....	26

B. Algoritma Pengaksesan Lintas Disk .....	26
Bab 6 .....	31
Proses .....	31
A. Konsep Proses .....	31
B. Tingkatan Proses .....	32
Bab 7 .....	35
Penjadwalan Proses.....	35
A. Konsep Penjadwalan Proses.....	35
B. Tujuan Penjadwalan Proses .....	35
C. Tipe Penjadwalan.....	35
Bab 8 .....	38
Pengenalan Distro Linux.....	38
A. Sejarah Perkembangan Linux .....	38
B. Distro Linux .....	38
Bab 9 .....	42
Pembuatan Distro Linux Slax .....	42
A. Pembuatan Slax.....	42
B. Tools untuk Pembuatan Slax.....	42
Bab 10 .....	43
Modifikasi Distro Linux .....	43
Bab 11 .....	44
Manajemen Memori.....	44
A. Latar Belakang .....	44
B. Address Binding.....	46
C. Ruang Alamat Logika dan Ruang Alamat Fisik .....	47
D. Dynamic Linking dan Shared Libraries .....	48
E. Swapping.....	49
F. Alokasi Memori .....	50
G. Fragmentasi.....	51

H. Segmentasi .....	52
I. Paging .....	52
J. Memory Manager.....	54
K. Fungsi manajemen memori.....	54
L. Strategi Alokasi Memori.....	55
Bab 12 .....	56
Manajemen Input/Output (I/O) .....	56
A. Pooling .....	56
B. Interupsi .....	56
C. DMA .....	57
A. Konsep Deadlock.....	60
B. Resource Allocation Graph.....	63
C. Metode Penanganan Deadlock.....	65
D. Deteksi Deadlock.....	65
E. Perbaikan dari <i>Deadlock</i> .....	66
Bab 14.....	68
Konkurensi dan Keamanan .....	68
A. Konkurensi.....	68
B. Keamanan .....	68

# Bab 1

## Pengenalan Sistem Operasi

### A. Pengertian Sistem Operasi

Pengertian sistem operasi secara umum adalah:

1. Sebagai Sebuah program yang mengatur hardware, dengan menyediakan landasan untuk aplikasi yang berada di atasnya.
2. Bertindak Sebagai penghubung antara user dengan hardware.
3. Bertugas untuk mengendalikan & mengkoordinasi penggunaan hardware untuk berbagai program aplikasi bagi bermacam-macam user.

Pengertian sistem operasi ditinjau dari 3 sudut pandang yang berbeda adalah:

1. Sudut pandang pengguna  
Sistem operasi adalah *alat untuk mempermudah penggunaan computer*. Sistem operasi seharusnya dirancang dengan mengutamakan kemudahan pengguna. dibandingkan menggunakan kinerja ataupun utilitas sumber daya. Sebaliknya dalam lingkungan multiuser, sistem operasi dapat dipandang sebagai alat untuk memaksimal.
2. Sudut pandang sistem  
Sistem operasi adalah sebagai alat yang menempatkan sumber daya secara efisien. Sistem operasi merupakan manajer bagi sumber daya yang menangani konflik permintaan sumber daya secara efisien.
3. Sudut pandang tujuan  
Sistem operasi adalah sebagai alat yang membuat komputer lebih nyaman digunakan untuk menjalankan aplikasi dan menyelesaikan masalah user.

### B. Komponen-komponen Komputer

#### a) *Hardware* (Perangkat Keras)

*Hardware* istilah lainnya perangkat keras. Suatu benda yang dapat dipegang dan dapat dilihat secara kasat mata. Tubuh manusia adalah termasuk kategori benda yang dapat dilihat dan dipegang oleh karena itu dapat disebut sebagai *hardware*. Istilah *hardware* berlaku untuk semua hal yang ada di sekitar kita.



**Gambar 1.1 Hardware Komputer**

Dalam hal kaitannya dengan komputer yang notabene alat elektronik yang berguna untuk membantu kinerja manusia, *hardware* adalah suatu perangkat fisik atau peripheral yang saling terkait dalam mengolah input dan menghasilkan output tertentu dalam satu kesatuan komputer.

Dalam membentuk perangkat komputer diperlukan suatu perangkat fisik agar dapat menjalankan segala perintah yang dimasukkan dalam bentuk bahasa pemrograman. Dalam kategori *hardware* komputer akan selalu terdiri dari alat elektronik yang tergabung dalam papan sirkuit. Papan sirkuit ini memiliki jalur untuk menghubungkan komponen elektronik yang biasanya terdiri dari kumpulan transistor, resistor, elko dsb.

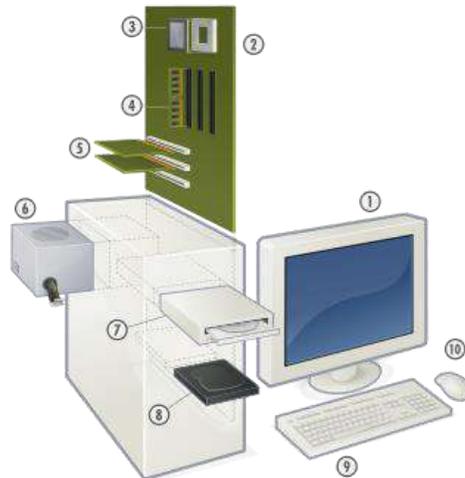
Sesuai keadaan tersebut menyatakan bahwa komputer merupakan salah satu bentuk alat elektronik yang tersusun atas gabungan komponen elektronik. Dengan gabungan tersebut pasti memerlukan sesuatu agar semua fungsi dari bahan tersebut berjalan semestinya. Sesuatu tersebut adalah adanya aliran listrik, karena pada dasarnya setiap alat elektronik pasti memerlukan aliran tersebut.

Dari kenyataan diatas bahwa jiwa atau komponen utama dalam sebuah *hardware* adalah listrik. Maka dalam ilmu komputer bahkan seluruh alat cerdas elektronik menggunakan konsep nol (0) dan satu (1). Konsep tersebut mewakili nol berarti tidak adanya aliran listrik dan satu berarti memiliki atau terjadi aliran listrik sebesar 5 volt. Dengan adanya konsep tersebut dalam ilmu komputer dipelajari dalam konsep bilangan biner. Kenapa dikatakan biner, karena memang semua perhitungan menggunakan dua jenis bilangan yaitu bilangan nol dan satu.

Dengan konsep tersebut pula proses untuk mengenali dan menjalankan antar *hardware* dilakukan. Proses tersebut akan dipelajari lebih lanjut pada bagian analisa bilangan. Komputer merupakan perangkat cerdas yang dapat mengalirkan perintah untuk menjalankan, mengontrol dan memantau perkembangan *hardware* yang tersambung didalamnya. Ketika suatu instruksi atau perintah diberikan maka *hardware* akan merespon berdasarkan perintah tersebut. Perintah tersebut akan direspon dalam bentuk nol dan satu, perintah tersebut diterjemahkan dengan ada listrik dan tidak adanya listrik.

Dalam suatu *hardware* komputer, biasanya komputer jinjing (notebook, netbook) atau pun PC (Personal Computer) terdapat istilah lain yaitu firmware. Firmware ini adalah suatu *hardware* yang didalamnya sudah terdapat instruksi yang tertanam didalamnya. Dalam *hardware* komputer salah satu contohnya adalah motherboard, merupakan perangkat utama dalam komputer. Tempat untuk menaruh dan mengoperasikan *hardware* keseluruhan dan pengoperasiannya melalui *hardware* ini.

Mengenai nama komponen komputer harus diketahui oleh user, untuk menghindari dari salah pembelian hardware maupun penipuan jenis alat. Karena banyak kasus penipuan dalam pembelian peralatan komputer, apabila terjadi demikian akan membahayakan komponen lainnya dalam PC. Karena komponen yang tidak sesuai akan menghambat aliran listrik dan data yang ada di dalamnya, sebagai akibatnya akan merusak komponen komputer lain.



**Gambar 1.2 Personal Computer (PC)**

Keterangan Gambar :

#### 1. Monitor

Monitor adalah suatu alat elektronik yang digunakan untuk menampilkan hasil keluaran secara grafik maupun video. Monitor terdiri dari gabungan beberapa pixel warna untuk menampilkan suatu yang memiliki warna.

Alat tersebut biasanya digunakan untuk analisa dan proses data, terkadang digunakan untuk hiburan atau transfer data seperti halnya video, film, dan lain sebagainya, sebagai alat keluaran. Monitor terdiri dari media tampilan, sirkuit. Media tampilan biasa menggunakan satuan pixel (satuan warna).

Satuan yang dipakai pada monitor beragam. Satuan tersebut bervariasi karena permintaan oleh user. Satuan yang sering dipakai adalah satuan inchi (“). Biasanya dalam penggunaan satuan itu diukur berdasarkan jarak satu sisi layar dengan sisi lain.

Jarak tersebut berbanding 4:3, misalnya 1024 maka layar tersebut memiliki jarak vertical 768 pixel. Terdapat ukuran sekitar 10”, 14”, 16 “ dan seterusnya. Terdapat jenis monitor tersebar antara lain CRT, LCD, PLASMA, semua memiliki keunggulan masing-masing.

## 2. Motherboard

Motherboard atau papan induk atau dikenal dengan *main board logic board* adalah suatu kumpulan dari alat elektronik yang berfungsi untuk menghubungkan komponen utama dalam sebuah computer. Alat tersebut merupakan pusat PCB (*Printed Circuit Board*) yang berfungsi mengkoneksikan peripheral yang ada pada computer modern.

Main board memiliki komponen untuk mendukung kinerja CPU dan kinerja software. Chipset merupakan salah satunya yang mendukung kinerja didalamnya. Salah satu komponen main board terdapat fungsi untuk mengenali dan mengatur pemakaian hardware yaitu BIOS (*Basic Input Output System*).

## 3. CPU

*Central Processing Unit* merupakan pusat dari computer yang berisi intruksi untuk menjalankan perintah dan mengeksekusinya serta mengeluarkan output tertentu. Perintah tersebut untuk melakukan operasi aritmatika, logika, serta operasi input/output dalam system.

## 4. RAM

*Random Access Memory* merupakan memori tempat aliran data, sekaligus tempat simpanan data komputer. Dengan maksud random diatas data yang tersimpan dapat diambil atau diakses sewaktu-waktu dalam tempat penyimpanan yang sembarang pula. Keadaan RAM yang seperti itu merupakan bentuk yang diambil dari beberapa IC penyusunnya. Sifat dari IC tersebut *volatile* alias data akan hilang dari RAM setelah tidak adanya listrik.

DRAM merupakan salah satu contoh dari RAM. Memory tersebut merupakan contoh kinerja memory yang notabene merupakan sebagai jalan masuk-keluar data dari dan ke prosesor atau CPU. Karena tanpa adanya RAM ini maka system operasi tidak akan berjalan karena tidak ada jalan atau media simpanan sementara menuju pemroses perintah utama.

## 5. Piringan kartu tambahan (card)

Merupakan alat atau *tools* tambahan yang dipakai untuk membantu memaksimalkan kinerja computer. Bentuk dari expansion card adalah berbentuk kotak dengan tempat tembaga untuk transfer data yang ditancapkan pada motherboard. Biasanya card berbentuk slot, sebagai contoh card yang dipakai untuk membantu memaksimalkan tampilan gambar pada monitor disebut VGA Card.

## 6. Power Supply

Power Supply merupakan alat elektronik yang terdiri dari komponen utama dynamo untuk distribusi listrik. Distribusi yang dilakukan sangat perlu mengingat daya yang diterima komponen dalam computer minimal 5 volt maksimal 12 volt. Sedangkan pada listrik standar adalah dengan tegangan 220 volt, apabila tidak ada pemampatan listrik dan pendistribusian akan terjadi kelebihan daya dan terbakar. Biasanya alat ini dipasang pada bagian belakang untuk mendekatkan alat pada sumber listrik. Dengan

daya masuk 220 menjadi terdistribusi 5 volt, alat ini merupakan jenis top-down. Maksud dari top-down mampu merubah tegangan besar menjadi kecil.

#### 7. Optical disc drive

Sebuah disc drive yang menggunakan sinar lase atau gelombang elektromagnetik untuk membaca atau menulis kedalam disc dari maupun ke optical disc. Dalam pembacaan disc tersebut terdapat beberapa istilah yaitu reader atau recordable. Sedangkan proses penulisan biasa dikatakan sebagai proses burning atau penulis. Karenan pada dasarnya penulisan terhadap compact disc merupakan penembakan sinar laser sehingga terjadi lubang seperti terbakar pada piringan disc-nya. Compact disc, DVDs, HD, blue ray merupakan salah satu contoh tipe disc saat ini.

#### 8. Hard disk drive

HDD merupakan alat yang dipakai untuk media simpanan dengan criteria non volatile atau data tidak akan hilang meski aliran listrik sudah tidak ada. Saat ini teknologi hard disk yang berkembang adalah SATA selain jenis SAS. Didalam HDD sebenarnya terdapat kumpulan piringan seperti yang terdapat pada disc, piringan ini dipakai untuk menyimpan data dengan cara menembakkan sinar laser dalam ruangan hard disk yang hampa udara. Mengenai transfer data, hard disk memberikan transfer melalui satuan rotasi per menit atau rpm. Dengan satuan tersebut tersebar 4200 rpm, 5200 rpm, 5400 rpm, atau 7200 biasanya kategori ini merupakan hard disk mobile atau hard disk eksternal. Kecepatan transfer paling tinggi adalah 10000 rpm artinya hard disk tersebut mampu membaca piringannya sebesar 4200 putaran per menit apabila satuannya 4200 rpm.

#### 9. Keyboard

Keyboard merupakan alat salah satu jenis alat input berisi berbagai karakter yang dipakai untuk menginputkan segala perintah dari user kedalam system. Dalam dunia computer biasanya keyboard dipakai untuk mengetikkan karakter dalam pengerjaan word, tetapi dapat juga memberikan sebuah perintah / command seperti pada windows, ctrl+alt+del.

#### 10. Mouse

Jenis alat input untuk mengaktifkan pada titik tertentu atau untuk menunjuk pada posisi tertentu. Mouse bentuknya kecil dengan struktur yang hampir mirip seperti tikus, itulah kenapa disebut mouse. Mouse yang beredar di pasar ada terdapat jenis scroll, dan optical. Dalam transfer data mouse menggunakan kabel dengan kabel penghubung berbentuk lingkaran atau dikatakan PS2. Selain bentuk tersebut terdapat penghubung lain yaitu usb mouse dan wireless mouse.

## **b) Software**

Software komputer adalah suatu kumpulan instruksi atau program yang dimaksudkan untuk mengerjakan perintah tertentu dalam menjalankan suatu fungsi interaksi antara user dengan hardware. Istilah lain software adalah perangkat lunak, dalam ilmu kemanusiaan. Dapat dikatakan bahwa manusia memiliki software yaitu berupa jiwa manusia, yang bisa berpikir dan memerintah atau mengkoordinasikan segala fungsi yang dimiliki manusia.

Software merupakan gabungan dari perintah yang berisikan kode dan manipulasi data bilangan. Data bilangan yang dipakai adalah nol (0) dan satu (1), tetapi dalam kehidupan informatika dialihkan kedalam bahasa yang mudah dimengerti manusia. Bahasa tersebut dapat dikatakan bahasa tingkat tinggi, yaitu bahasa yang mendekati bahasa manusia. Bahasa tersebut seperti bahasa pemrograman java, C, C++, asp, dll.

Software terbentuk untuk memfungsikan hardware. Dalam artian fungsi adalah menjembatani antara hardware dengan manusia atau brainware. Karena dalam dunia computer terdapat dunia hardware, dunia software, dan dunia brainware. Masing-masing memiliki peranan dan kecakapan dalam bidangnya sehingga, ketiga factor tersebut saling memanfaatkan.

### **Jenis Software**

Pengertian software sudah kita pelajari untuk selanjutnya, software tersebut dibagi menjadi 2 jenis yaitu software system dan software aplikasi. Software system adalah suatu software yang dipakai untuk mengkoneksikan, mengkoordinasi operasi hardware dan software aplikasi, serta mengatur segala transaksi didalamnya.

Dari pengertian diatas dapat diketahui bahwa software system merupakan system operasi. Itu dapat diketahui dari fungsi system operasi yaitu mengkoordinasi dan mengkoneksikan serta mengatur software aplikasi dan hardware yang menanca pada motherboard.

Sistem operasi sekarang ini memiliki berbagai versi yang muncul. Versi tersebut memiliki kriteria dan perlakuan yang berbeda terhadap hardware dan brainware. Seperti halnya LINUX dan WINDOWS. Kedua nama tersebut merupakan system operasi yang memiliki perbedaan di banyak tempat, mulai dari tampilan (interface), tingkat pemakai, jenis pemakaian (berbayar atau tidak), maupun dari segi yang lain. Sebagai contoh system operasi yang eksis sekarang ini adalah GNU/Linux, windows, Apple/Mac, Free BSD, Solaris, palm, symbian, android dan sebagainya.

Jenis sistem operasi saat ini sebagai berikut : Sistem operasi-sistem operasi utama yang digunakan komputer sistem umum (termasuk PC, komputer personal) terbagi menjadi 3 kelompok besar:

1. Keluarga [Microsoft Windows](#) - yang antara lain terdiri dari Windows Desktop Environment (versi 1.x hingga versi 3.x), Windows 9x (Windows 95, 98, dan Windows ME), dan Windows NT (Windows NT 3.x, Windows NT 4.0,

Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows 7 (Seven) yang dirilis pada tahun 2009, dan Windows Orient yang akan dirilis pada tahun 2014)).

2. Keluarga Unix yang menggunakan antarmuka sistem operasi POSIX, seperti SCO UNIX, keluarga BSD (*Berkeley Software Distribution*), GNU/Linux, MacOS/X (berbasis kernel BSD yang dimodifikasi, dan dikenal dengan nama *Darwin*) dan GNU/Hurd.
3. Mac OS, adalah sistem operasi untuk komputer keluaran Apple yang biasa disebut *Mac* atau *Macintosh*. Sistem operasi yang terbaru adalah Mac OS X versi 10.4 (*Tiger*). Awal tahun 2007 direncanakan peluncuran versi 10.5 (*Leopard*).

Sedangkan komputer Mainframe, dan Super komputer menggunakan banyak sekali sistem operasi yang berbeda-beda, umumnya merupakan turunan dari sistem operasi UNIX yang dikembangkan oleh vendor seperti IBM AIX, HP/UX, dll.

#### c) **USER**

User adalah manusia yang menjalankan program aplikasi atau komputer lain, biasa juga disebut dengan pengguna komputer.

### **C. Layanan Sistem Operasi**

#### **1. Pembuatan program**

Sistem operasi menyediakan berbagai fasilitas yang membantu programmer dalam membuat program seperti editor. Walaupun bukan bagian dari sistem operasi, tapi layanan ini diakses melalui sistem operasi

#### **2. Eksekusi program**

Sistem harus bisa me-load program ke memori dan menjalankan program tsb. Program harus bisa menghentikan pengekseskuan baik secara normal maupun dalam keadaan error.

#### **3. Operasi I / O**

Program sedang dijalankan kadang kala membutuhkan I / O. untuk efisiensi dan keamanan, pengguna biasanya tidak bisa mengatur piranti I / O secara langsung.

#### **4. Manipulasi sistem berkas**

Program harus membaca dan menulis berkas, kadangkala juga harus membuat dan menghapus berkas

#### **5. Komunikasi**

Kadangkala sebuah proses memerlukan informasi dari proses yang lain. Ada 2 cara umum komunikasi dilakukan : yaitu antara proses dalam satu komputer, atau antara proses yang berada dalam komputer yang berbeda. Tetapi dihubungkan oleh jaringan

komputer. Komunikasi dapat dilakukan dengan share-memori / message passing, yakni sejumlah informasi dipindahkan antara proses oleh sistem operasi.

## 6. Deteksi error

Sistem operasi harus selalu waspada terhadap kemungkinan error. Error dapat terjadi di CPU dan memori perangkat keras, I / O dan didalam program yang dijalankan user. Misalnya; dengan jalan menghentikan jalannya program, mencoba kembali melakukan operasi yang dijalankan atau melaporkan kesalahan yang terjadi agar pengguna dapat mengambil langkah selanjutnya.

## D. Jenis-jenis Sistem Operasi

Berikut merupakan jenis-jenis sistem operasi yang banyak digunakan oleh pengguna, yaitu :

1. DOS (*Disk Operating Sistem*)
2. WINDOWS
3. MACINTOSH
4. OS/2 (*Operating System/2*)
5. UNIX

### 1) DOS (*Disk Operating Sistem*)

DOS merupakan system operasi yang pertama kali digunakan hingga saat ini. DOS dibagi menjadi 3, yakni ;

- DR-DOS  
Dibuat oleh perusahaan digital research pada tahun 1988 dari Hasil transformasi CP/M (control program/monitor). *DigitalResearch didirikan oleh Gory Kildall (1942-1994)*. Sistem operasi Ini digunakan sampai pada komputer pentium.
- PC-DOS  
*Dirancang dan dibuat oleh perusahaan microsoft corp* namun dengan dikeluarkannya sistem operasi windows, sistem operasi ini sudah jarang / tidak digunakan lagi saat ini.
- MS-DOS  
*Dirancang dan dibuat oleh perusahaan microsft corp*. sistem operasi ini digunakan hingga saat ini dan terus dikembangkan versinya sesuai dengan kebutuhan.

### 2) WINDOWS

Dirancang oleh Microsoft Corp. pada awalnya dibuat nya windows bukan merupakan sistem operasi karena windows versi 3.x untuk menjalankannya masih bergantung pada sistem operasi DOS.

Tujuan awalnya hanya untuk mengatasi masalah yang dihadapi penggunaan computer dalam melakukan:

- MULTITASKING : penggunaan program computer secara bersamaan.
- MULTIUSER : pengaturan pemakai komputer oleh banyak pemakai.

### 3) MACINTOSH

*Macintosh pertama dikeluarkan oleh perusahaan APPLE CORP.* Berbeda dengan sistem operasi yang lain, sistem ini hanya ditujukan untuk komputer berbasis grafis dengan istilah GUI (*Graphical User Interface*).

Penggunaanya sangat spesifik karena tidak dapat digunakan pada komputer lain (PC), tetapi hanya dapat digunakan pada komputer macintosh saja, karena pada komputer macintosh tidak dikenal adanya *extention (.exe)* file seperti pada sistem Operasi DOS dan WINDOS.

### 4) OS / 2 (operating sistem / 2 )

OS/2 dibuat pertama kali dengan tujuan untuk menggantikan sistem operasi DOS. Namun sistem ini jarang digunakan karena harganya lebih mahal dan pengembangannya lebih lambat bila dibandingkan dengan DOS.

### 5) UNIX

UNIX pertama kali digunakan untuk komputer mini dan mainframe, tetapi kini sudah dapat digunakan oleh komputer mikro. Diciptakan oleh ahli riset BELL LABORATORIES (amerika). Tetapi kurang diminati oleh pengguna komputer mikro karena sulitnya menyatukan standard sistem. Perkembangannya tidak sepesat DOS, namun kini UNIX telah mengeluarkan versi terbarunya dengan nama LINUX. Untuk mempromosikan dan memperkenalkan sistem operasi LINUX. Pada Masyarakat, maka perusahaan pembuatnya memasarkan secara gratis.

### SOAL LATIHAN :

1. Jelaskan tentang jenis-jenis sistem operasi!
2. Jelaskan perbedaan sistem operasi Windows dan Linux!
3. Jelaskan tentang sistem operasi yang berbasis GUI!
4. Jelaskan tentang istilah multitasking!
5. Apa kelebihan dan kekurangan dari sistem operasi berbasis text?

## **Bab 2**

### **Konsep Dasar dan Sejarah Sistem Operasi**

#### **A. Konsep Dasar**

Konsep dasar sistem operasi ini di bagi menjadi tiga pengertian sistem operasi, yaitu:

- Sebagai Pelaksana Perintah: Memberi abstraksi mesin tingkat tinggi yang lebih sederhana menyembunyikan kerumitan perangkat keras. System operasi menyediakan system call (*API=Application Programming Interface*) yang berfungsi menghindarkan kompleksitas pemrograman dengan member sekumpulan instruksi yang mudah digunakan.
- Sebagai pelaksana tataolah aplikasi: Program aplikasi dijalankan di atas system operasi yang bertujuan untuk memanfaatkan dan mengendalikan sumber daya system kmputer secara benar,efisien,dan mudah dengan meminta layanan system operasi.
- Sebagai pengelola sumber daya (Fisik & Abstrak)

#### **B. Tujuan Sistem Operasi:**

- 1) Sistem operasi membuat komputer menjadi lebih mudah dan nyaman untuk digunakan.
- 2) Sistem operasi memungkinkan sumber daya sistem komputer untuk digunakan secara efisien.
- 3) Sistem operasi harus disusun sedemikian rupa sehingga memungkinkan pengembangan yang efektif, pengujian, dan penerapan fungsi baru tanpa mengganggu layanan yang sudah ada.

Fungsi sistem operasi yaitu membentuk/mengelola sistem file, menjalankan program dan mengatur penggunaan alat-alat yang berhubungan dengan komputer.

#### **C. Jenis Sistem Operasi**

Beberapa Jenis-jenis sistem operasi:

- a) Berdasarkan Interface  
Text Base shell vs GUI
- b) Berdasarkan uses (peruntukan)  
Stand alone vs Networking

Contoh-contoh sistem operasi:

- DOS
- OS/2
- Machintos
- Windows, Linux, dll

#### D. Perkembangan Generasi Sistem Operasi

**Sejarah-sejarah Sistem Operasi terdiri dari:**

- a. Operasi open shop => operasi langsung melalui *switch electric* pada komputer
- b. Operasi driven shop => admin mengelompokkan job berdasarkan jenis program dan dieksekusi sesuai jenis program yang dicoding melalui *punched card* secara sequence
- c. Operasi off-line => job yang akan dieksekusi disimpan dalam media offline sebelum eksekusi oleh komputer
- d. Operasi penampung (*buffer operation*)
- e. Operasi spool => bertindak sebagai buffer dan mampu menerima proses meskipun belum dikerjakan co: printer
- f. Operasi multitataolah tumpukan (*Batch multiprogramming operation*)
- g. Operasi berbagi waktu (*time-sharing*)
- h. Operasi olahan segera (*Real time programming operation*)

**Generasi system operasi:**

- Generasi Pertama (1945-1955): Generasi tanpa system operasi, komponen utama => tabung hampa, operasi secara manual melalui plugsboards, hanya bias menghitung (+, - dan \*)
- Generasi kedua (1955-1965): berbentuk tumpukan (batch system) komponen utama => transistor, input memakai punch card, sist operasi pertama => multics
- Generasi ketiga (1965-1980) dengan cirri-ciri:
  - a. Multi Programming => satu komp mengerjakan banyak program yang ada dalam memori utama
  - b. Kemandirian alat (device independency) => masing-masing alat memiliki device driver sendiri-sendiri co : printer

- c. Berbagi waktu (time sharing) => menjalankan banyak proses dalam satu waktu
- d. Spooling => mampu menerima proses meskipun belum dikerjakan dan bertindak sebagai buffer komponen utama => IC (Integrated Circuit)
- Generasi keempat (198X-199X) : Sistem tujuan umum (general purpose & multimodus)
  - a. Real-time application
  - b. Network Operation Sistem
  - c. Distributed Operation Sistem
  - d. Mesin semu (Virtual machin)
  - e. Distribusi data

Aplikasi komputer seperti spreadsheet, word processor, database atau grafis berkembang pesat, microprocessor berbasis RISC tuk PC mulai diperkenalkan.

Kelas sistem operasi :

- a. Kelas 1, pemakai tunggal
- b. Kelas 2, operasi berbentuk tumpukan
- c. Kelas 3, operasi olahan segera (olahan segera)
- d. Kelas 4, operasi multi proses
- e. Kelas 5, operasi berbagi waktu dan multi programming
- f. Kelas 6, operasi tersebar

## E. Faktor Sistem Operasi

Faktor sistem operasi adalah sebagai berikut.

- Faktor prosessor => *single/multi/processor*
- Faktor pemakai => *single user/multi user*
- Faktor waktu kerja => *offline/online*
- Faktor modus pekerjaan => *batch/real time*

Cakupan system operasi meliputi dari :

- Cakupan pengelolaan resources
- Cakupan struktur sistem operasi
- Cakupan sumber daya semu (file/program)
- Cakupan hubungan manusia dengan sistem operasi

Sistem Operasi mengatur pemakaian sumber daya tersebut dengan cara:

1. Pengaturan Memori, dengan cara:

- Mengawasi bagian-bagian memori, lokasinya statusnya, besarnya, bila terpakai siapa pemakainya
  - Menetapkan kebijaksanaan alokasi
  - Siapa yang berhak mendapatkan bagian memori beserta lokasinya
  - Memperbaharui informasi tentang status bagian memori
2. Pengaturan processor, dengan cara:
    - Mengawasi status prosesor
    - Menentukan proses yang mana dari beberapa proses yang sedang menanti atau menggunakan prosesor
    - Menyerahkan penggunaan prosesor dengan cara mengisi register-register yang perlu diisi
    - Menarik kembali pemakaian processor bila sudah tidak dipergunakan
  3. Pengaturan peralatan I/O
    - Mengawasi status peralatan I/O beserta kelengkapannya (channel)
    - Mengatur cara dan jadwal pemakaian peralatan I/O
    - Menyerahkan pemakaian peralatan I/O
    - Menerima kembali peralatan I/O bila sudah tidak dipergunakan
  4. Pengaturan Informasi (sistem file)
    - Mengawasi berbagai informasi, yaitu mengenal statusnya, lokasi, penggunaannya, dan sekuritinya.
    - Menetapkan proses-proses yang boleh mendapatkan informasi tertentu
    - Melepaskan atau memberikan informasi
    - Menarik kembali dan menyimpan informasi

### SOAL LATIHAN :

1. Pada generasi ke berapa aplikasi sistem operasi berkembang dengan pesat ?
2. Bagaimana perkembangan dari masing-masing generasi sistem operasi ?
3. Jelaskan perbedaan antara sistem operasi *text base shell* dan GUI !
4. Jelaskan perbedaan antara sistem operasi *stand alone* dan *networking* !
5. Apa kelebihan dan kekurangan dari sistem operasi stand alone dan *networking*?

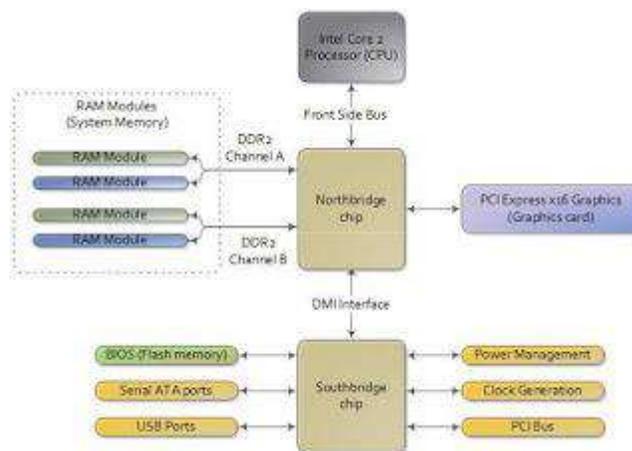
## Bab 3

### Skema Dasar dan Arsitektur Komputer

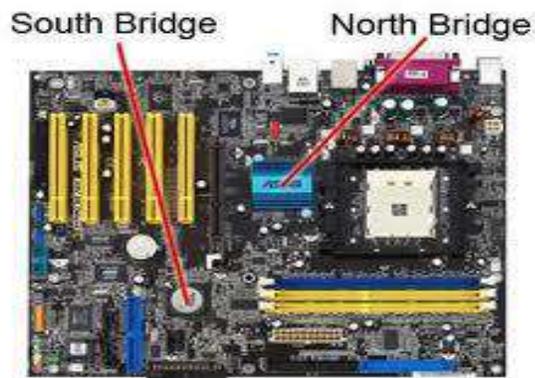
#### A. Perangkat Keras (*Hardware*)

Perangkat keras adalah komponen fisik komputer yang terdiri dari rangkaian elektronika dan peralatan mekanis lainnya, seperti disajikan pada Gambar 2.1 dan Gambar 2.2. Pada abstraksi tingkat atas terdiri dari empat komponen, yaitu :

- 1) Pemroses (*processor*)
- 2) Memori utama (main memori)
- 3) Perangkat masukan dan keluaran (device I/O)
- 4) Interkoneksi antar komponen (*user interface, device controller*)



Gambar 3.1 Skema dasar sistem komputer



Gambar 3.2 Letak north bridge dan south bridge

Kedua nama ini mengandung arti yang berbeda, yaitu

- *North Bridge* adalah chipset intel yang terhubung Langsung dengan processor computer dan memori controller yang memberikan akses cepat dari CPU ke memori, Peripheral Componen Interconect (PCI) bus, level 2 cache, dan semua kegiatan Accelerated Graphics Port (AGP).
- *Southbridge* mengandung setidaknya controller PCI, floppy / IDE / pengendali Hard disk, port serial dan parallel, dukungan USB dan fungsi manajemen daya.

**Northbridge pada CPU terkoneksi dengan:**

- RAM
- AGP bus
- PCI Express bus
- Built-in display Adapter

**Southbridge pada CPU terkoneksi dengan:**

- ATA (IDE) Drives
- USB bus
- FireWire bus
- Serial port
- Parallel port
- Built-in audio
- ISA bus (earlier PCs)

Sumber Daya Keras (Perangkat Keras) terdiri atas :

i. Pemrosesan (*Processor*)

Komponen computer yang bertugas untuk mengolah data dan melaksanakan berbagai perintah.

Pemroses terdiri dari:

- Bagian ALU (*Aritmatic Logic Unit*) untuk komputasi, berupa operasi-operasi aritmatika dan logika.
- Bagian CU (*control unit*) untuk pengendalian operasi yang dilaksanakan system computer

ii. Memori

- Memori berfungsi tempat penyimpanan data dan program.

Terdapat beberapa tipe memori:

- Register
- Memori case (*Chace Memory*)
- Memori kerja (*Main Memory*)
- Disk magnetic (*Magnetic Disc*)
- Disk optic (*Optical Disc*)

- Tape magnetic (*Magnetic Tape*)

Menurut urutan dari atas ke bawah dapat diukur hirarki dalam hal

- 1) Kecepatan Akses
- 2) Hubungan Kapasitas
- 3) Hubungan Frekwensi Pengaksesan
- 4) Hubungan Harga

Setiap kali pemroses melakukan eksekusi adanya lalu lintas data dengan memori utama, maka diimplementasikan adanya konsep *Chace Memori*, menanggulangi kelambatan proses. Juga pada memori utama dengan peralatan masukan/ keluaran saling berhubungan, maka diimplementasikan adanya konsep penampung sementara yang akan dikirim keperangkat masukan/keluaran berupa Buffering.

### iii. Perangkat Masukan/Keluaran

Perangkat masukan/keluaran terdiri dari dua bagian, yaitu:

- 1) Komponen mekanik adalah perangkat itu sendiri
- 2) Komponen elektronik, yaitu pengendali perangkat berupa *chip controller*

Pengendalian Perangkat (*Device Adapter*) terdapat dua macam pengendali alat:

- Penggerak alat (*Device Controller*)
- Pekerja alat (*Device Driver*)

Struktur I/O

- 1) I/O Interrupt => I/O Device kec rendah
  - 2) Struktur DMA => I/O Device kec tinggi
- DMA dibagi menjadi dua yaitu:
- Third Party DMA
  - First Party DMA

## B. Perangkat Lunak (Software)

- Merupakan komponen non fisik berupa kumpulan program beserta struktur datanya.
- Program adalah sekumpulan instruksi yang disusun sedemikian rupa untuk dapat menyelesaikan masalah-masalah tertentu sesuai dengan kebutuhan.

## SOAL LATIHAN :

1. Jelaskan tentang komponen-komponen perangkat keras dalam komputer !
2. Jelaskan tentang koneksi antara *northbridge* dan *southbridge* dalam CPU!
3. Jelaskan tentang cache memory!
4. Jelaskan tentang jenis-jenis prosesor yang anda ketahui !
5. Bagaimana perkembangan teknologi prosesor sampai dengan saat ini?

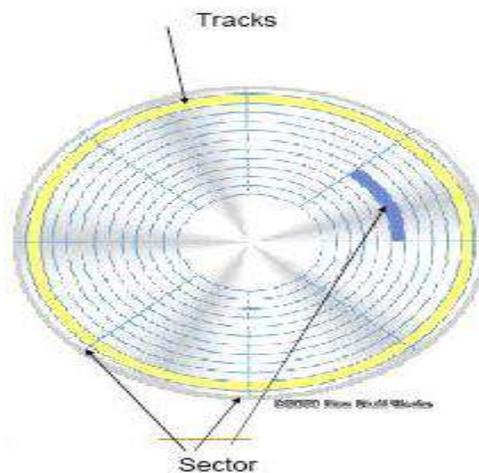
## Bab 4

### Struktur Penyimpanan Massal

#### A. Magnetic Disc

Magnetic disk merupakan penyimpan sekunder, berbentuk bundar permukaan magnetik. Penggerak disk berupa motor drive menggerakkan disk dengan kecepatan tinggi (kurang-lebih dari 60 putaran perdetik).

Kegiatan baca-tulis dilakukan *Read-write head*, yang diletakkan diatas piringan. Kepala baca-tulis sangat sensitif terhadap guncangan yang dapat menyebabkan disk rusak (*bad sector*). Ruang rekam terbagi atas beberapa track/lintasan dan tiap lintasan dibagi lagi dalam beberapa sector, seperti pada Gambar 4.1.



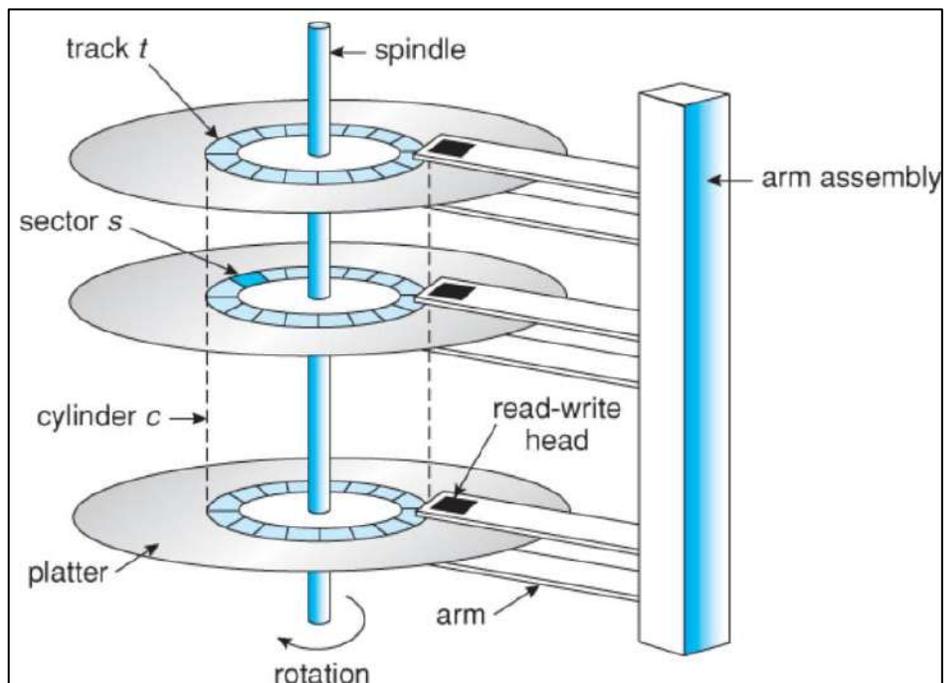
**Gambar 4.1 Penampang piringan (disc)**

Disk magnetik tradisional memiliki struktur dasar berikut:

- 1) Satu atau lebih piring-piring dalam bentuk disk yang ditutupi dengan media magnetik. Piring hard disk terbuat dari logam yang kaku, sedangkan disk "floppy" terbuat dari plastik yang lebih fleksibel.
- 2) Setiap piring memiliki dua permukaan yang berfungsi. Hard disk drive yang lebih tua kadang-kadang tidak akan menggunakan permukaan paling atas atau bawah dari tumpukan piring-piring, karena permukaan ini lebih rentan terhadap kerusakan potensial.
- 3) Setiap permukaan kerja dibagi menjadi sejumlah cincin konsentris yang disebut trek. Pengumpulan semua trek yang jaraknya sama dari tepi piring, (misalnya Semua trek tepat di atas satu sama lain dalam diagram berikut) disebut silinder.

- 4) Setiap lagu dibagi lagi menjadi beberapa sektor, secara tradisional masing-masing berisi 512 bytes data, meskipun beberapa disk modern terkadang menggunakan ukuran sektor yang lebih besar. (Sektor juga mencakup header dan trailer, termasuk informasi checksum antara lain.)
- 5) Ukuran sektor yang lebih besar mengurangi fraksi disk yang dikonsumsi oleh header dan trailer, tetapi meningkatkan fragmentasi internal dan jumlah disk yang harus ditandai buruk dalam kasus kesalahan.)
- 6) Data pada hard drive dibaca oleh kepala baca-tulis. Konfigurasi standar (diperlihatkan di bawah) menggunakan satu kepala per permukaan, masing-masing pada lengan yang terpisah, dan dikendalikan oleh unit lengan bersama yang menggerakkan semua kepala secara bersamaan dari satu silinder ke silinder lainnya. (Konfigurasi lain, termasuk kepala baca-tulis independen, dapat mempercepat akses disk, tetapi melibatkan kesulitan teknis yang serius.)
- 7) Kapasitas penyimpanan disk drive tradisional sama dengan jumlah kepala (yaitu jumlah permukaan yang bekerja), kali jumlah trek per permukaan, kali jumlah sektor per lagu, kali jumlah byte per sektor. Blok fisik data tertentu ditentukan dengan memberikan nomor kepala-sektor-silinder di mana ia berada.

Contoh pergerakan head disk disajikan pada Gambar 4.2 berikut.



Gambar 4.2 Mekanisme Moving Head Disk

Dalam operasi, disk berputar dengan kecepatan tinggi, seperti 7200 rpm (120 putaran per detik). Kecepatan transfer data dari disk ke komputer terdiri dari beberapa langkah:

- Waktu pencarian posisi, a.k.a., waktu mencari atau waktu akses acak adalah waktu yang diperlukan untuk memindahkan kepala dari satu silinder ke silinder lain, dan agar kepala diam setelah gerakan. Ini biasanya merupakan langkah paling lambat dalam proses dan hambatan utama terhadap tingkat transfer keseluruhan.
- Latency rotasi adalah jumlah waktu yang diperlukan untuk sektor yang diinginkan untuk berputar dan berada di bawah kepala baca-tulis. Ini dapat berkisar dari nol hingga satu revolusi penuh, dan rata-rata akan sama dengan setengah revolusi. Ini adalah langkah fisik lain dan biasanya langkah paling lambat kedua di belakang mencari waktu. (Untuk disk yang berputar pada 7200 rpm, latensi rotasi rata-rata adalah 1/2 revolusi / 120 putaran per detik, atau hanya lebih dari 4 milidetik, lama menurut standar komputer.
- Kecepatan transfer, yang merupakan waktu yang diperlukan untuk memindahkan data secara elektronik dari disk ke komputer. (Beberapa penulis juga dapat menggunakan istilah transfer rate untuk merujuk ke keseluruhan transfer rate, termasuk mencari waktu dan latensi rotasi serta laju transfer data elektronik.)

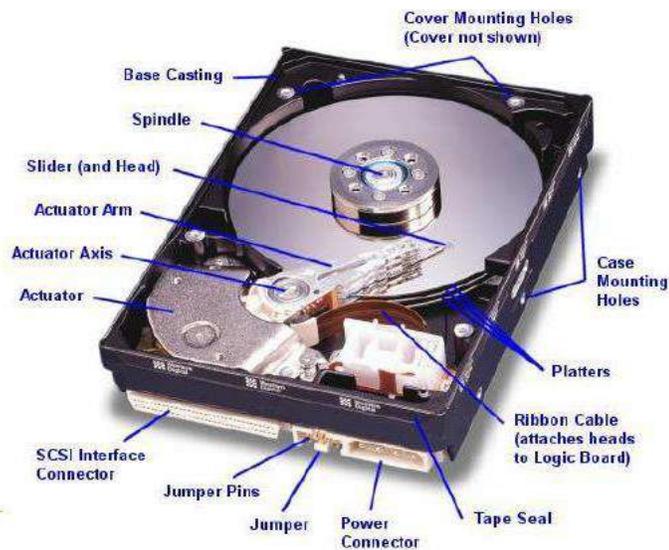
Kepala disk "terbang" di atas permukaan di atas bantalan udara yang sangat tipis. Jika mereka secara tidak sengaja menghubungi disk, maka terjadi crash kepala, yang mungkin atau mungkin tidak merusak disk secara permanen atau bahkan menghancurkannya sepenuhnya. Untuk alasan ini, biasanya memarkir kepala disk saat mematikan komputer, yang berarti memindahkan kepala dari disk atau ke area disk di mana tidak ada data yang disimpan.

Floppy disk biasanya dapat dilepas. Hard drive juga dapat dilepas, dan beberapa bahkan hot-swappable, yang berarti mereka dapat dihapus saat komputer sedang berjalan, dan hard drive baru dimasukkan di tempatnya.

Disk drive terhubung ke komputer melalui kabel yang dikenal sebagai I/O Bus. Beberapa format antarmuka umum termasuk Enhanced Integrated Drive Electronics, EIDE; Lampiran Teknologi Canggih, ATA; Serial ATA, SATA, Universal Serial Bus, USB; Fibre Channel, FC, dan Antarmuka Sistem Komputer Kecil, SCSI.

Pengontrol host berada di ujung komputer bus I/O, dan pengontrol disk dibuat ke dalam disk itu sendiri. CPU mengeluarkan perintah ke pengontrol host melalui port I/O. Data ditransfer antara permukaan magnetik dan cache onboard oleh pengontrol disk, dan kemudian data ditransfer dari cache itu ke pengontrol host dan memori motherboard pada kecepatan elektronik.

- Harddisk diukur dengan kecepatan RPM (Rotation Per Minute) ex. 5400, 7200, 10000 RPM
- Besarnya harddisk di ukur dengan ukuran 3GB, 6 GB bahkan ada yang mencapai 1 Tera (1000 GB)
- Harddisk Digunakan untuk menyimpan data berupa folder sampai sub folder terakhir
- Folder2 ini tersimpan pada sebuah track. Kumpulan track2 di sebut sector → masing2 sektor terdiri dari 512 bytes.
- Sector di kelompokkan lagi men jadi cluster



Gambar 4.3 Permukaan harddisk

Bagian-bagian dari hardisk adalah sebagai berikut:

- 1) Plater  
Adalah bagian dari harddisk untuk menyimpan data, berbentuk cakram, bundar
- 2) Board Logic  
Kontrol spindle drive, head actuator, dan semua perlintasan data
- 3) Head  
Piranti baca/tulis elekkromagnetis
- 4) Motor Spindle  
memutar plater
- 5) Head Actuator  
Mengontrol Pergerakan Head
- 6) Track dan Silinder  
Track = lingkaran konsentris pada platter

Silinder = multiple track, daerah sepusat dengan track

7) Sector dan Cluster

Sector= Bagian dari Track, 512 Bytes

Cluster= bagian dari Track

8) Boot Area

Menempati sector kedua pada hardisk, track pertama pada disk pertama (head 0)

## **B. Teknologi Hardisk**

Beberapa teknologi harddisk, antara lain:

1. Shock Protection System (SPS)
2. Self-Monitoring Analysis and Reporting (SMART)
3. Solid State Disk (SSD)
4. Magneto Resistive (MR)
5. Plug and Play ATA
6. Environment Protection Agency (EPA)

Adapun fungsinya adalah sebagai berikut:

1. Shock Protection System (SPS)

- Sebagian besar kerusakan yang timbul pada harddisk disebabkan karena guncangan. Guncangan pada harddisk dapat menyebabkan tergoncangnya head sehingga akan merusak piringan.
- Guncangan yang paling membahayakan adalah guncangan dengan intensitas tinggi dan dalam waktu yang singkat.
- Dengan menggunakan teknologi SPS, maka guncangan dapat diredam sehingga head tidak terangkat ketika terjadi guncangan.

2. Self-Monitoring Analysis and Reporting (SMART)

- Dengan menggunakan teknologi SMART, harddisk dapat berkomunikasi dengan komputer melalui software.
- Komunikasi yang dilakukan berisi tentang status keandalan harddisk, kemungkinan terjadinya kerusakan, dsb.
- Teknologi SMART sangat berguna bagi komputer-komputer yang memiliki data-data penting pada hardisk dan komputer yang dinyalakan terus menerus.

3. Solid State Disk (SSD)

- SSD yang dikembangkan saat ini tidak lagi menggunakan piringan magnetic, tetapi menggunakan DRAM (Dynamic RAM).

- SSD yang dikembangkan menggunakan antar muka SCSI memang dirancang untuk sistem komputer yang memerlukan akses data yang cepat, seperti server, server database.
4. Magneto Resistive (MR)
- Saat ini, head hardisk yang digunakan dikenal dengan nama induktif head.
  - Untuk teknologi MR, head ini digunakan untuk membaca saja, untuk menulis digunakan jenis head Thin Film.
  - Diimplementasikan pada harddisk dengan ukuran 1 GB sampai dengan 30 GB.
5. Plug and Play ATA (Advance Technology Attachment)
- Sistem PnP adalah melakukan konfigurasi secara otomatis dan akan memudahkan pengaturan, cukup melalui software saja tidak perlu mengubah jumper, dsb.
  - Harddisk dengan Serial Advanced Technology Attachment (SATA) telah banyak dipakai di seluruh dunia. Para pengembang motherboard juga lebih cenderung mengembangkan dukungan harddisk SATA daripada IDE (Integrated Drive Electronic) yang sudah kuno.
6. Environment Protection Agency (EPA)
- Harddisk termasuk komponen yang menghabiskan tenaga listrik cukup banyak pada PC, apalagi pada notebook/laptop.
  - Untuk itu, harddisk yang mendukung program EPA mempunyai kemampuan untuk menghemat listrik, misalnya untuk fungsi sleep, standby, dsb.

### **C. Disk Attachment**

Disk drive dapat dipasang langsung ke host tertentu (disk lokal) atau ke jaringan.

#### **1. Host-Attached Storage**

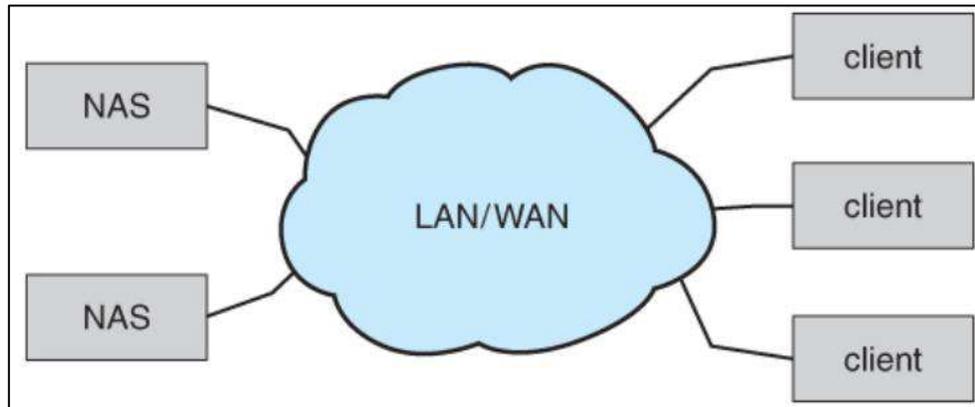
- Disk lokal diakses melalui I / O Ports seperti dijelaskan sebelumnya.
- Antarmuka yang paling umum adalah IDE atau ATA, yang masing-masing memungkinkan hingga dua drive per pengontrol host.
- SATA mirip dengan pemasangan kabel yang lebih sederhana.
- Stasiun kerja kelas atas atau sistem lain yang membutuhkan lebih banyak disk biasanya menggunakan disk SCSI:
  - Standar SCSI mendukung hingga 16 target pada setiap bus SCSI, yang salah satunya umumnya berupa adaptor host dan 15 lainnya dapat berupa disk atau drive tape.

- Sasaran SCSI biasanya satu drive, tetapi standar juga mendukung hingga 8 unit dalam setiap target. Ini umumnya akan digunakan untuk mengakses disk individu dalam array RAID.
- Standar SCSI juga mendukung beberapa adaptor host dalam satu komputer, yaitu beberapa bus SCSI.
- Kemajuan modern dalam SCSI termasuk versi "cepat" dan "lebar", serta SCSI-2.
- Kabel SCSI dapat berupa 50 atau 68 konduktor. Perangkat SCSI dapat berupa eksternal maupun internal.
- FC adalah arsitektur serial berkecepatan tinggi yang dapat beroperasi melalui serat optik atau kabel tembaga empat konduktor, dan memiliki dua varian.
- Kain sakelar besar yang memiliki ruang alamat 24-bit. Varian ini memungkinkan beberapa perangkat dan beberapa host saling terhubung, membentuk dasar untuk jaringan area penyimpanan, SAN, untuk didiskusikan di bagian selanjutnya.
- Loop arbitrase, FC-AL, yang dapat menangani hingga 126 perangkat (drive dan pengontrol).

## **2. Network-Attached Storage (NAS)**

Penyimpanan yang terhubung jaringan menghubungkan perangkat penyimpanan ke komputer menggunakan panggilan prosedur jarak jauh, RPC, antarmuka, biasanya dengan sesuatu seperti mount sistem file NFS. Ini nyaman untuk memungkinkan beberapa komputer dalam kelompok mengakses umum dan konvensi penamaan untuk penyimpanan bersama.

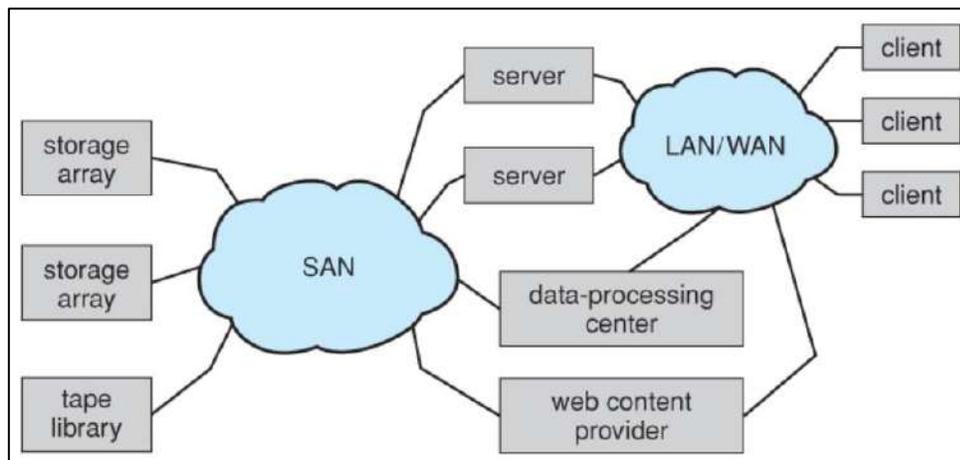
NAS dapat diimplementasikan menggunakan kabel SCSI, atau iSCSI menggunakan protokol Internet dan koneksi jaringan standar, yang memungkinkan akses jarak jauh ke file bersama. NAS memungkinkan komputer untuk dengan mudah berbagi penyimpanan data, tetapi cenderung kurang efisien daripada penyimpanan yang dipasang oleh host standar, arsitektur NAS dapat dilihat pada Gambar 4.3 berikut.



Gambar 4.3 *Network-attached storage*

### 3. Storage Area Network (SAN)

*Storage Area Network* (SAN) menghubungkan komputer dan perangkat penyimpanan di jaringan, menggunakan protokol penyimpanan selain menggunakan protokol jaringan. Salah satu keuntungannya adalah akses penyimpanan tidak mengikat bandwidth jaringan biasa. SAN sangat fleksibel dan dinamis, memungkinkan host dan perangkat untuk memasang dan melepaskan dengan cepat. SAN juga dapat dikontrol, memungkinkan akses terbatas ke host dan perangkat tertentu, seperti ditunjukkan pada Gambar 4.4 berikut.



Gambar 4.4. *Storage area network*

**SOAL LATIHAN:**

1. Jelaskan tentang teknologi harddisk yang anda ketahui!
2. Jelaskan keunggulan dari teknologi harddisk SPS !
3. Jelaskan tentang cara kerja dari teknologi harddisk EPA !
4. Jelaskan tentang perbedaan teknologi harddisk ATA dan SATA !
5. Sebutkan bagian-bagian dari magnetic disc !

## **Bab 5**

### **Pengaksesan Lintas Disk**

#### **A. Waktu Akses**

Waktu akses adalah yang diperlukan oleh kepala baca untuk menulis atau membaca isi sektor

**Terdiri dari 4 komponen waktu :**

- Waktu cari, waktu untuk mencapai lintas atau silinder yang dikehendaki.
- Waktu mantap, waktu untuk hulu tulis baca menjadi mantap di lintas atau silinder.
- Waktu latensi, waktu untuk mencapai hulu tulis baca.
- Waktu salur, waktu untuk menulis atau membaca isi sector
- Serta gabungan waktu yaitu waktu inkuiri dan waktu pemutakhiran atau pergantian.

#### **B. Algoritma Pengaksesan Lintas Disk**

Terdapat 7 algoritma pengaksesan disk:

1. Algoritma *First Come First Serve* (FCFS)/Pertama Tiba Pertama Dilayani (PTPD)
2. Algoritma Pick Up
3. Algoritma *Shortest Seek Time First* (SSTF)/Waktu Cari Terpendek Dipertamakan (WCTD)
4. Algoritma Look
5. Algoritma Circular Look (C-Look)
6. Algoritma Scan
7. Algoritma Circular Scan (C-Scan)

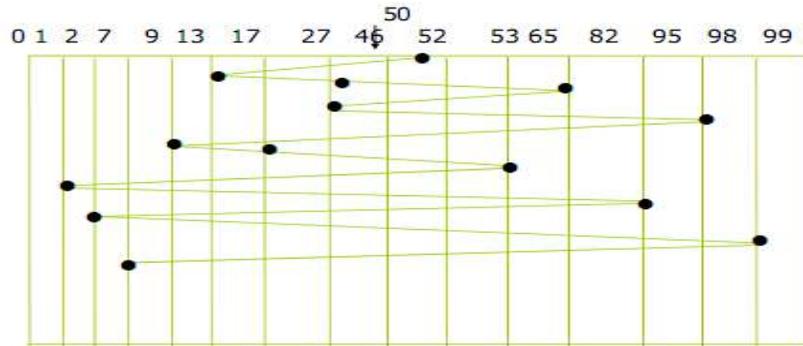
Penjelasannya adalah sebagai berikut :

##### **1. Algoritma FCFS/PTPD**

Proses pengaksesan akan dimulai secara berurutan sesuai dengan urutan tiba atau kedudukan antrian.

Contoh : diketahui antrian akses track dengan saat awal 50

13,46,65,27,95,82,9,17,1,82,2,17,98,7

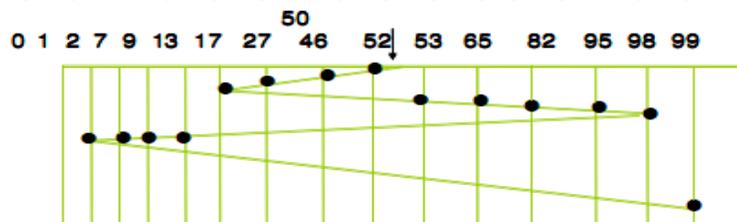


Langkah proses :  
 Dari 50 menuju ke lintasan 13, kemudian ke 46, ke 65, dan seterusnya. Setiap lintasan yang dilalui dihitung.

## 2. Algoritma PICK UP

Pada algoritma ini hulu tulis baca akan membaca atau menuju ke track yang terdapat pada urutan awal antrian, sambil mengakses track yang dilalui. Mirip seperti metode PTPD, tetapi lintasan yang dilewati dipungut/diambil, sehingga tidak perlu diakses lagi.

Contoh : diketahui antrian akses track dengan saat awal 50  
 13, 46, 65, 27, 95, 82, 9, 17, 52, 53, 17, 1, 82, 2, 17, 98, 7



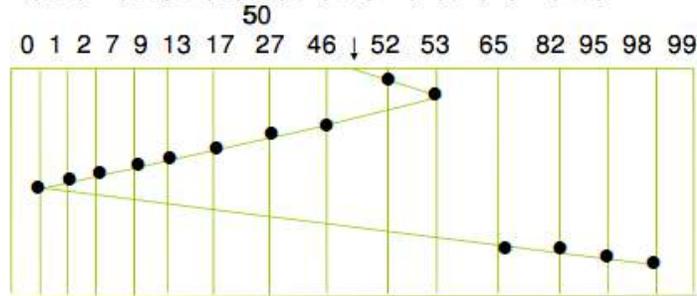
Total lintasan atau track yang di lewati adalah 310 track

Langkah proses :  
 Dari 50 menuju ke lintasan 13, lintasan yang dilewati 46, 27, dan 17 sekaligus dipungut/diakses. Sehingga selanjutnya tidak ke 46, tetapi ke 65, sekaligus memungut 52 dan 53. Karena 27 sudah diambil maka selanjutnya menuju 95, sekaligus memungut 82. Karena 82 sudah dipungut maka langsung menuju 1, dan seterusnya. Perhitungan 50-13, 13-65, 65-95, 95-1, dan seterusnya

## 3. Algoritma SSTF / WCTD

Proses dilaksanakan terhadap track yang terdekat dengan hulu baca/tulis di atas/bawah. Kemudian mencari letak track yang terdekat di atas/bawah dan seterusnya.

Contoh : diketahui antrian akses track dengan saat awal 50  
 13, 46, 65, 27, 95, 82, 9, 17, 52, 53, 17, 1, 82, 2, 17, 98, 7



Total lintas atau track yang di lewati adalah 152 track

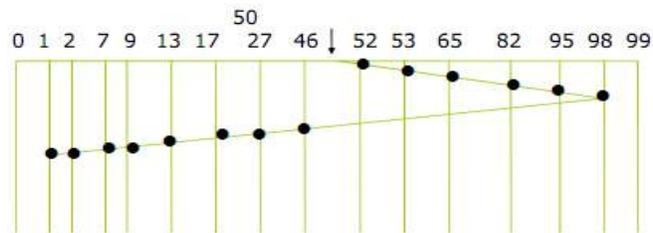
Langkah proses :

Hulu baca tulis mulai dari 50, antara 46 dan 52 yang terdekat 52, sehingga menuju ke 52. Selanjutnya dari 52, antara 46 dan 53 yang terdekat 53, dan seterusnya. perhitungan 50-52, 52-53, dan seterusnya.

#### 4. Algoritma Look

Pada algoritma ini, hulu baca/tulis akan bergerak nail seperti lift menuju antrian track terbesar pada disk sambil mengakses antrian track yang dilalui, kemudian turun menuju antrian track yang terkecil sambil mengakses track yang dilalui dan track yang telah diakses tidak diakses lagi.

Contoh : diketahui antrian akses track dengan saat awal 50  
 13, 46, 65, 27, 95, 82, 9, 17, 52, 53, 17, 1, 82, 2, 17, 98, 7



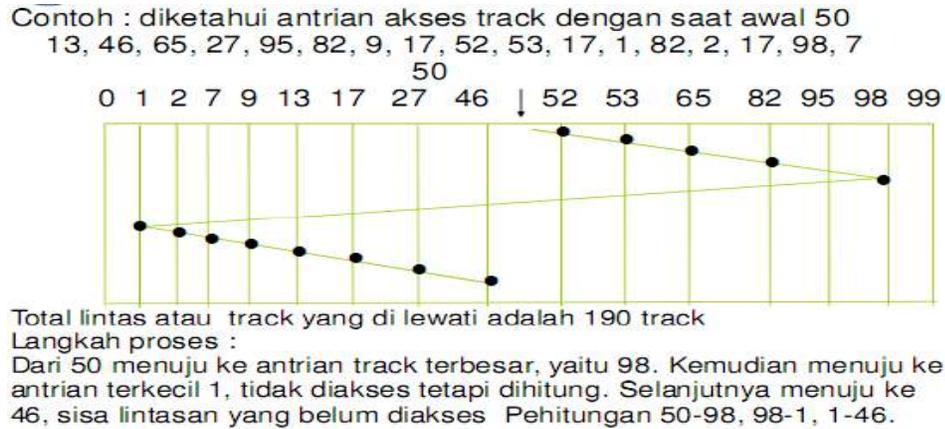
Total lintas atau track yang di lewati adalah 145 track

Langkah proses :

Dari 50 menuju ke antrian track terbesar, yaitu 98. Selanjutnya menuju ke antrian terkecil 1. Perhitungan 50-98, 98-1.

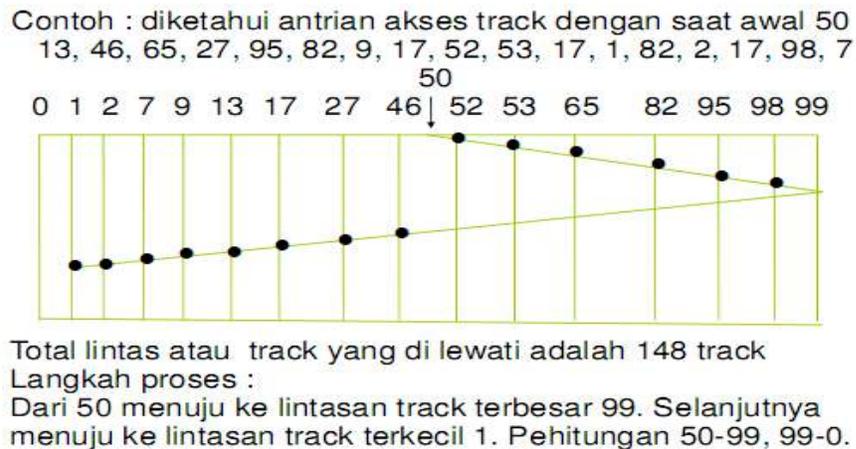
#### 5. Algoritma C-Look

Pada algoritma ini, hulu baca/tulis akan bergerak nail seperti lift menuju antrian track terbesar pada disk sambil mengakses antrian track yang dilalui, kemudian turun menuju antrian track yang terkecil tetapi tidak mengakses track yang dilalui, baru pada saat naik akan mengakses track yang belum diakses.



## 6. Algoritma Scan

Pada algoritma ini, hulu baca/tulis akan bergerak naik seperti lift menuju antrian track terbesar pada disk sambil mengakses antrian track yang dilalui, kemudian turun menuju antrian track yang terkecil sambil mengakses track yang dilalui, dan track yang telah diakses tidak boleh diakses lagi.



## 7. Algoritma C-Scan

Pada algoritma ini, hulu baca/tulis akan bergerak naik seperti lift menuju antrian track terbesar pada disk sambil mengakses antrian track yang dilalui, kemudian turun menuju antrian track yang terkecil tetapi tidak mengakses track yang dilalui, baru pada saat naik akan mengakses track yang belum diakses.

Contoh : diketahui antrian akses track dengan saat awal 50  
 13, 46, 65, 27, 95, 82, 9, 17, 52, 53, 17, 1, 82, 2, 17, 98, 7  
 50



Total lintas atau track yang di lewati adalah 194 track

Langkah proses :

Dari 50 menuju ke lintasan track terbesar 99. Selanjutnya menuju ke lintasan track terkecil 1, tidak diakses tetapi dihitung. Selanjutnya menuju ke 46, sisa lintasan yang belum diakses. Perhitungan 50-99, 99-0, 0-46.

### SOAL LATIHAN :

1. Jelaskan perbedaan algoritma pengaksesan disk menggunakan Circular Scan dan Scan!
2. Jelaskan perbedaan algoritma pengaksesan disk menggunakan Circular Look dan Look!
3. Diketahui antrian akses track dengan saat awal 45. Barisannya adalah 23, 46, 74, 1, 23, 65, 33, 22, 12, 87, 55, 90. Buatlah skema barisan proses menggunakan algoritma PTPD!
4. Dengan antrian yang sama dengan soal no.3, buatlah skema barisan prosesnya menggunakan algoritma SSTF!
5. Dengan antrian yang sama dengan soal no.3, buatlah skema barisan prosesnya menggunakan algoritma Scan!

## Bab 6

### Proses

#### A. Konsep Proses

Memori proses dibagi menjadi empat bagian seperti yang ditunjukkan pada Gambar 6.1 di bawah ini:

- 1) Bagian teks terdiri dari kode program yang dikompilasi, dibaca dari penyimpanan non-volatil ketika program diluncurkan.
- 2) Bagian data menyimpan variabel global dan statis, dialokasikan dan diinisialisasi sebelum mengeksekusi utama.
- 3) Heap digunakan untuk alokasi memori dinamis, dan dikelola melalui panggilan ke yang baru, hapus, malloc, gratis, dll.
- 4) Batch (Tumpukan) digunakan untuk variabel lokal. Ruang pada stack dicadangkan untuk variabel lokal ketika mereka dideklarasikan (di pintu masuk fungsi atau di tempat lain, tergantung pada bahasa), dan ruang dibebaskan ketika variabel keluar dari ruang lingkup. Perhatikan bahwa tumpukan juga digunakan untuk nilai pengembalian fungsi, dan mekanisme pasti pengelolaan tumpukan mungkin spesifik bahasa.

Perhatikan bahwa tumpukan dan tumpukan mulai di ujung yang berlawanan dari ruang bebas proses dan tumbuh menuju satu sama lain. Jika mereka pernah bertemu, maka kesalahan stack overflow akan terjadi, atau panggilan ke malloc baru akan gagal karena memori yang tersedia tidak mencukupi.

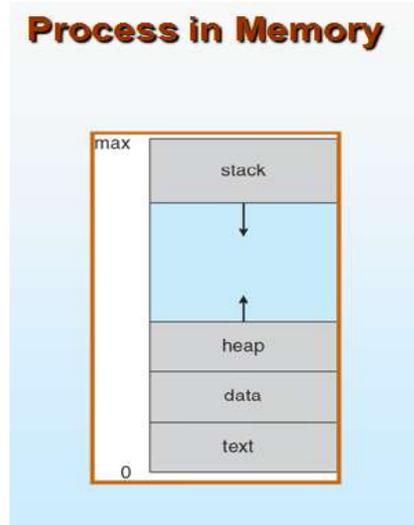
Ketika proses ditukar keluar dari memori dan kemudian dipulihkan, informasi tambahan juga harus disimpan dan dipulihkan. Kunci di antara mereka adalah penghitung program dan nilai semua register program.

Sebuah sistem operasi mengeksekusi berbagai macam program

- *Batch system – jobs*
- *Time shared systems – user program / tasks*

Sebuah proses mencakup :

- program counter dan proses register
- *stack and heap*
- *text section*
- *data section*

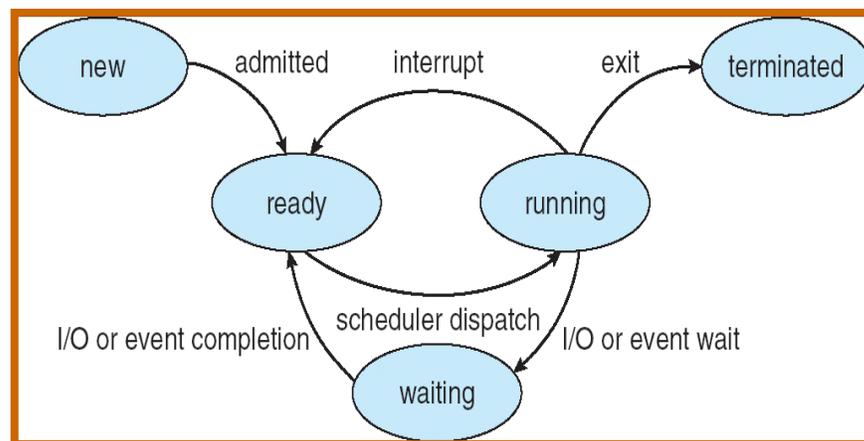


Gambar 6.1 Proses pada Memori

## B. Tingkatan Proses

Perubahan tingkatan akan terjadi saat proses dieksekusi (Gambar 6.2):

1. **New** : proses sedang dibuat
2. **Running** : instruksi sedang dieksekusi
3. **Waiting** : proses sedang menunggu untuk suatu kondisi (misalnya untuk penerimaan sinyal, I/O completion)
4. **Ready** : proses sedang menunggu untuk diterima prosesor
5. **Terminated** : proses telah selesai dieksekusi

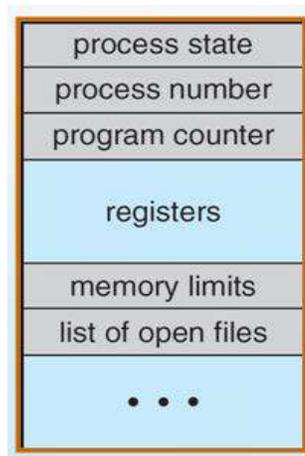


Gambar 6.2 Tingkatan Proses

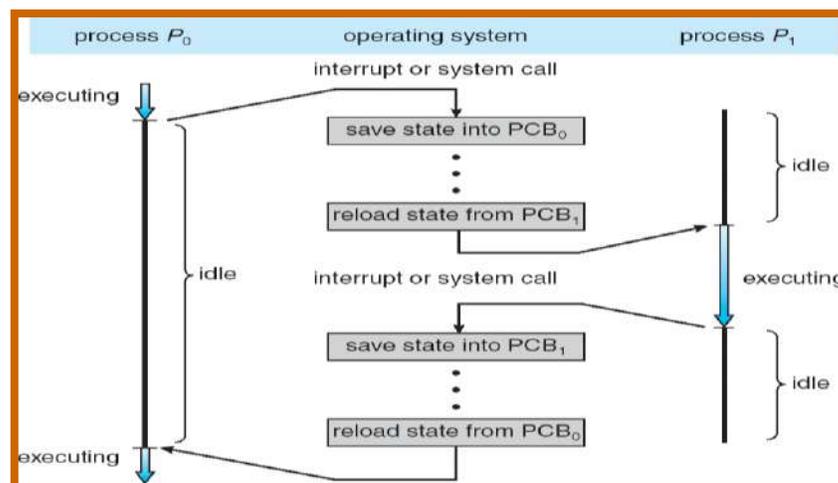
### *Proses Control Block (PBC)*

Informasi yang menyertai setiap proses adalah sebagai berikut (Gambar 6.3 dan Gambar 6.4) :

- 1) Process state
- 2) Program counter
- 3) CPU registers
- 4) CPU scheduling information
- 5) Memory – management information
- 6) Accounting information  
I/O status information



Gambar 6.3 Informasi setiap proses



Gambar 6.2 CPU mengubah dari Proses ke Proses

**SOAL LATIHAN :**

1. Jelaskan tentang konsep proses !
2. Jelaskan tentang tingkatan proses !
3. Bagaimana CPU mengubah dari proses ke proses ?

## **Bab 7**

### **Penjadwalan Proses**

#### **A. Konsep Penjadwalan Proses**

Penjadwalan merupakan kumpulan kebijaksanaan dan mekanisme di system operasi yang berkaitan dengan urutan kerja yang dilakukan sistem komputer.

Penjadwalan bertugas memutuskan: Proses harus berjalan, kapan dan berapa lama proses itu berjalan.

#### **B. Tujuan Penjadwalan Proses**

- Supaya semua pekerjaan memperoleh pelayanan yang adil (fairness)
- Supaya pemakaian prosesor dapat dimaksimumkan
- Supaya waktu tanggap dapat diminimumkan, berupa waktu tanggap nyata dan waktu tanggap maya
- Supaya pemakaian sumber daya seimbang
- Turn around time, waktu sejak program masuk ke system sampai proses selesai.
- Efisien, proses tetap dalam keadaan sibuk tidak mengganggu
- Supaya terobosan (*throughput*) dapat dimaksimumkan

#### **C. Tipe Penjadwalan**

1. Penjadwalan jangka pendek → Bertugas menjadwalkan alokasi pemroses di antara proses-proses ready di memori utama. Penjadwalan dijalankan setiap terjadi pengalihan proses untuk memilih proses berikutnya yang harus dijalankan.
2. Penjadwalan jangka menengah → Proses status Ready – running – Waiting – ready
3. Penjadwalan jangka panjang → Penjadwal ini bekerja terhadap antrian batch dan memilih batch berikutnya yang harus di eksekusi. Batch biasanya adalah proses-proses dengan penggunaan sumber daya yang intensif (yaitu waktu pemroses, memori, perangkat masukan/keluaran), program-program ini berprioritas rendah, digunakan sebagai pengisi (agar pemroses sibuk) selama periode aktivitas job-job interaktif rendah.

3 istilah yang digunakan pada penjadwalan proses

1. Antrian  
Sejumlah proses yang menunggu menggunakan prosesor dan akan diproses sesuai dengan urutan antrian proses.

Proses berada dalam memori utama

2. Prioritas

Mendahului pada antrian proses, kalau proses itu berada di bagian belakang antrian, maka dengan pemberian prioritas, proses itu langsung berada di bagian paling depan pada antrian itu sambil menunggu sampai kerja prosesor selesai.

3. Prempsi

Mendahului pada antrian proses, kalau proses itu berada di bagian belakang antrian, maka dengan pemberian prempsi, proses itu langsung berada di bagian paling depan pada antrian itu bahkan akan memberhentikan kerja prosesor untuk mengerjakan proses yang prempsi tersebut.

**1. Algoritma Penjadwalan Pertama Tiba Pertama Dilayani (PTPD/FIFO/FCFS)**

Penjadwalan tanpa prioritas tanpa prempsi, proses yang tiba lebih dahulu akan dilayani lebih dahulu, jika tiba pada waktu yang bersamaan akan dilayani sesuai dengan urutan pada antrian.

Contoh :

Nama Proses	Saat Tiba	Lama Proses
A	0	4
B	1	7
C	3	3
D	7	8

**2. Algoritma Penjadwalan Proses Terpendek Dipertamakan – Non Preemptive (PTD/SJF/SJN)**

Penjadwalan dengan prioritas tanpa prempsi, terdapat 2 langkah :

Penentuan berdasarkan pendeknyaproses yang dilayani jika proses yang terpendek tersebut belum tiba maka prosesor akan melayani proses yang telah tiba sampai proses tersebut selesai.

Contoh :

Tabel proses PTD

Nama Proses	Saat Tiba	Lama Proses
A	0	5
B	3	7
C	5	2
D	6	4

### 3. Algoritma Penjadwalan Proses Terpendek Dipertamakan – Preemptive (PTDP)

Penjadwalan dengan prioritas dengan prempsi beberapa ketentuan :

- ⊙ Prioritas berdasarkan pendeknya sisa proses
- ⊙ Diperhatikan saat proses tiba atau saat proses selesai
- ⊙ Menghitung lama sisa proses dari semua proses yang ada
- ⊙ Jika ada proses dengan sisa proses yang lebih pendek dari proses yang sedang di kerjakan, maka atas dasar prempsi proses yang sedang dikerjakan akan dikeluarkan dari prosesor.

#### SOAL LATIHAN :

1. Jelaskan tentang algoritma penjadwalan proses FCFS !
2. Jelaskan tentang algoritma penjadwalan proses SJF-Non Preemptive !
3. Jelaskan tentang istilah preemptive dan non preemptive !
4. Mengapa perlu dilakukan penjadwalan proses ?

## **Bab 8**

### **Pengenalan Distro Linux**

#### **A. Sejarah Perkembangan Linux**

Sejarah sistem operasi Linux berkaitan erat dengan proyek GNU, proyek program bebas *freeware* terkenal diketuai oleh Richard Stallman. Proyek GNU diawali pada tahun 1983 untuk membuat sistem operasi seperti Unix lengkap kompilasi, utiliti aplikasi, utiliti pembuatan seterusnya diciptakan sepenuhnya dengan perangkat lunak bebas. Pada tahun 1991, pada saat versi pertama kerangka Linux ditulis, proyek GNU telah menghasilkan hampir semua komponen sistem ini kecuali kernel. Torvalds dan pembuat kernel seperti Linux menyesuaikan kernel mereka supaya dapat berfungsi dengan komponen GNU, dan seterusnya mengeluarkan Sistem operasi yang cukup berfungsi. Oleh karena itu, Linux melengkapi ruang terakhir dalam rancangan GNU/Linux (atau GNU/Linux) adalah operating system open source yang mulai dikembangkan oleh Linus Torvalds dan masih dikembangkan terus hingga sekarang oleh banyak sukarelawan dan developer di seluruh dunia. Istilah "Linux Indonesia" adalah istilah generik yang dipakai untuk mengacu kepada keseluruhan komunitas Linux Indonesia. Linux adalah sebuah sistem yang multiuser, multitasking dengan tools kompatibel UNIX yang lengkap. Sistem file-nya mengikuti semantic UNIX tradisional dan mengimplementasikan secara penuh standar model network.

#### **B. Distro Linux**

Distro LINUX adalah LINUX yang dibuat dengan memaketkan program tertentu berdasarkan source yang ada dan dikemas sedemikian rupa sehingga mempunyai tampilan atau fitur yang berbeda-beda sesuai dengan keinginan si pembuat distro. Ada distro yang berfokus pada server artinya distro tersebut lebih dioptimasi ke sistem server sehingga software atau program yang di paketkan adalah khusus untuk server saja contoh untuk distro ini antara lain : Trustix Secure Linux, Turbo Linux Server, Red Hat Enterprise Server, Server Optimize Linux (SOL), Mandrake Security dan masih banyak lagi contoh lainnya.

Ketika pengguna Linux baru berkenalan dengan Linux, biasanya mereka terbentur pada masalah distro yang akan dipakai/dipilih. Secara garis besar distro Linux sama yaitu menggunakan 'kernel' Linux, perbedaannya hanya pada paket program, program instalasi, organisasi direktori dan berkas, program aplikasi dan utilitas tambahan. Distro Linux berbeda untuk kebutuhan yang berbeda. Dibawah ini merupakan beberapa distro Linux, diantaranya :

## 1. Lycoris

Lycoris adalah distro linux yang telah membuat versi linux yang cantik dan menyerupai WinXP, dari segi warna, icon, maupun kemudahan-kemudahan yang ada. Dari tampilan-tampilan screenshot pada situsnya Anda dapat mengamati bahwa memang Lycoris merupakan distro Linux yang pintar mempercantik diri sehingga dengan menggunakan Lycoris Anda akan merasa bekerja dengan Windows XP. Bahkan OpenOffice yang dipaketkan dengan Lycoris memiliki icon-icon yang cantik dan khas Lycoris. Lycoris memiliki paket-paket yang berbeda, dan paket distro Linuxnya dinamakan Desktop /LX. Lycoris memiliki fasilitas IRIS yaitu Internet Rapid Installer for Software untuk mengupdate Linux Anda dengan software-software baru. Hanya saja distro ini tidaklah gratis dan Anda perlu membayar dengan sejumlah "dollar US".

## 2. Xandros

Xandros memiliki integrasi lebih baik dengan jaringan Windows, mampu menjalankan aplikasi OfficeXP, mampu meresize partisi NTFS saat instalasi, dll. Xandros juga bukan merupakan produk gratis tetapi komersial. Secara singkat Xandros menawarkan kelebihan berikut:

- Lima langkah instalasi grafis yang mudah
- Manajemen partisi terintegrasi
- Tampilan yang familiar
- Aplikasi berkualitas dari para engineer Xandros
- Kompatibilitas dengan file format Microsoft

## 3. Lindows

Lindows masih kalah dibanding Lycoris dalam segi kecantikan tampilan tetapi Lindows telah lebih matang dan telah mencapai versi 4.0. Fasilitas yang menarik dari Lindows adalah Click-n-Run yaitu fasilitas mendapatkan ribuan software Linux yang sudah dikonfigurasi untuk Lindows dengan sekali klik dan proses instalasi ini juga dapat menambahkan shortcut di desktop dan daftar program secara otomatis. Lindows juga merupakan produk komersial.

## 4. Linare

Linare memiliki distro Linux khusus desktop dan paket CPU yang sudah terinstall Linare. Linare masih jarang terdengar dan merupakan pendatang baru. Linare juga merupakan produk komersial.

## **5. Debian/GNU Linux**

Debian GNU/Linux adalah distro non komersial yang dihasilkan oleh para sukarelawan dari seluruh dunia yang saling bekerjasama melalui Internet. Distro ini menginginkan adanya semangat open-source yang harus tetap ada pada Debian. Kedinamisan distro ini membuat setiap rilis paket-paketnya di-update setiap waktu dan dapat diakses melalui utilitas apt-get. Apt-get adalah sebuah utilitas baris-perintah yang dapat digunakan secara dinamis untuk meng-upgrade sistem Debian GNU/Linux melalui apt-repository jaringan archive Debian yang luas. Milis dan forum debian selalu penuh dengan pesan-pesan baik mengenai bug, masalah, sharing, dll. Dengan adanya sistem komunikasi ini bug dan masalah keamanan pada tiap paket dapat dilaporkan oleh para pengguna dan pengembang Debian dengan cepat. Keuntungan dari Debian adalah upgradability, ketergantungan antar paket didefinisikan dengan baik, dan pengembangannya secara terbuka.

## **6. Linux-Mandrake**

Linux-Mandrake adalah salah satu distro turunan dari Red Hat Linux yang menyediakan banyak pengembangan dan aplikasi 'pre-configured' dan didukung banyak bahasa di seluruh dunia. Distro ini dikenal mudah untuk pemula dan cocok untuk kelas desktop tapi tidak menutup kemungkinan untuk dijadikan platform pada server. Optimasi untuk prosesor kelas Pentium ke atas membuat Linux-Mandrake dapat berjalan dengan baik pada platform tersebut.

## **7. Red Hat Linux**

Red Hat adalah distro yang cukup populer di kalangan pengembang dan perusahaan Linux. Dukungan-dukungan secara teknis, pelatihan, sertifikasi, aplikasi pengembangan, bergabungnya para hacker dankernel dan free-software seperti Alan Cox, Michael Johnson, Stephen Tweedie menjadikan Red Hat berkembang cepat dan digunakan pada perusahaan. Poin terbesar dari distro ini adalah Red Hat Package Manager (RPM). RPM adalah sebuah perangkat lunak untuk manajemen paket-paket pada sistem Linux kita dan dianggap sebagai standar de-facto dalam pemaketan pada distro-distro turunannya dan yang mendukung distro ini secara luas.

## **8. Slackware**

Distronya Patrick Volkerding yang terkenal pertama kali setelah SLS. Slackware dikenal lebih dekat dengan gaya UNIX, sederhana, stabil, mudah dikustom, dan didesain untuk komputer 386/486 atau lebih tinggi. Distro ini termasuk distro yang cryptic dan manual sekali bagi pemula Linux, tapi dengan menggunakan distro ini

beberapa penggunanya dapat mengetahui banyak cara kerja sistem dan distro tersebut. Slackware adalah tertua yang masih tegar bertahan di segala jaman. Hari ini dengan bangga merilis generasi barunya Slackware versi 11 diantara gelombang peluncuran distro-distro besar lainnya seperti Mandriva 2007 dan tidak lama lagi Fedora 6.

## **9. Turbo Linux**

Turbo Linux merupakan salah satu distro Linux yang diminati oleh perusahaan dan perorangan di Jepang dan Asia. Produk berbasis Linux dengan kinerja tinggi ini dimanfaatkan untuk pasar workstation dan server terutama untuk penggunaan clustering dan orientasinya ke perusahaan. Beberapa produk-produknya: TurboLinux Workstation untuk dekstopnya, TurboLinux Server untuk backend server dengan kinerja tinggi terutama untuk penggunaan bisnis di perusahaan, e-commerce dan transaksi B2B (*Business-to-Business*).

## **10. Knoppix**

Knoppix merupakan distro Linux live-cd yang dapat dijalankan melalui CD-ROM tanpa menginstalnya di hard-disk. Aplikasinya sangat lengkap dan cocok untuk demo atau belajar Linux bagi yang belum mempunyai ruang pada hard-disknya, dapat juga untuk CD rescue. Kelemahan dari knoppix adalah diperlukannya memori yang besar untuk menggunakan modus grafisnya yaitu 96 MB walaupun bisa juga dijalankan pada memori 64 MB dengan swap pada hard-disk seperti pengalaman penulis tapi Anda perlu sedikit bersabar. Distro ini berbasis Debian GNU/Linux.

### **SOAL LATIHAN:**

1. Jelaskan tentang istilah open source !
2. Jelaskan tentang distro Linux yang paling awal muncul !
3. Jelaskan tentang kelebihan dan kekurangan sistem open source !

## **Bab 9**

### **Pembuatan Distro Linux Slax**

#### **A. Pembuatan Slax**

Saat ini komputer sudah merupakan kebutuhan yang sangat penting untuk menunjang pekerjaan. dalam sebuah komputer harus memiliki operating sistem untuk mengoperasikan software lainnya. banyak software yang berbayar yang bisa digunakan, namun bila belum cukup mampu untuk membelinya bisa menggunakan operating sistem linux. pada saat ini bagi para pemula bisa membuat sendiri sistem operasi linux dengan menggunakan Myslax creator.

#### **B. Tools untuk Pembuatan Slax**

Myslax creator merupakan tool software yang digunakan untuk membuat sistem operasi berbasis linux sendiri, dengan Myslax creator sistem operasi yang dibuat dapat dimodifikasi sesuai keinginan dan kebutuhan. untuk membuat distro linux dengan Myslax creator ini diperlukan tool software tambahan yaitu microsoft visual studio versi 2008 keatas dan file linux dengan format ISO yang digunakan sebagai installer, serta *tool software* VMware atau sejenisnya untuk menjalankan atau mencoba distro linux yang telah dibuat. dengan Myslax creator ini dapat pula ditambahkan beberapa software standart seperti mozilla untuk browsing internet, game dan beberapa software lain yang dibutuhkan. mungkin untuk pemula dibidang komputer atau bukan seorang pekerja programmer, menggunakan sistem operasi sendiri yang telah dibuat pada komputer untuk bekerja dapat menimbulkan kesenangan tersendiri dan dapat meningkatkan semangat bekerja.

Untuk membuat distro linux dengan myslax creator, download Myslax creator versi 1.6 kemudian download juga sistem boot file slaxnya yang format .ISO untuk dijalankan melalui Hard drive atau CD atau dengan format .tar agar aplikasi bisa jalan pada USB atau flas disk. untuk merubah script sistem bootnya digunakan SDK (*software development kit*) yang ada pada microsoft visual studio ver 2008 dan setelahnya. setelah itu download module software yang akan dimasukkan ke sistem operasi yang akan dibuat yang bisa didownload pada [www.slax.org](http://www.slax.org). setelah itu jalankan myslax creator dan masukkan modul software yang telah didownload. setelah itu proses pembuatan sistem operasi selesai dan siap digunakan. MYslax creator sangat mudah digunakan oleh para pemula dibidang komputer yang ingin membuat sistem operasi sendiri. sehingga para pengguna komputer tidak harus dirisaukan dengan membeli lisensi yang mahal bila digunakan untuk individu.

#### **SOAL :**

Apa yang dimaksud dengan Myslax Creator? Jelaskan tentang proses remastering!

## **Bab 10**

### **Modifikasi Distro Linux**

Saat ini komputer sudah merupakan kebutuhan yang sangat penting untuk menunjang pekerjaan. dalam sebuah komputer harus memiliki operating sistem untuk mengoperasikan software lainnya. banyak software yang berbayar yang bisa digunakan, namun bila belum cukup mampu untuk membelinya bisa menggunakan operating sistem linux. pada saat ini bagi para pemula bisa membuat sendiri sistem operasi linux dengan menggunakan Myslax creator. Myslax creator merupakan tool software yang digunakan untuk membuat sistem operasi berbasis linux sendiri. dengan Myslax creator sistem operasi yang dibuat dapat dimodifikasi sesuai keinginan dan kebutuhan. Untuk membuat distro linux dengan Myslax creator ini diperlukan tool software tambahan yaitu microsoft visual studio versi 2008 keatas dan file linux dengan format ISO yang digunakan sebagai installer, serta *tool software* VMware atau sejenisnya untuk menjalankan atau mencoba distro linux yang telah dibuat. dengan Myslax creator ini dapat pula ditambahkan beberapa software standart seperti mozilla untuk browsing internet, game dan beberapa software lain yang dibutuhkan. Mungkin untuk pemula dibidang komputer atau bukan seorang pekerja programmer, menggunakan sistem operasi sendiri yang telah dibuat pada komputer untuk bekerja dapat menimbulkan kesenangan tersendiri dan dapat meningkatkan semangat bekerja.

Saat ini bahkan sudah ada metode remaster dengan menggunakan wizard. Seperti MySlax untuk Slackware dan Revisor pada Fedora versi 7. Akan tetapi metode ini kurang bisa dikustomisasi dan memerlukan modul-modul siap instal. Sehingga metode yang akan dipakai dalam penelitian ini adalah remaster secara manual. Hal ini dimaksudkan agar dapat juga memasukkan paket-paket yang hanya tersedia dalam bentuk source code dan dapat melakukan kustomisasi lebih banyak.

#### **SOAL :**

1. Bagaimana cara memodifikasi linux slax yang sudah terinstal pada PC/laptop?
2. Jelaskan tentang kelebihan dan kekurangan menggunakan Linux Slax !

## **Bab 11**

### **Manajemen Memori**

#### **A. Latar Belakang**

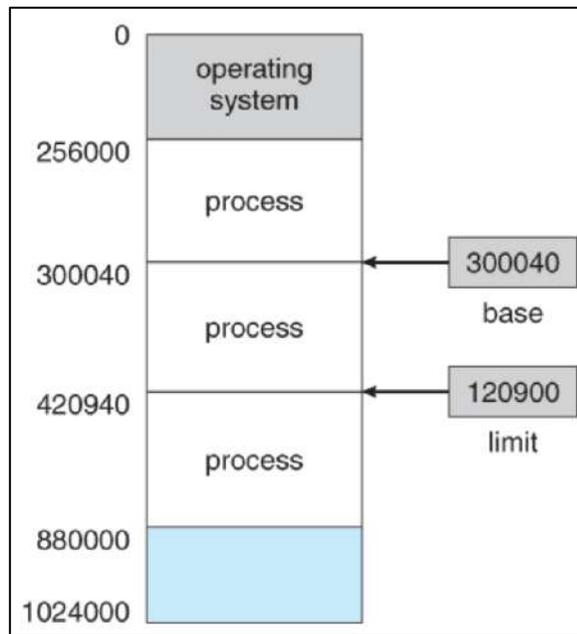
Akses memori dan manajemen memori jelas merupakan bagian yang sangat penting dari operasi komputer modern. Setiap instruksi harus diambil dari memori sebelum dapat dieksekusi, dan sebagian besar instruksi melibatkan pengambilan data dari memori atau menyimpan data dalam memori atau keduanya. Munculnya multi-tasking OS menambah kompleksitas manajemen memori, karena ketika proses ditukar keluar dan masuk dari CPU, kode dan data mereka harus ditukar keluar dan masuk memori, semuanya dengan kecepatan tinggi dan tanpa mengganggu proses lainnya. Memori bersama, memori virtual, klasifikasi memori sebagai read-only versus read-write, dan konsep-konsep seperti forking copy-on-write semua semakin memperumit masalah.

Perlu dicatat bahwa dari sudut pandang chip memori, semua akses memori setara. Perangkat keras memori tidak tahu untuk apa bagian tertentu dari memori itu digunakan. CPU hanya dapat mengakses register dan memori utamanya. Misalnya, membuat akses langsung ke hard drive, sehingga setiap data yang disimpan di sana harus ditransfer terlebih dahulu ke dalam chip memori utama sebelum CPU dapat bekerja dengannya. Driver perangkat berkomunikasi dengan perangkat keras mereka melalui interupsi dan akses "memori", mengirim instruksi singkat misalnya untuk mentransfer data dari hard drive ke lokasi yang ditentukan dalam memori utama. Pengontrol disk memantau bus untuk instruksi tersebut, mentransfer data, dan kemudian memberi tahu CPU bahwa data ada di sana dengan gangguan lain, tetapi CPU tidak pernah mendapatkan akses langsung ke disk. Akses memori ke register sangat cepat, umumnya satu clock tick, dan CPU mungkin dapat menjalankan lebih dari satu instruksi mesin per clock tick.

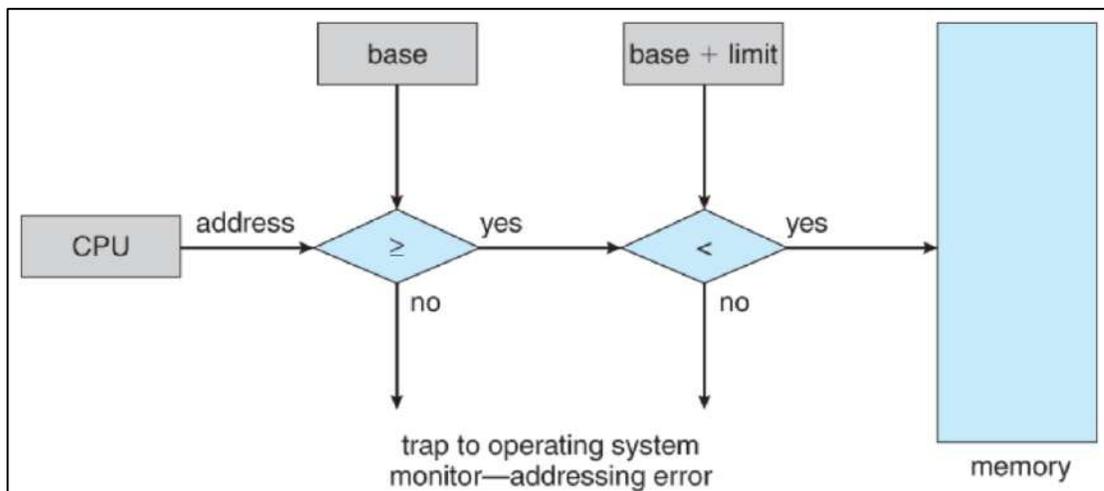
Akses memori ke memori utama relatif lambat, dan mungkin perlu beberapa detik untuk menyelesaikannya. Ini akan membutuhkan penantian yang tidak dapat ditoleransi oleh CPU jika bukan karena cache memori sebagai perantara yang dibangun ke dalam sebagian besar CPU modern. Ide dasar cache adalah untuk mentransfer potongan memori pada suatu waktu dari memori utama ke cache, dan kemudian untuk mengakses lokasi memori individual satu per satu dari cache.

Proses pengguna harus dibatasi sehingga mereka hanya mengakses lokasi memori yang dimiliki proses tertentu. Ini biasanya diimplementasikan menggunakan register dasar dan register batas untuk setiap proses, seperti yang ditunjukkan pada Gambar 11.1 dan 11.2 di bawah ini. Setiap akses memori yang dibuat oleh proses pengguna

diperiksa terhadap dua register ini, dan jika akses memori dicoba di luar rentang yang valid, maka dihasilkan kesalahan fatal. OS jelas memiliki akses ke semua lokasi memori yang ada, karena ini diperlukan untuk menukar kode dan data pengguna masuk dan keluar dari memori. Juga harus jelas bahwa mengubah isi register dasar dan limit adalah aktivitas istimewa, hanya diizinkan untuk kernel OS.



Gambar 11.1 Pembagian ruang alamat logika (*logical address*)



Gambar 11.2 Penggunaan register dasar dan batas untuk proteksi alamat fisik

## **B. Address Binding**

Program pengguna biasanya merujuk ke alamat memori dengan nama simbolis seperti "i", "count", dan "averageTemperature". Nama simbolis ini harus dipetakan atau diikat ke alamat memori fisik, yang biasanya terjadi dalam beberapa tahap:

### a) Waktu Kompilasi

Jika diketahui pada waktu kompilasi di mana suatu program akan berada dalam memori fisik, maka kode absolut dapat dihasilkan oleh kompiler, yang berisi alamat fisik aktual. Namun jika alamat muat berubah beberapa waktu kemudian, maka program harus dikompilasi ulang. Program DOS .COM menggunakan kompilasi yang mengikat waktu.

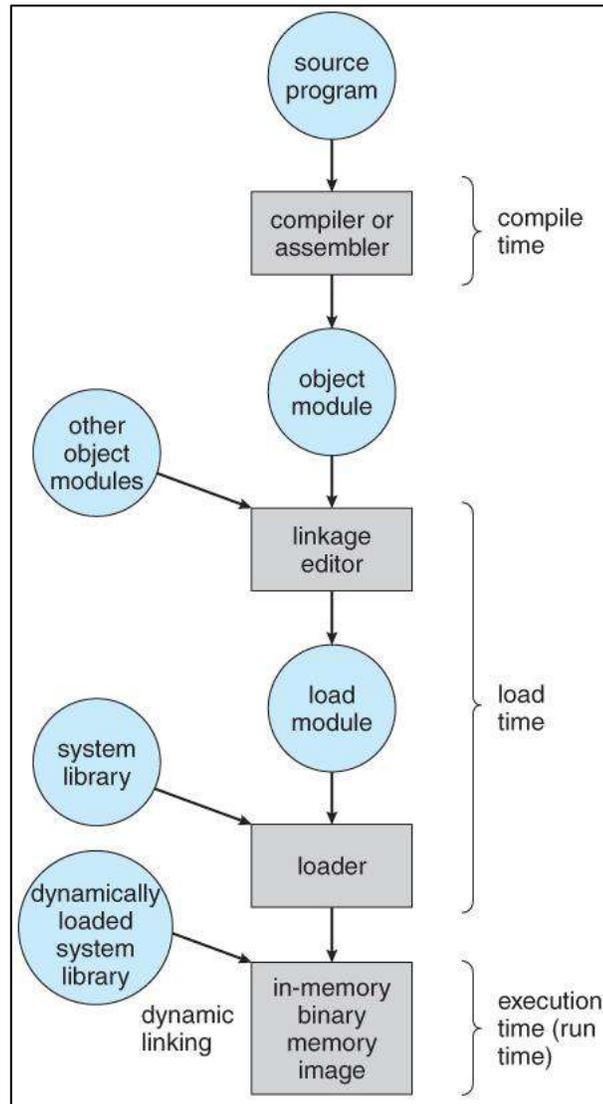
### b) Load Time

Jika lokasi di mana suatu program akan dimuat tidak diketahui pada waktu kompilasi, maka kompiler harus menghasilkan kode yang dapat dipindahkan, yang merujuk alamat relatif terhadap awal program. Jika alamat awal itu berubah, maka program harus dimuat ulang tetapi tidak dikompilasi ulang.

### c) Waktu Eksekusi

Jika suatu program dapat dipindahkan dalam memori selama pelaksanaannya, maka mengikat harus ditunda sampai waktu eksekusi. Ini membutuhkan perangkat keras khusus, dan merupakan metode yang diterapkan oleh sebagian besar OS modern.

Gambar 11.3 menunjukkan berbagai tahap proses pengikatan dan unit yang terlibat dalam setiap tahap.

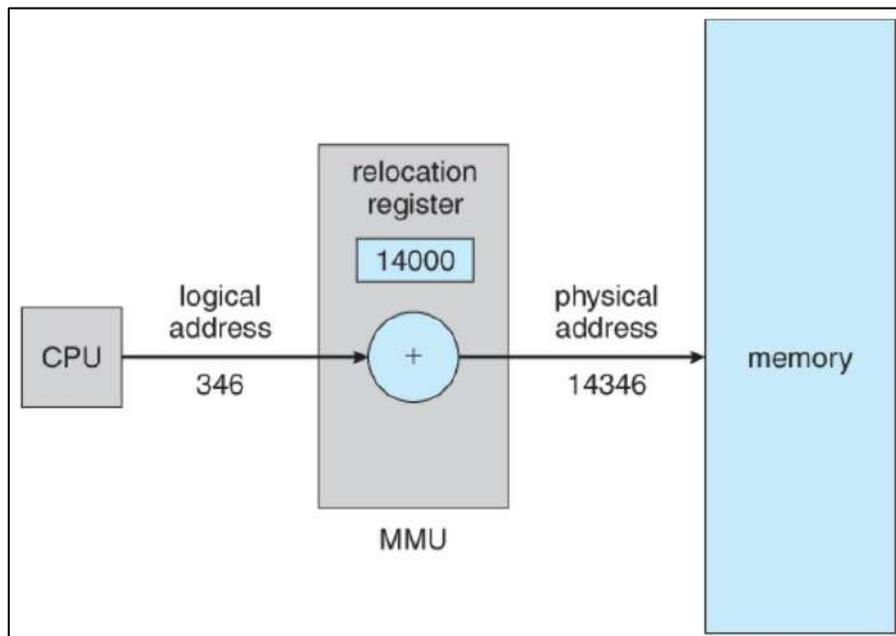


Gambar 11.3 Tahapan dalam pengikatan alamat

### C. Ruang Alamat Logika dan Ruang Alamat Fisik

Alamat yang dihasilkan oleh CPU adalah alamat logika, sedangkan alamat yang benar-benar dilihat oleh perangkat keras memori adalah alamat fisik. Alamat terikat pada waktu kompilasi atau waktu pengambilan memiliki alamat logis dan fisik yang identik. Namun, alamat yang dibuat pada waktu eksekusi memiliki alamat logis dan fisik yang berbeda.

Himpunan semua alamat logika yang digunakan oleh suatu program akan menyusun ruang alamat logika, dan himpunan semua alamat fisik yang sesuai akan menyusun ruang alamat fisik. Pemetaan run time dari alamat logika ke fisik ditangani oleh unit manajemen memori (MMU). MMU dapat mengambil banyak bentuk. Salah satu yang paling sederhana adalah modifikasi dari skema register-dasar yang dijelaskan sebelumnya. Register dasar sekarang disebut register relokasi, yang nilainya ditambahkan ke setiap permintaan memori di tingkat perangkat keras, seperti pada Gambar 11.4. Perhatikan bahwa program pengguna tidak pernah melihat alamat fisik. Program pengguna bekerja sepenuhnya dalam ruang alamat logika, dan setiap referensi atau manipulasi memori dilakukan menggunakan alamat logika murni. Hanya ketika alamat dikirim ke chip memori fisik adalah alamat memori fisik yang dihasilkan.



Gambar 11.4 Register relokasi

#### D. Dynamic Linking dan Shared Libraries

Dengan modul pustaka tautan statis yang sepenuhnya dimasukkan dalam modul yang dapat dieksekusi, membuang ruang disk dan penggunaan memori utama, karena setiap program yang menyertakan rutin tertentu dari perpustakaan harus memiliki salinan sendiri dari rutin yang terhubung ke kode yang dapat dieksekusi. Namun, dengan tautan dinamis, hanya tulisan rintisan yang ditautkan ke modul yang dapat dieksekusi, yang berisi referensi ke modul perpustakaan aktual yang ditautkan pada saat dijalankan.

Metode ini menghemat ruang disk, karena rutinitas perpustakaan tidak perlu sepenuhnya dimasukkan dalam modul yang dapat dieksekusi, hanya stubs. Manfaat tambahan dari pustaka yang terhubung secara dinamis (DLL, juga dikenal sebagai pustaka bersama atau objek bersama pada sistem UNIX) melibatkan peningkatan dan pembaruan yang mudah. Ketika suatu program menggunakan rutin dari pustaka standar dan perubahan rutin, maka program harus dibangun kembali (re-linked) untuk memasukkan perubahan. Namun jika DLL digunakan, maka selama rintisan tidak berubah, program dapat diperbarui hanya dengan memuat versi baru DLL ke sistem. Informasi versi dipertahankan di kedua program dan DLL, sehingga suatu program dapat menentukan versi tertentu dari DLL jika perlu. Dalam praktiknya, pertama kali sebuah program memanggil rutin DLL, stub akan mengenali fakta dan akan mengganti dirinya dengan rutin aktual dari pustaka DLL. Panggilan lebih lanjut ke rutin yang sama akan mengakses rutin secara langsung dan tidak dikenakan biaya akses rintisan.

## **E. Swapping**

Suatu proses harus dimuat ke dalam memori untuk dieksekusi. Jika tidak ada cukup memori yang tersedia untuk menjaga semua proses yang berjalan dalam memori pada saat yang sama, maka beberapa proses yang saat ini tidak menggunakan CPU mungkin memiliki memori mereka ditukar ke disk lokal cepat yang disebut backing store.

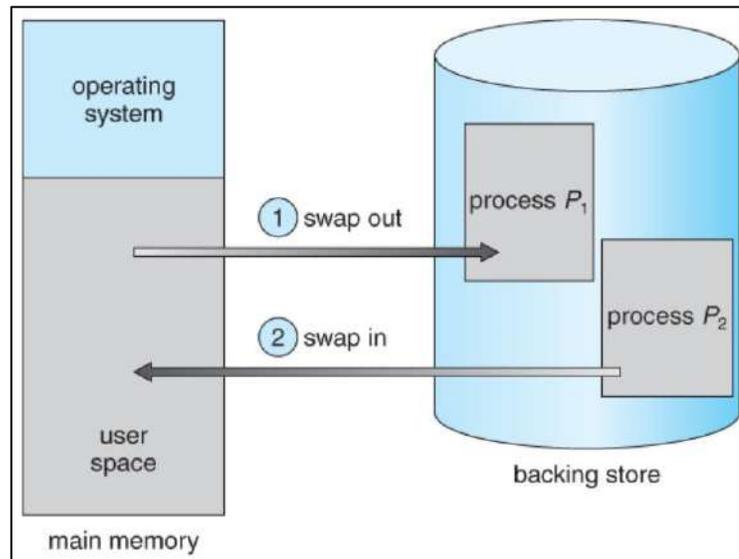
Jika kompilasi waktu atau pengikatan alamat waktu digunakan, maka proses harus ditukar kembali ke lokasi memori yang sama dari mana mereka bertukar. Jika mengikat waktu eksekusi digunakan, maka proses dapat ditukar kembali ke lokasi yang tersedia. Swapping adalah proses yang sangat lambat dibandingkan dengan operasi lain. Misalnya, jika proses pengguna menempati 10 MB dan kecepatan transfer untuk toko dukungan adalah 40 MB per detik, maka akan dibutuhkan 1/4 detik (250 milidetik) hanya untuk melakukan transfer data. Menambahkan jeda latensi 8 milidetik dan mengabaikan kepala mencari waktu untuk saat ini, dan lebih lanjut mengakui bahwa swapping melibatkan pemindahan data lama serta data baru, waktu transfer keseluruhan yang diperlukan untuk swap ini adalah 512 milidetik, atau lebih dari setengah kedua. Untuk penjadwalan prosesor yang efisien, irisan waktu CPU harus jauh lebih lama daripada waktu transfer yang hilang ini.

Untuk mengurangi swapping transfer overhead, diinginkan untuk mentransfer sesedikit mungkin informasi, yang mengharuskan sistem mengetahui berapa banyak memori yang digunakan suatu proses, berbeda dengan berapa banyak yang mungkin digunakan. Pemrogram dapat membantu dengan membebaskan memori dinamis yang tidak lagi mereka gunakan.

Sebuah hal yang penting untuk menukar proses dari memori hanya ketika mereka tidak digunakan, atau lebih tepatnya, hanya ketika tidak ada operasi I/O yang tertunda.

Jika tidak, operasi I / O yang tertunda dapat menulis ke dalam ruang memori proses yang salah. Solusinya adalah dengan menukar hanya proses yang benar-benar mengganggu, atau melakukan operasi I / O hanya masuk dan keluar dari buffer OS, yang kemudian ditransfer ke atau dari memori utama proses sebagai langkah kedua.

Sebagian besar OS modern tidak lagi menggunakan swapping, karena terlalu lambat dan ada alternatif yang lebih cepat tersedia, misalnya paging. Namun beberapa sistem UNIX masih akan meminta swapping jika sistem menjadi sangat penuh, dan kemudian menghentikan swapping ketika beban berkurang lagi. Windows 3.1 akan menggunakan versi modifikasi swapping yang agak dikontrol oleh pengguna, menukar proses jika perlu dan kemudian hanya menukar mereka kembali ketika pengguna fokus pada jendela tertentu. Contoh swapping pada dua proses ditunjukkan pada Gambar 11.4 berikut.



Gambar 11.5 Swapping dua proses

## F. Alokasi Memori

Salah satu metode mengalokasikan memori yang berdekatan adalah untuk membagi semua memori yang tersedia menjadi partisi berukuran sama, dan untuk menetapkan setiap proses ke partisi mereka sendiri. Ini membatasi jumlah proses simultan dan ukuran maksimum dari setiap proses, dan tidak lagi digunakan. Pendekatan alternatif adalah menyimpan daftar blok (lubang) memori yang tidak digunakan (gratis), dan menemukan lubang dengan ukuran yang sesuai setiap kali suatu proses perlu dimuat ke dalam memori. Ada banyak strategi berbeda untuk menemukan alokasi memori "terbaik" untuk proses, termasuk tiga yang paling umum dibahas:

### 1) First Fit

Mencari daftar lubang sampai ditemukan cukup besar untuk memenuhi permintaan, dan menetapkan sebagian lubang itu untuk proses itu. Fraksi apa pun dari lubang yang tidak diperlukan oleh permintaan dibiarkan pada daftar gratis sebagai lubang yang lebih kecil. Permintaan selanjutnya dapat mulai mencari dari awal daftar atau dari titik di mana pencarian ini berakhir.

### 2) Best Fit

Mengalokasikan lubang terkecil yang cukup besar untuk memenuhi permintaan. Ini menghemat lubang besar untuk permintaan proses lain yang mungkin membutuhkannya nanti, tetapi bagian yang tidak terpakai dari lubang mungkin terlalu kecil untuk digunakan, dan karenanya akan sia-sia. Menjaga daftar bebas diurutkan dapat mempercepat proses menemukan lubang yang tepat.

### 3) Worst fit

Mengalokasikan lubang terbesar yang tersedia, sehingga meningkatkan kemungkinan bagian yang tersisa dapat digunakan untuk memenuhi permintaan di masa depan.

Simulasi menunjukkan bahwa baik First Fit atau Best Fit lebih baik daripada Worst Fit dalam hal penggunaan waktu dan penyimpanan. First Fit dan Best Fit hampir sama dalam hal pemanfaatan penyimpanan, tetapi First Fit lebih cepat.

## **G. Fragmentasi**

Semua strategi alokasi memori mengalami fragmentasi eksternal, meskipun yang pertama dan paling cocok mengalami masalah lebih dari yang paling buruk. Fragmentasi eksternal berarti bahwa memori yang tersedia dipecah menjadi banyak bagian-bagian kecil, tidak ada yang cukup besar untuk memenuhi kebutuhan memori berikutnya, walaupun jumlah totalnya bisa. Jumlah memori yang hilang karena fragmentasi dapat bervariasi dengan algoritma, pola penggunaan, dan beberapa keputusan desain seperti ujung mana yang harus dialokasikan dan yang ujungnya disimpan pada daftar.

Analisis statistik dari First Fit, misalnya, menunjukkan bahwa untuk  $N$  blok memori yang dialokasikan,  $0,5 N$  lainnya akan hilang karena fragmentasi. Fragmentasi internal juga terjadi, dengan semua strategi alokasi memori. Ini disebabkan oleh fakta bahwa memori dialokasikan dalam blok-blok dengan ukuran yang tetap, sedangkan memori aktual yang dibutuhkan jarang akan sebesar itu. Untuk distribusi acak

permintaan memori, rata-rata 1/2 blok akan terbuang per permintaan memori, karena rata-rata blok yang dialokasikan terakhir hanya setengah penuh. Perhatikan bahwa efek yang sama terjadi pada hard drive, dan bahwa perangkat keras modern memberi kita semakin besar drive dan memori dengan mengorbankan ukuran blok yang semakin besar, yang diterjemahkan menjadi lebih banyak memori yang hilang karena fragmentasi internal.

Beberapa sistem menggunakan blok ukuran variabel untuk meminimalkan kerugian akibat fragmentasi internal. Jika program dalam memori dapat dipindahkan, menggunakan pengikatan alamat waktu eksekusi, maka masalah fragmentasi eksternal dapat dikurangi melalui pemadatan, misalnya memindahkan semua proses ke satu ujung memori fisik. Ini hanya melibatkan register relokasi untuk setiap proses, karena semua pekerjaan internal dilakukan menggunakan alamat logika. Solusi lain seperti yang akan kita lihat di bagian yang akan datang adalah untuk memungkinkan proses menggunakan blok memori fisik yang tidak bersebelahan, dengan register relokasi yang terpisah untuk setiap blok.

## **H. Segmentasi**

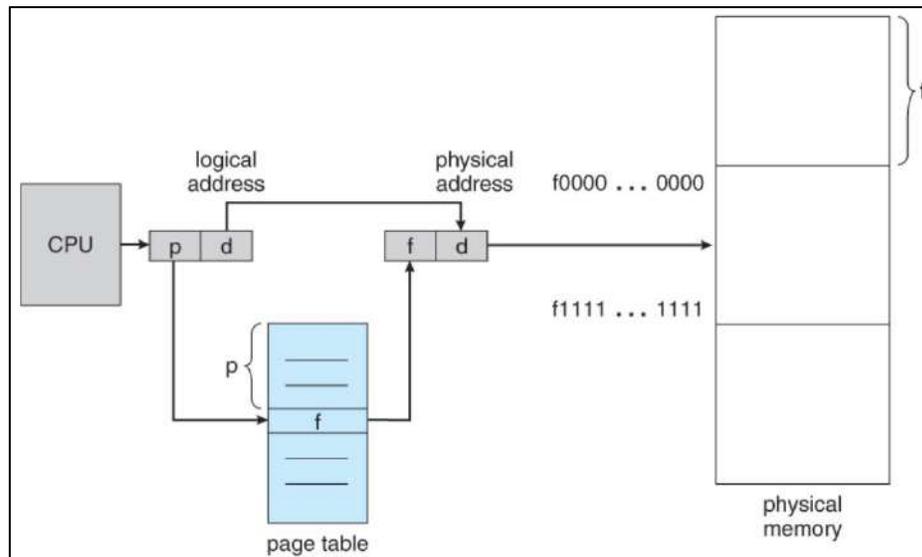
Sebagian besar pengguna (pemrogram) tidak menganggap program mereka ada dalam satu ruang alamat linear kontinu. Sebaliknya mereka cenderung memikirkan ingatan mereka dalam banyak segmen, masing-masing didedikasikan untuk penggunaan tertentu, seperti kode, data, tumpukan, tumpukan, dll. Segmentasi memori mendukung tampilan ini dengan menyediakan alamat dengan nomor segmen (dipetakan ke alamat basis segmen) dan offset dari awal segmen itu. Sebagai contoh, kompiler C dapat menghasilkan 5 segmen untuk kode pengguna, kode perpustakaan, variabel global (statis), tumpukan, dan tumpukan

## **I. Paging**

Paging adalah skema manajemen memori yang memungkinkan proses memori fisik terputus, dan yang menghilangkan masalah dengan fragmentasi dengan mengalokasikan memori dalam blok berukuran sama yang dikenal sebagai halaman. Paging menghilangkan sebagian besar masalah dari metode lain yang dibahas sebelumnya, dan merupakan teknik manajemen memori yang dominan digunakan saat ini.

Ide dasar di balik paging adalah untuk membagi memori fisik menjadi sejumlah blok berukuran sama yang disebut bingkai, dan untuk membagi ruang memori logis program menjadi blok dengan ukuran yang sama yang disebut halaman. Halaman apa pun (dari proses apa pun) dapat ditempatkan ke dalam bingkai apa pun yang tersedia.

Tabel halaman digunakan untuk mencari bingkai apa yang disimpan halaman tertentu saat ini. Dalam contoh berikut, misalnya, halaman 2 dari memori logis program saat ini disimpan dalam bingkai 3 dari memori fisik, seperti disajikan pada Gambar 11.6 berikut.



Gambar 11.6 Proses paging

Alamat logika terdiri dari dua bagian: Nomor halaman tempat alamat berada, dan offset dari awal halaman itu. Jumlah bit dalam jumlah halaman membatasi berapa banyak halaman yang dapat ditangani oleh satu proses. Jumlah bit dalam offset menentukan ukuran maksimum setiap halaman, dan harus sesuai dengan ukuran frame sistem.

Tabel halaman memetakan nomor halaman ke nomor bingkai, untuk menghasilkan alamat fisik yang juga memiliki dua bagian: Nomor bingkai dan offset dalam bingkai itu. Jumlah bit dalam jumlah frame menentukan berapa banyak frame yang dapat ditangani sistem, dan jumlah bit dalam offset menentukan ukuran setiap frame.

Nomor halaman, nomor bingkai, dan ukuran bingkai ditentukan oleh arsitektur, tetapi biasanya kekuatan dua, memungkinkan alamat untuk dibagi pada sejumlah bit. Misalnya, jika ukuran alamat logis adalah  $2^m$  dan ukuran halaman adalah  $2^n$ , maka bit  $m-n$  orde tinggi dari alamat logis menunjuk nomor halaman dan  $n$  bit yang tersisa mewakili offset.

Perhatikan juga bahwa jumlah bit dalam jumlah halaman dan jumlah bit dalam jumlah frame tidak harus sama. Yang pertama menentukan rentang alamat dari ruang alamat logis, dan yang terakhir berhubungan dengan ruang alamat fisik.

## **J. Memory Manager**

Memory Manager, merupakan Salah satu bagian system operasi yang dipengaruhi dalam menentukan proses mana yang diletakkan pada antrian.

Jenis-jenis Memori:

- Memori Kerja
  - ROM/PROM/EPROM/EEPROM
  - RAM
  - Cache memory
- Memory Dukung
  - Floppy, harddisk,CD,dll.

Alamat Memori:

- Alamat memori mutlak (alamat fisik)
- Alamat memori relative (alamat logika)
- Hubungan antara alamat mutlak dan alamat relative
- Jenis memori dan alamat memori

Isi Memori :

- Sistem bahasa penataolahan
- Sistem Utilitas
- Inti Sistem Operasi
- Sistem Operasi
- Pengendali alat (device drivers)
- File pemakai

## **K. Fungsi manajemen memori**

Fungsi manajemen memori adalah sebagai berikut.

- Mengelola informasi yang dipakai dan tidak dipakai
- Mengalokasikan memori ke proses yang memerlukan
- Mendelokasikan memori dari proses telah selesai
- Mengelola swapping atau paging antara memori utama dan disk

Manajemen Memori berdasarkan keberadaan Swapping atau Paging terbagi menjadi dua, yaitu:

1. Manajemen tanpa swapping atau paging
2. Manajemen dengan swapping atau paging

Memori tanpa Swapping atau Paging yaitu manajemen memori tanpa pemindahan citra proses antara memori utama dan disk selama eksekusi.

Terdiri dari:

- *Monoprogramming*  
Ciri-ciri:
  - Hanya satu proses pada satu saat
  - Hanya satu proses menggunakan semua memori
  - Pemakai memuatkan program ke seluruh memori dari disk atau tape
  - Program mengambil kendali seluruh mesin
- *Multiprogramming* dengan pemartisian statis  
Terbagi menjadi dua, yaitu:
  1. Pemartisian menjadi partisi-partisi berukuran sama, yaitu ukuran semua partisi memori adalah sama
  2. Pembagian partisi menjadi partisi-partisi berukuran berbeda, yaitu ukuran semua partisi memori adalah berbeda.

#### L. Strategi Alokasi Memori

- *First fit algorithm*: memory manager men-scan list untuk menemukan hole yang cukup untuk menampung proses yang baru. Proses akan menempati hole pertama yang ditemuinya yang cukup untuk dirinya.
- *Next fit algorithm*: sama dengan first fit, tetapi pencarian hole dimulai dari hole ditemuinya dari scan sebelumnya.
- *Best fit algorithm*: dicari hole yang akan menghasilkan sisa paling sedikit setelah dimasuki proses.
- *Worst fit algorithm*: kebalikan dari best fit.
- *Quick fit algorithm*: mengelompokkan hole-hole dan membuat listnya sendiri. Misalnya, ada list untuk hole 4K, satu list 8K, dst.

Dari berbagai cara alokasi tersebut di atas, sebuah hole yang ditempati proses akan terbagi menjadi bagian yang dipakai proses dan memori yang tidak terpakai (fragmen). Timbulnya memori yang tidak terpakai disebut fragmentasi.

Ada dua macam fragmen:

- Internal: sisa hole yang tidak terpakai setelah terisi proses.
- Eksternal: hole yang secara utuh terlalu kecil untuk dipakai oleh proses manapun.

#### SOAL LATIHAN:

1. Jelaskan tentang konsep *swapping*!
2. Jelaskan tentang konsep fragmentasi!
3. Jelaskan tentang perbedaan internal dan eksternal fragmentation!
4. Bagaimana cara mengalokasikan memori?

## Bab 12

### Manajemen Input/Output (I/O)

#### A. Pooling

*Busy-waiting/polling* adalah ketika *host* mengalami *looping* yaitu membaca status register secara terus-menerus sampai status *busy* di-clear.

Pada dasarnya *polling* dapat dikatakan efisien. Akan tetapi *polling* menjadi tidak efisien ketika setelah berulang-ulang melakukan *looping*, hanya menemukan sedikit *device* yang siap untuk *service*, karena CPU *processing* yang tersisa belum selesai.

#### B. Interupsi

Mekanisme Dasar Interupsi :

Ketika CPU mendeteksi bahwa sebuah *controller* telah mengirimkan sebuah sinyal ke interrupt request line (membangkitkan sebuah interupsi), CPU kemudian menjawab interupsi tersebut (juga disebut menangkap interupsi) dengan menyimpan beberapa informasi mengenai *state* terkini CPU- contohnya nilai instruksi *pointer* dan memanggil *interrupt handler* agar handler tersebut dapat melayani controller atau alat yang mengirim interupsi tersebut.

Fitur Tambahan pada Komputer Modern :

Pada arsitektur komputer modern, tiga fitur disediakan oleh CPU dan *interrupt controller* (pada perangkat keras) untuk dapat menangani interupsi dengan lebih bagus. Fitur-fitur ini antara lain adalah kemampuan menghambat sebuah proses *interrupt handling* selama prosesi berada dalam *critical state*, efisiensi penanganan interupsi sehingga tidak perlu dilakukan *polling* untuk mencari *device* yang mengirimkan interupsi, dan fitur yang ketiga adalah adanya sebuah konsep *multilevel* interupsi sedemikian rupa sehingga terdapat prioritas dalam penanganan interupsi (diimplementasikan dengan *interrupt priority level system*).

*Interrupt Request Line*

Pada peranti keras CPU terdapat kabel yang disebut *interrupt request line*, kebanyakan CPU memiliki dua macam *interrupt request line*, yaitu *nonmaskable interrupt* dan *maskable interrupt*. *Maskable interrupt* dapat dimatikan/ dihentikan oleh CPU sebelum pengeksesian

deretan *critical instruction (critical instruction sequence)* yang tidak boleh diinterupsi. Biasanya, interrupt jenis ini digunakan oleh *device controller* untuk meminta pelayanan CPU.

#### *Interrupt Vector dan Interrupt Chaining*

Sebuah mekanisme interupsi akan menerima alamat *interrupt handling routine* yang spesifik dari sebuah set, pada kebanyakan arsitektur komputer yang ada sekarang ini, alamat ini biasanya berupa sekumpulan bilangan yang menyatakan *offset* pada sebuah tabel (biasa disebut *interrupt vector*). Tabel ini menyimpan alamat-alamat *interrupt handler* spesifik di dalam memori. Keuntungan dari pemakaian vektor adalah untuk mengurangi kebutuhan akan sebuah *interrupt handler* yang harus mencari semua kemungkinan sumber interupsi untuk menemukan pengirim interupsi.

#### Penyebab Interupsi

Interupsi dapat disebabkan berbagai hal, antara lain *exception, page fault*, interupsi yang dikirimkan oleh *device controllers*, dan *system call Exception* adalah suatu kondisi dimana terjadi sesuatu/ dari sebuah operasi didapat hasil tertentu yang dianggap khusus sehingga harus mendapat perhatian lebih, contohnya pembagian dengan 0 (nol), pengaksesan alamat memori yang *restricted* atau bahkan tidak valid, dan lain-lain.

*System call* adalah sebuah fungsi pada aplikasi (perangkat lunak) yang dapat mengeksekusikan instruksi khusus berupa *software interrupt* atau *trap*.

### C. DMA

DMA adalah sebuah prosesor khusus (*special purpose processor*) yang berguna untuk menghindari pembebanan CPU utama oleh program I/O (PIO).

#### a. Transfer DMA

Untuk memulai sebuah transfer DMA, *host* akan menuliskan sebuah *DMA command block* yang berisi *pointer* yang menunjuk ke sumber transfer, *pointer* yang menunjuk ke tujuan/ destinasi transfer, dan jumlah *byte* yang ditransfer, ke memori. CPU kemudian menuliskan alamat *command block* ini ke *DMA controller*, sehingga *DMA controller* dapat kemudian mengoperasikan *bus* memori secara langsung dengan menempatkan alamat-alamat pada *bus* tersebut untuk melakukan transfer tanpa bantuan CPU. Tiga langkah dalam transfer DMA: Prosesor menyiapkan DMA transfer dengan menyediakan data-data dari *device*, operasi yang akan ditampilkan, alamat memori yang menjadi sumber dan tujuan data, dan banyaknya *byte* yang di transfer.

*DMA controller* memulai operasi (menyiapkan bus, menyediakan alamat, menulis dan membaca data), sampai seluruh blok sudah di transfer.

DMA *controller* meng-interupsi prosesor, dimana selanjutnya akan ditentukan tindakan berikutnya.

Pada dasarnya, DMA mempunyai dua metode yang berbeda dalam mentransfer data. Metode yang pertama adalah metode yang sangat baku dan simple disebut HALT, atau *Burst Mode* DMA, karena DMA *controller* memegang kontrol dari sistem bus dan mentransfer semua blok data ke atau dari memori pada *single burst*. Selagi transfer masih dalam progres, sistem mikroprosesor di-*setidle*, tidak melakukan instruksi operasi untuk menjaga internal register. Tipe operasi DMA seperti ini ada pada kebanyakan komputer.

Metode yang kedua, mengikut-sertakan DMA *controller* untuk memegang kontrol dari sistem bus untuk jangka waktu yang lebih pendek pada periode dimana mikroprosesor sibuk dengan operasi internal dan tidak membutuhkan akses ke sistem bus. Metode DMA ini disebut *cycle stealing mode*. *Cycle stealing* DMA lebih kompleks untuk diimplementasikan dibandingkan HALT DMA, karena DMA *controller* harus mempunyai kepintaran untuk merasakan waktu pada saat sistem bus terbuka.

b. *Handshaking*

Proses *handshaking* antara DMA *controller* dan *device controller* dilakukan melalui sepasang kabel yang disebut DMA-*request* dan DMA-*acknowledge*. Device *controller* mengirimkan sinyal melalui DMA-*request* ketika akan mentransfer data sebanyak satu *word*. Hal ini kemudian akan mengakibatkan DMA *controller* memasukkan alamat-alamat yang diinginkan ke kabel alamat memori, dan mengirimkan sinyal melalui kabel DMA-*acknowledge*. Setelah sinyal melalui kabel DMA-*acknowledge* diterima, *device controller* mengirimkan data yang dimaksud dan mematikan sinyal pada DMA-*request*.

Hal ini berlangsung berulang-ulang sehingga disebut *handshaking*. Pada saat DMA *controller* mengambil alih memori, CPU sementara tidak dapat mengakses memori (dihalangi), walau pun masih dapat mengakses data pada cache primer dan sekunder. Hal ini disebut *cycle stealing*, yang walau pun memperlambat komputasi CPU, tidak menurunkan kinerja karena memindahkan pekerjaan data transfer ke DMA *controller* meningkatkan performa sistem secara keseluruhan.

c. Cara-cara Implementasi DMA

Dalam pelaksanaannya, beberapa komputer menggunakan memori fisik untuk proses DMA, sedangkan jenis komputer lain menggunakan alamat virtual dengan melalui tahap “penerjemahan” dari alamat memori virtual menjadi alamat memori fisik, hal ini disebut *direct virtual-memory address* atau DVMA.

Keuntungan dari DVMA adalah dapat mendukung transfer antara dua *memory mapped device* tanpa intervensi CPU.

**SOAL LATIHAN :**

1. Jelaskan tentang konsep pooling !
2. Jelaskan tentang direct memory access !
3. Jelaskan perbedaan antara DMA dan DVMA !

## Bab 13

### Deadlock

#### A. Konsep Deadlock

*Deadlock*, adalah Suatu kondisi dimana proses tidak berjalan lagi atau pun tidak ada komunikasi lagi antar proses. *Deadlock* disebabkan karena proses yang satu menunggu sumber daya yang sedang dipegang oleh proses lain yang sedang menunggu sumber daya yang dipegang oleh proses tersebut.

Untuk keperluan pembahasan *deadlock*, suatu sistem dapat dimodelkan sebagai kumpulan sumber daya terbatas, yang dapat dipartisi ke dalam kategori yang berbeda, untuk dialokasikan ke sejumlah proses, masing-masing memiliki kebutuhan yang berbeda. Kategori sumber daya dapat mencakup memori, printer, CPU, file terbuka, tape drive, CD-ROM, dll. Menurut definisi, semua sumber daya dalam kategori adalah setara, dan permintaan kategori ini dapat sama-sama dipenuhi oleh salah satu sumber daya dalam kategori itu. Jika ini bukan masalahnya (yaitu jika ada beberapa perbedaan antara sumber daya dalam suatu kategori), maka kategori itu perlu dibagi lagi menjadi beberapa kategori yang terpisah. Misalnya, "printer" mungkin perlu dipisahkan menjadi "printer laser" dan "printer inkjet warna". Beberapa kategori mungkin memiliki satu sumber daya.

Dalam operasi normal suatu proses harus meminta sumber daya sebelum menggunakannya, dan melepaskannya ketika selesai, dalam urutan berikut:

1) Request

Jika permintaan tidak dapat segera dikabulkan, maka proses harus menunggu sampai sumber daya yang dibutuhkan tersedia. Misalnya panggilan sistem `open()`, `malloc()`, `new()`, dan `request()`.

2) Use

Proses menggunakan sumber daya, mis. mencetak ke printer atau membaca dari file.

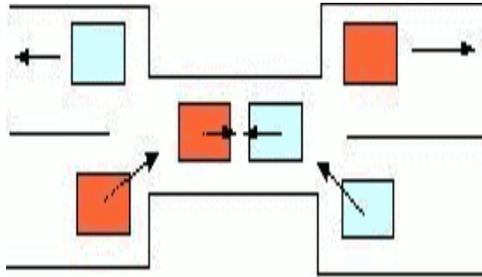
3) Release

Proses melepaskan sumber daya. sehingga menjadi tersedia untuk proses lain. Misalnya, `tutup()`, `gratis()`, `hapus()`, dan `lepaskan()`.

Untuk semua sumber daya yang dikelola kernel, kernel melacak sumber daya apa yang gratis dan yang dialokasikan, ke proses mana mereka dialokasikan, dan antrian proses menunggu sumber daya ini tersedia. Sumber daya yang dikelola aplikasi dapat dikontrol menggunakan mutex atau panggilan `wait()` dan `signal()`, misal biner atau penghitungan semafor.

Seperangkat proses menemui jalan buntu ketika setiap proses dalam set sedang menunggu sumber daya yang saat ini dialokasikan untuk proses lain dalam set (dan yang hanya bisa dirilis ketika proses menunggu lainnya membuat kemajuan).

Contoh deadlock:



**Gambar 13.1 Contoh Deadlock**

Ada 4 kondisi yang menyebabkan *deadlock*:

- Mutual Eksklusif
- Memegang dan menunggu
- Tidak ada preemption
- Circular wait

Cara menanggulangi masalah *deadlock*:

1. Mengabaikan masalah *deadlock*
2. Mendeteksi dan memperbaiki
3. *Deadlock avoidance* sistem
4. *Deadlock prevention* sistem

Untuk mencegah *deadlock* dilakukan dengan meniadakan salah satu dari syarat perlu sebagai berikut :

- **Mencegah *Mutual Exclusion***  
*Mutual exclusion* benar-benar tak dapat dihindari. Hal ini dikarenakan tidak ada sumber daya yang dapat digunakan bersama-sama, jadi sistem harus membawa sumber daya yang tidak dapat digunakan bersama-sama
- **Mencegah *Hold and Wait***  
Untuk mencegah hold and wait, sistem harus menjamin bila suatu proses meminta sumber daya, maka proses tersebut tidak sedang memegang sumber daya yang lain. Proses harus meminta dan dialokasikan semua sumber daya yang diperlukan sebelum proses memulai eksekusi atau mengijinkan proses meminta sumber daya

hanya jika proses tidak membawa sumber daya lain. Model ini mempunyai utilitas sumber daya yang rendah dan kemungkinan terjadi *starvation* jika proses membutuhkan sumber daya yang populer sehingga terjadi keadaan menunggu yang tidak terbatas karena setidaknya satu dari sumber daya yang dibutuhkannya dialokasikan untuk proses yang lain.

- **Mencegah Non Preemption**

Peniadaan *non preemption* mencegah proses-proses lain harus menunggu. Seluruh proses menjadi *preemption*, sehingga tidak ada tunggu menunggu. Cara mencegah kondisi *non preemption* :

- Jika suatu proses yang membawa beberapa sumber daya meminta sumber daya lain yang tidak dapat segera dipenuhi untuk dialokasikan pada proses tersebut, maka semua sumber daya yang sedang dibawa proses tersebut harus dibebaskan.
- Proses yang sedang dalam keadaan menunggu, sumber daya yang dibawanya ditunda dan ditambahkan pada daftar sumber daya.
- Proses akan di restart hanya jika dapat memperoleh sumber daya yang lama dan sumber daya baru yang diminta.

- **Mencegah Kondisi Menunggu Sirkular**

Sistem mempunyai total permintaan global untuk semua tipe sumber daya. Proses dapat meminta proses kapanpun menginginkan, tapi permintaan harus dibuat terurut secara numerik. Setiap proses yang membutuhkan sumber daya dan memintanya maka nomor urut akan dinaikkan. Cara ini tidak akan menimbulkan siklus. Masalah yang timbul adalah tidak ada cara pengurutan nomor sumber daya yang memuaskan semua pihak.

Cara mendeteksi *deadlock* diantaranya adalah :

Jika sistem tidak menyediakan algoritma mencegah *deadlock* dan menghindari *deadlock*, maka terjadi *deadlock*. Pada lingkungan ini sistem harus menyediakan :

- Algoritma yang menguji state sistem untuk menentukan apakah *deadlock* telah terjadi.
- Algoritma untuk memperbaiki dari *deadlock*.

Jika semua sumber daya hanya mempunyai satu anggota, kita dapat menentukan algoritma mendeteksi *deadlock* menggunakan bentuk *resource allocation graph* yang disebut *wait-for graph*.

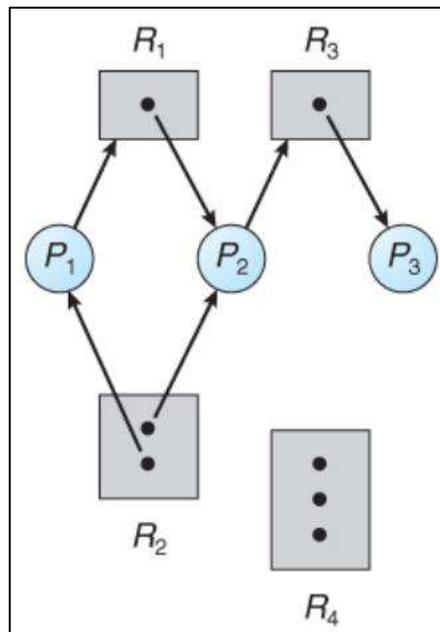
## B. Resource Allocation Graph

Dalam beberapa kasus, kebuntuan dapat dipahami lebih jelas melalui penggunaan Resource Allocation Graph, yang memiliki sifat-sifat berikut:

- Satu set kategori sumber daya,  $\{R_1, R_2, R_3, \dots, R_N\}$ , yang muncul sebagai node kuadrat pada grafik. Titik-titik di dalam node sumber daya menunjukkan contoh spesifik sumber daya, misalnya dua titik mungkin mewakili dua printer laser.
- Seperangkat proses,  $\{P_1, P_2, P_3, \dots, P_N\}$
- Request Edges - Serangkaian busur diarahkan dari  $P_i$  ke  $R_j$ , menunjukkan bahwa proses  $P_i$  telah meminta  $R_j$ , dan saat ini sedang menunggu sumber daya itu tersedia.
- Assignment Edges - Serangkaian busur diarahkan dari  $R_j$  ke  $P_i$  menunjukkan bahwa sumber daya  $R_j$  telah dialokasikan untuk memproses  $P_i$ , dan bahwa  $P_i$  saat ini memegang sumber daya  $R_j$ .

Perhatikan bahwa tepi permintaan dapat dikonversi menjadi tepi penugasan dengan membalik arah busur saat permintaan diberikan. Namun perhatikan juga bahwa ujung permintaan menunjuk ke kotak kategori, sedangkan tepi penetapan berasal dari titik instance tertentu di dalam kotak.

Sebagai contoh, pada Gambar 13.2 berikut.

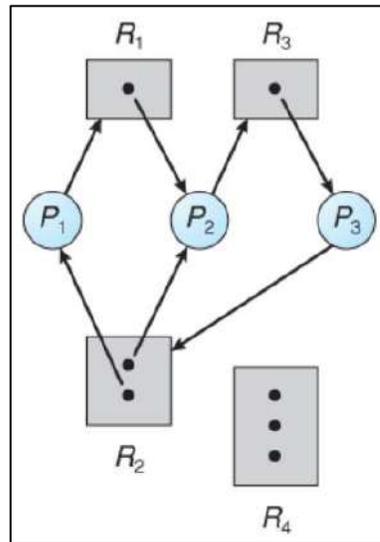


Gambar 13.2 Resource allocation graph

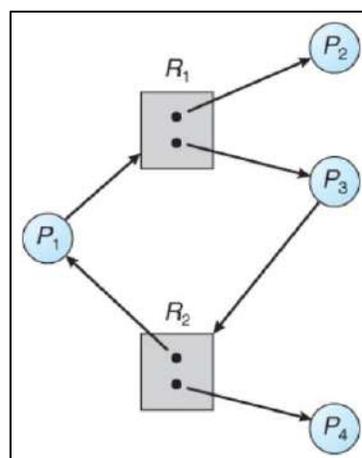
Jika RAG tidak mengandung siklus, maka sistem tidak menemui jalan buntu. Saat mencari siklus, ingatlah bahwa ini adalah grafik berarah. Seperti pada Gambar 13.2 di atas.

Jika RAG memang mengandung siklus dan setiap kategori sumber daya hanya berisi satu kejadian tunggal, maka ada jalan buntu (deadlock).

Jika kategori sumber daya berisi lebih dari satu contoh, maka kehadiran siklus dalam grafik alokasi sumber daya menunjukkan kemungkinan deadlock, tetapi tidak menjamin satu. Perhatikan Gambar 13.3 dan 13.4 berikut.



Gambar 13.3 Resource allocation graph dengan deadlock



Gambar 13.4 Resource allocation graph dengan siklus tapi tidak deadlock

### C. Metode Penanganan Deadlock

Secara umum ada tiga cara menangani deadlock:

- 1) Deadlock Prevention or avoidance. Jangan biarkan sistem masuk ke kondisi menemui jalan buntu.
- 2) Deadlock detection and recovery. Membatalkan proses atau mendahului beberapa sumber daya saat kebuntuan terdeteksi.
- 3) Ignore the problem all together - Jika kebuntuan hanya terjadi setahun sekali atau lebih, mungkin lebih baik membiarkannya terjadi dan reboot sesuai kebutuhan daripada menanggung denda konstan dan hukuman kinerja sistem yang terkait dengan pencegahan atau deteksi kebuntuan. Ini adalah pendekatan yang diambil oleh Windows dan UNIX.

Untuk menghindari deadlock, sistem harus memiliki informasi tambahan tentang semua proses. Secara khusus, sistem harus tahu sumber daya apa yang akan atau mungkin diminta oleh suatu proses di masa depan.

Deteksi deadlock cukup mudah, tetapi pemulihan deadlock membutuhkan proses pembatalan atau preempting sumber daya, yang keduanya bukan merupakan alternatif yang menarik.

Jika kebuntuan tidak dicegah atau terdeteksi, maka ketika deadlock terjadi, sistem secara bertahap akan melambat, karena semakin banyak proses menjadi macet menunggu sumber daya yang saat ini dipegang oleh kebuntuan dan oleh proses menunggu lainnya. Sayangnya perlambatan ini dapat dibedakan dari perlambatan sistem umum ketika proses real-time memiliki kebutuhan komputasi yang berat.

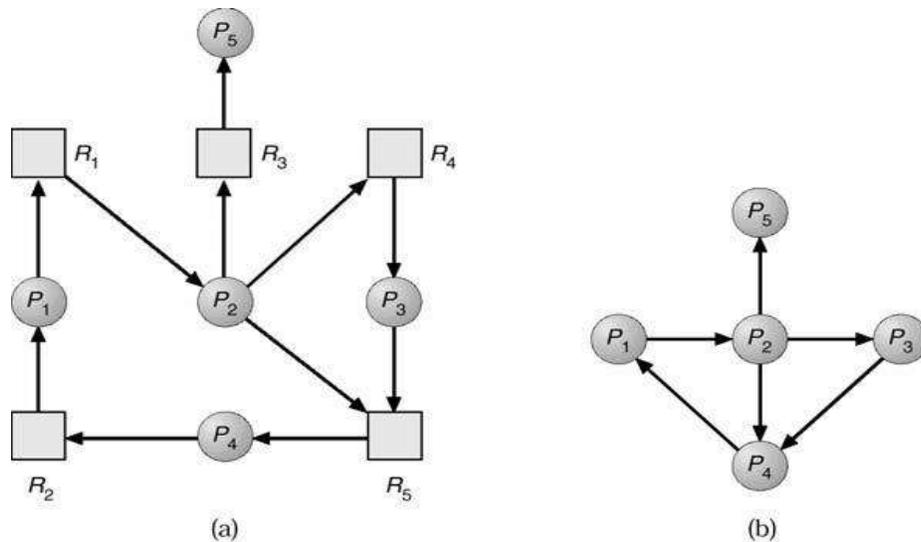
### D. Deteksi Deadlock

Jika deadlock tidak dihindari, maka pendekatan lain adalah mendeteksi kapan mereka telah terjadi dan entah bagaimana pulih. Selain hit kinerja dari terus-menerus memeriksa deadlock, kebijakan/algorithm harus ada untuk pulih dari deadlock, dan ada potensi untuk pekerjaan yang hilang ketika proses harus dibatalkan atau sumber daya mereka diambil terlebih dahulu.

Garis dari  $P_i \rightarrow P_j$  pada *wait-for graph* menandakan bahwa proses  $P_i$  menunggu

$P_j$  melepaskan sumber daya yang dibutuhkan  $P_i$ . Garis  $P_i \rightarrow P_j$  terdapat pada *wait-for graph* jika dan hanya jika *resource allocation graph* berisi dua garis  $P_i \rightarrow R_q$  dan  $R_q \rightarrow P_j$  untuk beberapa sumber daya  $R_q$  seperti gambar 13.5.

Secara periodik sistem menggunakan algoritma yang mencari siklus pada graph. Algoritma untuk mendeteksi siklus pada graph membutuhkan operasi  $n^2$  dimana  $n$  adalah jumlah titik pada graph.



Gambar 13.5(a) Resource allocation graph (b) Wait-for graph

### E. Perbaikan dari *Deadlock*

Terdapat dua pilihan untuk membebaskan *deadlock*. Satu solusi sederhana adalah dengan menghentikan satu atau beberapa proses untuk membebaskan kondisi menunggu sirkular. Pilihan kedua adalah menunda beberapa sumber daya dari satu atau lebih proses yang *deadlock*.

#### Terminasi Proses

Untuk memperbaiki *deadlock* dengan terminasi proses, dapat digunakan salah satu dari dua metode di bawah ini:

- Menghentikan (*abort*) semua proses yang *deadlock*.
- Menghentikan satu proses setiap waktu sampai siklus *deadlock* hilang.

Untuk menentukan urutan proses yang harus dihentikan ada beberapa faktor yang harus diperhatikan :

- Prioritas proses.
- Berapa lama proses dijalankan dan berapa lama lagi selesai.
- Sumber daya yang digunakan proses.
- Sumber daya proses yang diperlukan untuk menyelesaikan task.
- Berapa proses yang perlu diterminasi.
- Apakah proses interaktif atau batch.

#### SOAL IATIHAN:

1. Jelaskan tentang konsep *deadlock* !
2. Bagaimana cara mencegah terjadinya *deadlock* ?
3. Bagaimana cara mengatasi jika terjadi *deadlock* ?

## **Bab 14**

### **Konkurensi dan Keamanan**

#### **A. Konkurensi**

Proses-Proses yang disebut konkurensi apabila proses-proses (lebih dari satu proses) berada pada saat yang sama. Karena proses tersebut bisa saja tidak saling bergantung tetapi saling berinteraksi.

#### **Permasalahan konkurensi:**

- 1) Lost update/kehilangan modifikasi.  
Masalah operasi update yang sukses dari seorang pengguna kemudian ditimpali oleh operasi update dari pengguna lain
- 2) Uncommitted dependency/ketergantungan pada saat belum commit  
Permasalahan timbul pada saat transaksi dibiarkan melihat hasil dari transaksi lain yang belum commit.
- 3) Inconsistent analysis/analisis yang tidak konsisten  
Permasalahan timbul apabila suatu transaksi membaca beberapa nilai tetapi transaksi berikutnya memodifikasi salah satu nilai.

#### **B. Keamanan**

**Keamanan sistem komputer** adalah untuk menjamin sumber daya tidak digunakan atau dimodifikasi orang tak terotorisasi.

#### **Keamanan dibagi menjadi 3 yaitu:**

- Keamanan Eksternal (*External Security*) berkaitan dengan pengamanan fasilitas komputer dari penyusup (*Hacker*) dan bencana seperti kebakaran dan banjir.
- Keamanan Interface (*User Interface Security*) berkaitan dengan identifikasi pemakai sebelum pemakai diijinkan mengakses program data yang disimpan.
- Keamanan Internal berkaitan dengan pengamanan beragam kendali yang dibangun pada perangkat keras dan sistem operasi yang menjamin operasi yang handal dan tak terkorupsi untuk menjaga integritas program dan data.

Masalah – Masalah Keamanan, terdapat dua masalah penting yaitu :

- Kehilangan data (*data loss*) dapat disebabkan karena:
  1. Bencana
    - Kebakaran
    - Banjir
    - Gempa Bumi
    - Perang
    - Kerusakan
    - Binatang
  2. Kesalahan perangkat keras dan perangkat lunak

- Tidak berfungsinya pemroses
- Disk atau tape yang tidak terbaca
- Kesalahan telekomunikasi
- Kesalahan program (*bugs*)
- 3. kesalahan atau kelalaian manusia
  - kesalahan pemasukan data
  - memasang tape atau disk yang salah,
  - Eksekusi program yang salah
  - Kehilangan disk atau tape
- Penyusup (*Intruder/Hacker*)
  - penyusup pasif, yaitu yang membaca data yang tak diotorisasi
  - penyusup aktif, yaitu yang mengubah data yang tak diotorisasi

Berdasarkan fungsi ini, ancaman terhadap sistem komputer dapat dikategorikan menjadi empat ancaman, yaitu :

### 1. Interupsi (*interruption*)

Sumber daya sistem komputer dihancurkan atau menjadi tak tersedia atau tak berguna. Interupsi merupakan ancaman terhadap ketersediaan.

**Contoh** : penghancuran bagian perangkat keras, seperti harddisk, pemotongan kabel komunikasi.

**Interupsi adalah** pusat sistem operasi, yang menyediakan cara efisien bagi sistem operasi untuk berinteraksi dan bereaksi terhadap lingkungannya. Pemrograman berbasis interupsi secara langsung didukung hampir seluruh CPU modern. interupsi menyediakan cara otomatis menyimpan isi register local dan menjalankan kode khusus sebagai respon terhadap sebuah kejadian. Bahkan komputer yang paling dasar sekalipun mendukung interupsi hardware dan membolehkan pemrogram untuk menentukan kode yang akan di jalankan ketika terjadi sebuah kejadian.

Ketika sebuah interupsi diterima, hardware komputer secara otomatis menunda program apapun yang sedang dijalankan, menyimpan statusnya dan menjalankan kode komputer yang berhubungan dengan interupsi yang terjadi sebelumnya. Dalam sistem operasi modern, interupsi ditangani oleh kernel sistem operasi. Interupsi bisa datang dari hardware komputer atau program yang sedang jalan.

Ada beberapa tahapan dalam penanganan interupsi:

1. *Controller* mengirimkan sinyal interupsi melalui *interrupt-request-line*
2. Sinyal dideteksi oleh prosesor
3. Prosesor akan terlebih dahulu menyimpan informasi tentang keadaan *state*-nya (informasi tentang proses yang sedang dikerjakan)
4. Prosesor mengidentifikasi penyebab interupsi dan mengakses tabel vektor interupsi untuk menentukan *interrupt handler*
5. Transfer kontrol ke *interrupt handler*

6. Setelah interupsi berhasil diatasi, prosesor akan kembali ke keadaan seperti sebelum terjadinya interupsi dan melanjutkan pekerjaan yang tadi sempat tertunda.

### **2. Intersepsi (interception)**

Pihak tak diotorisasi dapat mengakses sumber daya. Interupsi merupakan ancaman terhadap kerahasiaan. Pihak tak diotorisasi dapat berupa orang atau program komputer.

**Contoh** : penyadapan untuk mengambil data rahasia, mengetahui file tanpa diotorisasi.

### **3. Modifikasi (modification)**

Pihak tak diotorisasi tidak hanya mengakses tapi juga merusak sumber daya. Modifikasi merupakan ancaman terhadap integritas.

**Contoh** : mengubah nilai-nilai file data, mengubah program sehingga bertindak secara berbeda, memodifikasi pesan-pesan yang ditransmisikan pada jaringan.

### **4. Fabrikasi (fabrication)**

Pihak tak diotorisasi menyisipkan/memasukkan objek-objek palsu ke sistem. Fabrikasi merupakan ancaman terhadap integritas.

**Contoh** : memasukkan pesan-pesan palsu ke jaringan, penambahan record ke file.

Ada empat tingkatan di mana suatu sistem harus dilindungi:

#### 1) Fisik

Cara termudah untuk mencuri data adalah dengan mengantongi kaset cadangan. Juga, akses ke konsol root akan sering memberi pengguna hak istimewa, seperti me-reboot sistem sebagai root dari media yang dapat dilepas. Bahkan akses umum ke terminal di ruang komputer menawarkan beberapa peluang bagi penyerang, meskipun lingkungan jaringan modern berkecepatan tinggi saat ini memberikan semakin banyak peluang untuk serangan jarak jauh.

#### 2) Manusia

Ada beberapa kekhawatiran bahwa manusia yang diizinkan mengakses sistem dapat dipercaya, dan bahwa mereka tidak dapat dipaksa untuk melanggar keamanan. Namun semakin banyak serangan hari ini dilakukan melalui rekayasa sosial, yang pada dasarnya berarti membodohi orang yang dapat dipercaya untuk secara tidak sengaja melanggar keamanan.

Phishing melibatkan pengiriman email atau situs web yang terlihat tidak bersalah yang dirancang untuk menipu orang agar mengungkapkan informasi rahasia. Misalnya, email spam yang berpura-pura berasal dari e-Bay, PayPal, atau salah satu dari sejumlah bank atau perusahaan kartu kredit.

Dumpster Diving melibatkan pencarian tempat sampah atau lokasi lain untuk kata sandi yang ditulis. Catatan: Kata sandi yang terlalu sulit untuk diingat, atau yang harus sering diubah lebih cenderung ditulis di suatu tempat dekat dengan stasiun pengguna.

Password Cracking melibatkan meramalkan kata sandi pengguna, baik dengan menonton mereka mengetik kata sandi mereka, mengetahui sesuatu tentang mereka seperti nama hewan peliharaan mereka, atau hanya mencoba semua kata dalam kamus umum. Catatan: kata sandi "Bagus" harus melibatkan jumlah

karakter minimum, termasuk karakter non-abjad, dan tidak muncul dalam kamus apa pun (dalam bahasa apa pun), dan harus sering diubah. Perhatikan juga bahwa etiket yang pantas untuk berpaling dari keyboard saat orang lain memasukkan kata sandi mereka.

3) Sistem Operasi

OS harus melindungi dirinya dari pelanggaran keamanan, seperti proses denial of service, pelanggaran akses memori, pelanggaran stack overflow, peluncuran program dengan hak istimewa yang berlebihan, dan banyak lainnya.

4) Jaringan

Saat komunikasi jaringan menjadi semakin penting dan menyebar di lingkungan komputasi modern, menjadi semakin penting untuk melindungi area sistem ini. Keduanya melindungi jaringan itu sendiri dari serangan, dan melindungi sistem lokal dari serangan yang masuk melalui jaringan.

**SOAL LATIHAN:**

1. Jelaskan yang dimaksud tentang kongkurensi !
2. Jelaskan tentang konsep keamanan sistem !
3. Sebutkan masalah-masalah dalam keamanan sistem !

## Daftar Pustaka

- Hariyanto, B. 2005. *Sistem Operasi: Edisi Kedua*. Bandung: Informatika.
- Prayitno, W dan Wiryana, I.M. 2002. *Be LINUXer with Mandrake 8.0*. Yogyakarta: Penerbit ANDI.
- Sofyan, Ahmad. 2006. *Membuat Distro Linux Sendiri*. Jakarta: Dian Rakyat.
- Silberschatz, A et al. 2009. *Operating System Concepts: 8<sup>th</sup> Edition*. John Wiley and Sons, Inc.
- Arpaci-Dusseau, R.H and Arpaci-Dusseau, A.C. 2015. *Operating Systems: Three Easy Pieces*. CreateSpace Independent Publishing Platform
- Tanenbaum, A.S. 2014. *Modern Operating Systems (3<sup>rd</sup> Edition)*. Prentice Hall.
- Kusnadi, et al. 2011. *Sistem Operasi*. Penerbit ANDI
- Kusumadewi, S. 2002. *Sistem Operasi (Edisi 2)*. Graha Ilmu.