



# ARSITEKTUR DAN ORGANISASI KOMPUTER

**Prio Handoko, S.Kom., M.T.I.**

Fakultas Teknologi dan Desain | Program Studi Informatika

## **PART 3: THE CENTRAL PROCESSING UNIT**

### **CHAPTER 8: OPERATING SYSTEM SUPPORT**

## CHAPTER 8: OPERATING SYSTEM SUPPORT

### Kompetensi Dasar

Mahasiswa memiliki pengetahuan mengenai dukungan sistem operasi terhadap sistem komputer

### Agenda

- Operating System Overview
- Scheduling
- Memory Management OS

## 8.1 Operating System Overview

- Perangkat lunak yang mengendalikan pengolahan program aplikasi
- Antarmuka antara pengguna dengan perangkat keras

### Program

- **I/O-bound program:**  
program akan cenderung menggunakan sebagian besar waktunya untuk melakukan operasi I/O.

## 8.1 Operating System Overview

- **Processor-bound program:**  
program akan cenderung menghabiskan sebagian besar waktunya untuk pengolahan CPU.

### Sasaran OS

1. Kenyamanan
2. Efisiensi

## 8.1 Operating System Overview

### Fungsi OS

#### 1. Extended Machine

- Menyembunyikan kompleksitas pemrograman hardware dari programmer atau user
- Menyajikan fasilitas yang lebih mudah untuk menggunakan hardware

#### 2. Resource Manager

- Mengatur dan mengefisienkan penggunaan sumber daya komputer
- Masing-masing program mendapatkan waktu dan space terhadap sumber daya

## 8.1 Operating System Overview

### Layanan OS

- **Pembuatan Program:** Biasanya dalam bentuk **utility** (editor, **debugger** untuk membantu pemrogram)
- **Eksekusi Program:** Menaruh instruksi dan data ke main memory
- **Akses ke perangkat I/O:** Sistem operasi mengatasi masalah detail operasional perangkat I/O.

## 8.1 Operating System Overview

- **Controlled access to files:** Sistem multi user, o/s menyediakan mekanisme perlindungan untuk akses ke file
- **System access:** Perlindungan sumber daya dan data
- **Error detection and response**
- **Accounting:** OS mengumpulkan statistik pemakaian untuk berbagai sumber daya, yang berguna untuk peningkatan kinerja

## 8.1 Operating System Overview

### Jenis OS

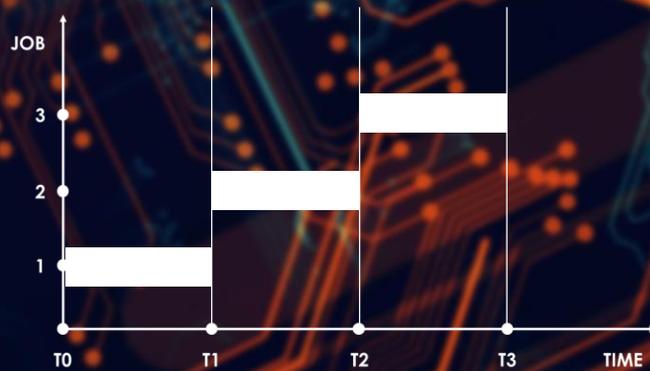
- **Interactive:** User berinteraksi secara langsung dengan komputer
- **Batch:** Program user ditampung bersama-sama dengan user yang lainnya, dan kemudian disampaikan ke operator komputer.
- **Single program (Uni-programming):** Mengerjakan satu program setiap saat

## 8.1 Operating System Overview

- **Multi-programming (Multi-tasking):**
  1. Menjaga agar prosesor sibuk setiap saat, dgn mengerjakan program lebih dari satu setiap saat.
  2. Beberapa program di-load ke main memory dan prosesor beralih dari satu program ke program yang lainnya.

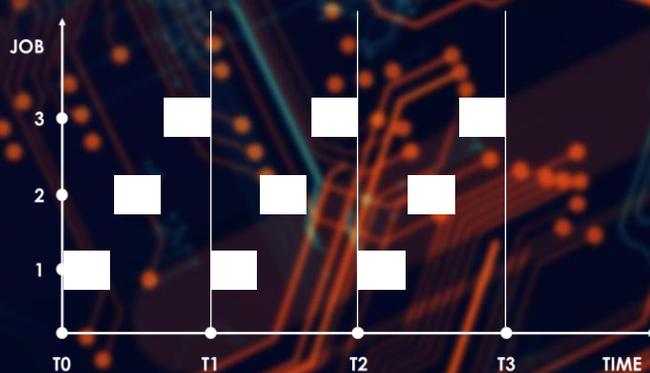
## 8.1 Operating System Overview

### Mono Programming



## 8.1 Operating System Overview

### Multi Programming



## 8.1 Operating System Overview

### Paralel Programming

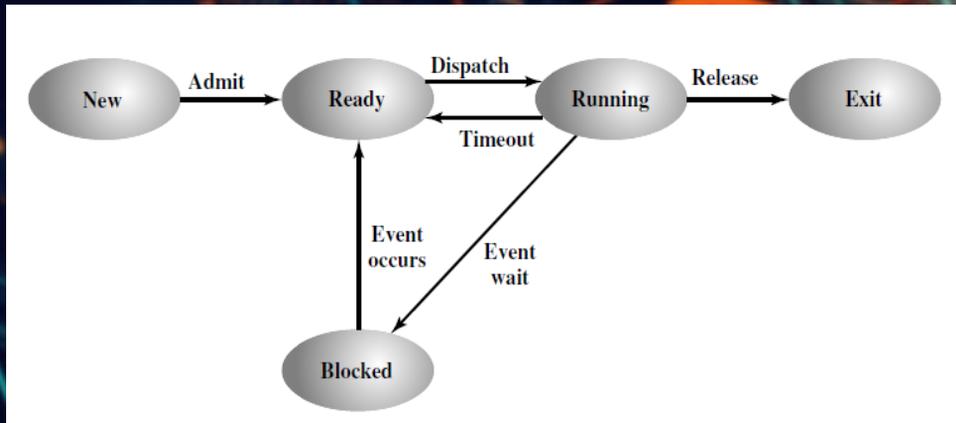


## 8.2 Scheduling

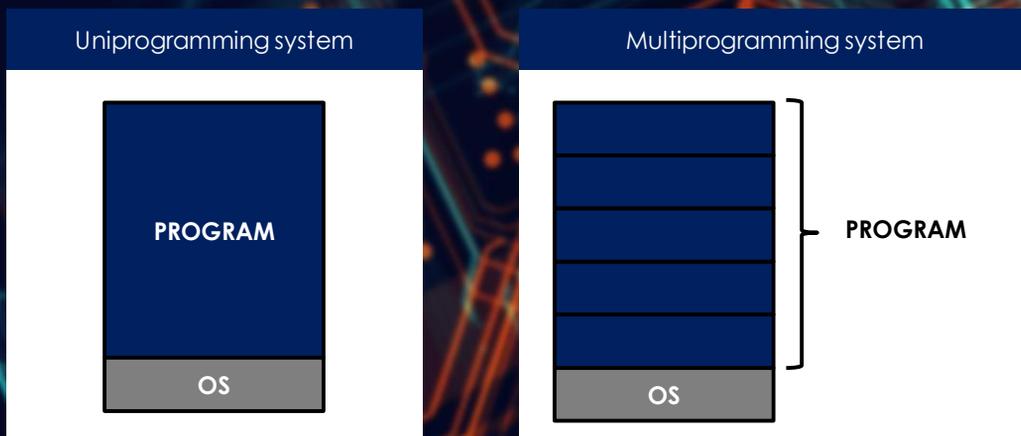
1. **Jangka Pendek (*short-term scheduler*):** menjadwalkan alokasi memori untuk job-job ready
2. **Jangka Menengah (*medium-term scheduler*):** mengendalikan transisi *suspended* → *ready* job-job yang di-*swapping*
3. **Jangka Panjang (*long-term scheduler*):** mengelola antrian batch dan memilih batch berikutnya yang akan diproses/dieksekusi

## 8.2 Scheduling

### Five-State Process Model



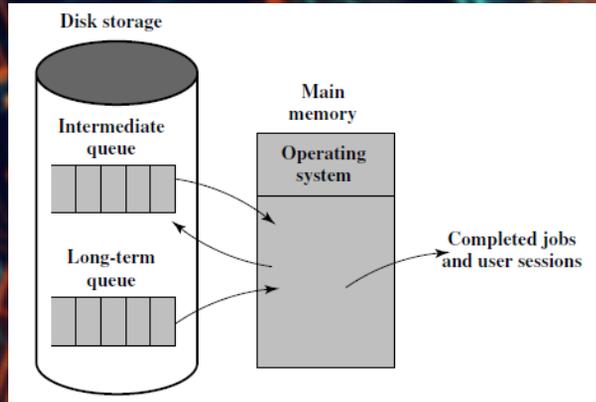
## 8.3 Memory Management OS



## 8.3 Memory Management OS

### Swapping

Adanya pertukaran data antara virtual memory dan physical memory untuk memaksimalkan ruang dalam memori



## 8.3 Memory Management OS

### Partitioning

Membagi-bagi memori menjadi beberapa bagian (partisi), baik berukuran sama ataupun berbeda ukuran, statik atau dinamik, untuk dapat memuatkan proses-proses ke dalam memori sehingga dapat digunakan dalam pengolahan CPU.

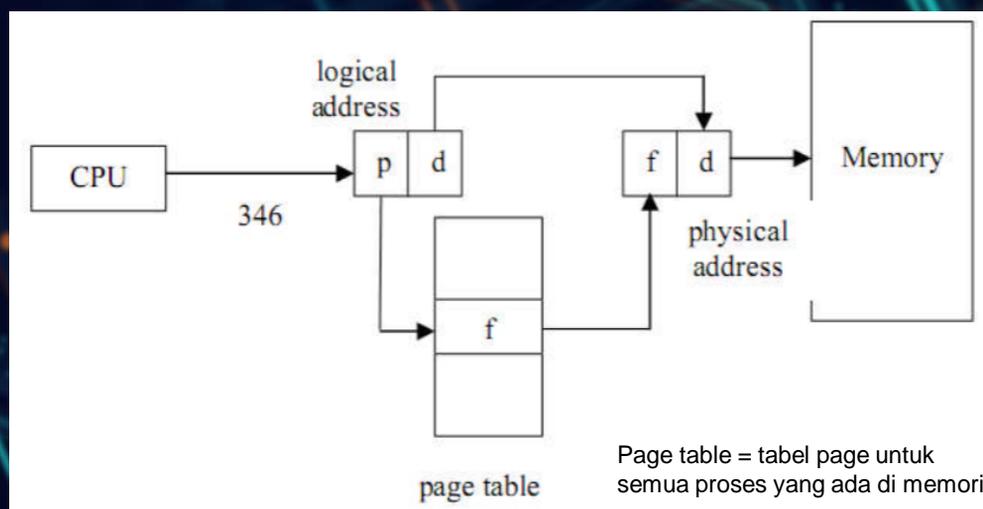
## 8.3 Memory Management

### Virtual Memory

Sebuah teknik manajemen memori yang menyebabkan seolah-olah sistem komputer memiliki kapasitas memori yang lebih besar dari memory fisiknya.

Menggunakan sebagian kecil **hard disk** dan akan menyalinkan datanya ke RAM jika diperlukan.

## 8.3 Memory Management



## 8.3 Memory Management OS

### Paging

- Salah satu cara untuk mengatasi fragmentasi eksternal (proses lebih besar daripada partisi yang tersedia)
- **Paging:** memori fisik dibagi menjadi blok-blok dengan ukuran tertentu yang disebut dengan frame/page frame, sedangkan memori logika/maya dibagi menjadi blok-blok yang disebut page.

## 8.3 Memory Management

### Paging

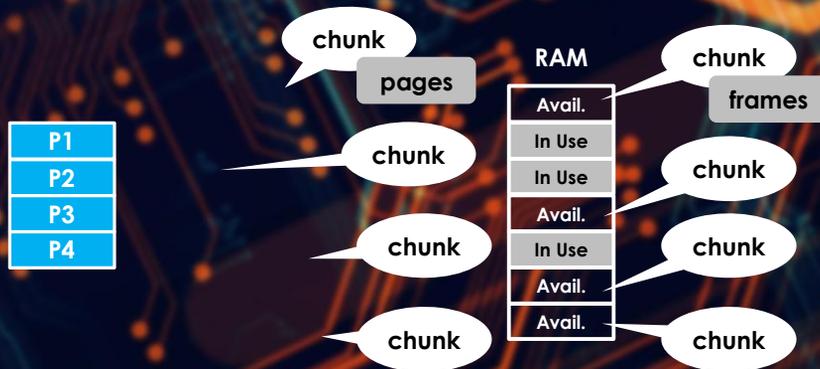
P1
P2
P3
P4

#### RAM

Avail.
In Use
In Use
Avail.
In Use
Avail.
Avail.

## 8.3 Memory Management

### Paging



## 8.3 Memory Management

### Paging



## 8.3 Memory Management OS

### Alamat Logika dan Fisik

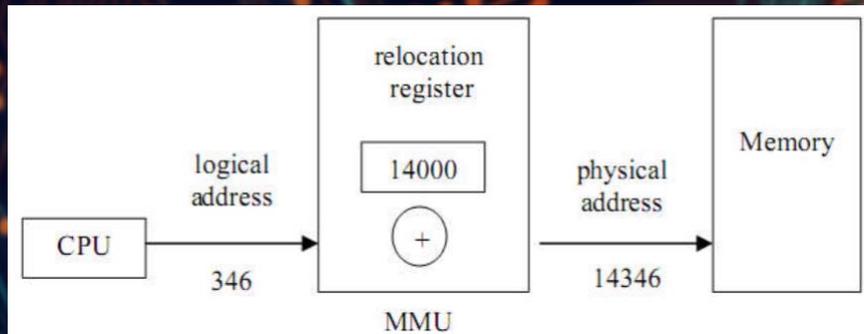
- Alamat logika (logical address/virtual address) = alamat yang dihasilkan oleh CPU disebut alamat logika/alamat maya
- Alamat fisik (physical address) = alamat program yang sesungguhnya pada memori
- Pada saat eksekusi setiap alamat logik harus dipetakan ke alamat fisik sehingga alamat logik berbeda dengan alamat fisik.

## 8.3 Memory Management OS

- Pemetaan dari alamat logik ke alamat fisik dilakukan dengan menggunakan perangkat keras yang disebut Memory Management Unit(MMU).
- MMU memiliki register relokasi yang berisi alamat awal proses.
- Nilai alamat awal ini akan ditambahkan ke setiap alamat logik pada proses untuk menciptakan alamat fisik.

## 8.3 Memory Management OS

- Contoh, bila alamat awal adalah 14000, maka pengguna yang ingin mengakses lokasi 0, secara otomatis akan dipetakan kealamat 14000. Akses ke lokasi 346 akan dipetakan ke lokasi 14346.



## 8.4 Sistem Paging

Pada sistem paging alamat logika terdiri dari 2 bagian yaitu :

- nomor page  $p$  : digunakan sebagai indeks untuk page table yang berisi alamat awal untuk setiap page pada memori
- offset page  $d$  : offset page ditambahkan pada alamat awal untuk menghasilkan alamat memori sebenarnya.

## 8.4 Sistem Paging

Pada sistem paging alamat logika terdiri dari 2 bagian yaitu :

- nomor page  $p$  : digunakan sebagai indeks untuk page table yang berisi alamat awal untuk setiap page pada memori
- offset page  $d$  : offset page ditambahkan pada alamat awal untuk menghasilkan alamat memori sebenarnya.
- alamat offset diperoleh dari nilai absolut alamat logika yang ditentukan dikurangi dengan alamat logika awal dari page yang diketahui. **Contoh:** jika alamat logika 380 dari alamat logika awal 320, berarti alamat offsetnya adalah absolut  $(380 - 320) = 60$ .

## 8.4 Sistem Paging

### Contoh.

Diketahui:

1. Ukuran page = 4 byte dan ukuran memori = 32 byte (8 page).
2. Alamat logik 0 memiliki nomor page 0 dan offset 0. Ketika dihubungkan dengan page table, maka diketahui bahwa page 0 terletak pada frame 5.

## 8.4 Sistem Paging

Tentukanlah alamat fisik dari:

1. alamat logik 0 (page 0 dan offset 0)
2. alamat logik 3 (page 0 dan offset 3)
3. alamat logik 4 (page 1 dan offset 0)
4. alamat logik 20 (page 4 dan offset 13)
5. alamat logik 15 (page 6 dan offset 100)
6. alamat logik 107 (page 7 dan offset 23)
7. alamat logik 67 (page 8 dan offset 27)

## 8.4 Sistem Paging

### Contoh.

Diasumsikan suatu sistem komputer memiliki memori utama dengan kapasitas 16 MB. Diketahui ukuran page sebesar 64 byte, maka:

1. Berapa jumlah frame yang tersedia
2. Jika suatu program TEST berukuran 950 byte, berapakah banyanya page yang dibutuhkan?
3. Apabila diketahui page table sebagai berikut:

## 8.4 Sistem Paging

Nomor page	Frame
0	8
1	2
2	10
3	22
4	12
5	1
:	:
:	:

Dengan asumsi bahwa program membutuhkan page secara berurutan dari 0 sampai n, dimanakah letak alamat fisik dari alamat logika 50, 121, dan 380

## 8.3 Memory Management

### Latihan 1.

Diketahui isi **page tabel** seperti di samping ini, semua angka dalam desimal. Jika ukuran **page** adalah 64 byte, hitung **physical address** dari **virtual address** berikut:

1. 10534
2. 22288
3. 14920

Virtual Page Number (VPN)	Valid Bit	Page Frame Number (PFN)
0	1	4
1	1	7
2	0	11
3	1	2
4	0	9
5	1	0

## 8.3 Memory Management

### Latihan 2.

Diketahui isi **page tabel** seperti di samping ini, semua angka dalam desimal. Jika ukuran **page** adalah 256 byte, hitung **physical address** (jika ada) dari **virtual address** berikut:

1. 30834
2. 62226
3. 51010

Virtual Page Number (VPN)	Valid Bit	Page Frame Number (PFN)
0	0	12
1	1	7
2	1	5
3	0	20
4	1	0
5	0	13

## PART 3: THE CENTRAL PROCESSING UNIT

### CHAPTER 8: OPERATING SYSTEM SUPPORT



UNTIL THEN...