



## HARDWARE

### PROCESSOR

- ALU
- REGISTER
- CU

### MEMORI

- MEMORI REGISTER
- MEMORI CASE
- MEMORI KERJA
- DISK MAGNETIK
- DIS OPTIK
- MAGNETIC TAPE

### DEVICE I/O

- K.MEKANIK
- K.ELEKTRONIK

### INTERKONEKSI ANTAR KOMPONEN(BUS)

- BUS ALAMAT
- BUS KONTROL
- BUS DATA

## SOFTWARE

### FIRMWARE

### SISTEM OPERASI

### APLIKASI

## SIKLUS INSTRUKSI

### MENGAMBIL INSTRUKSI

### MENJALANAN INSTRUKSI

# Review Pert 2



## **Pertemuan 3**

# **Manajemen Memori, Manajemen Masukan-Keluaran Dan Manajemen File**



## I. MANAJEMEN MEMORI

Terdapat 2 (dua) manajemen memori :

### a. manajemen memori statis

Dengan pemartisian statis, jumlah, lokasi dan ukuran proses dimemori **tidak beragam** sepanjang waktu secara tetap.

### b. manajemen memori dinamis

Dengan pemartisian dinamis , jumlah, lokasi dan ukuran proses dimemori dapat **beragam** sepanjang waktu secara dinamis.



## 1. Manajemen Memori Berdasarkan Alokasi memori

Terdapat 2 (dua) cara **menempatkan informasi** ke dalam memori kerja, yaitu:

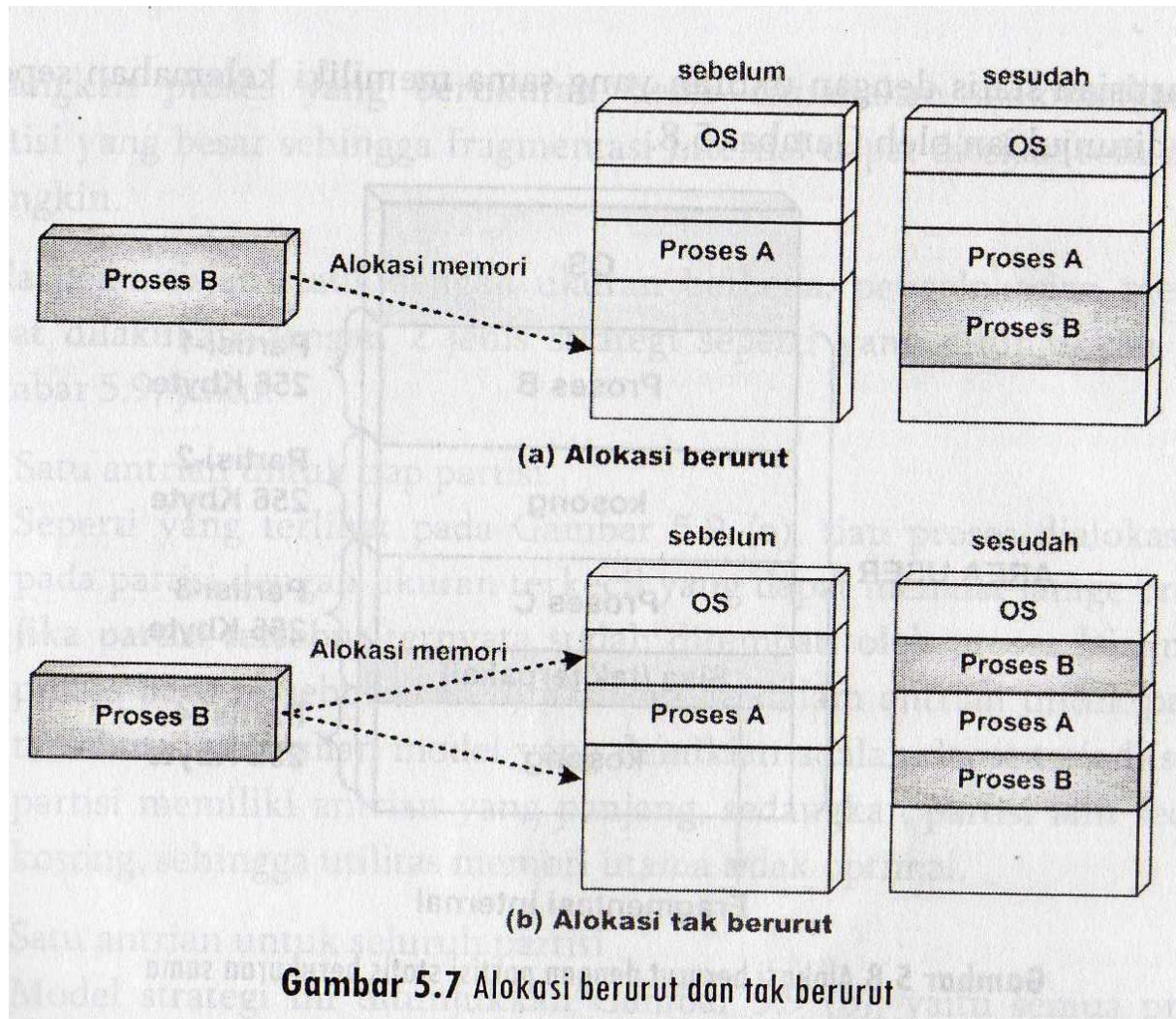
### a. Alokasi Memori Berurutan (*Contiguous Allocation*)

- ❑ Pada alokasi memori berurutan, setiap proses menempati satu blok tunggal lokasi memori yang berurutan.
- ❑ Kelebihan : sederhana, tidak ada rongga memory bersebaran, proses berurutan dapat dieksekusi secara cepat.
- ❑ Kekurangan : memori boros, tidak dapat disisip apabila tidak ada satu blok memori yang mencukupi.



## b. Alokasi Memori Tak Berurutan (*Non Contiguous Allocation*)

- ❑ Program/proses ditempatkan pada beberapa segmen berserakan, tidak perlu saling berdekatan atau berurutan. Biasanya digunakan untuk lokasi memori maya sebagai lokasi page-page.
  
- ❑ Kelebihan : sistem dapat memanfaatkan memori utama secara lebih efisien, dan sistem operasi masih dapat menyisip proses bila jumlah lubang-lubang memori cukup untuk memuat proses yang akan dieksekusi.
  
- ❑ Kekurangan : memerlukan pengendalian yang lebih rumit dan memori jadi banyak yang berserakan tidak terpakai.



**Gambar 5.7** Alokasi berurut dan tak berurut



Alokasi pada sistem multiprogramming					
Alokasi berurut (contiguous)			Alokasi tidak berurut (non-contiguous)		
Partisi statis	Partisi dinamis	Sistem buddy	Paging	Segmentasi	kombinasi



Terdapat 2 (dua) macam pemilahan, yaitu:

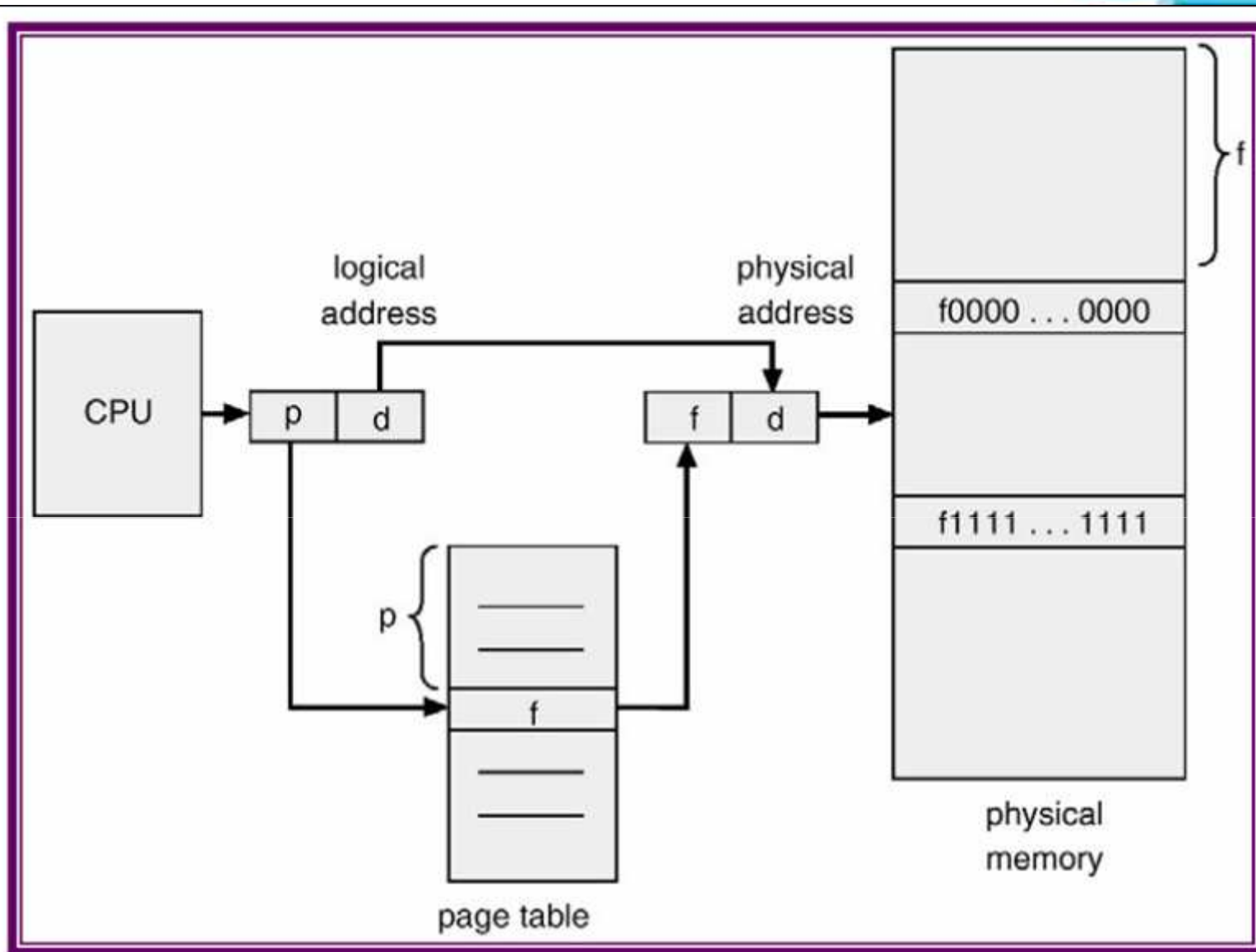
**a. Berpilah suku (*paging*)**

Informasi atau pekerjaan di dalam memori dukung dipilah ke dalam sejumlah suku (*page*), dan memori kerja dipilah ke dalam sejumlah rangka (*frame*)

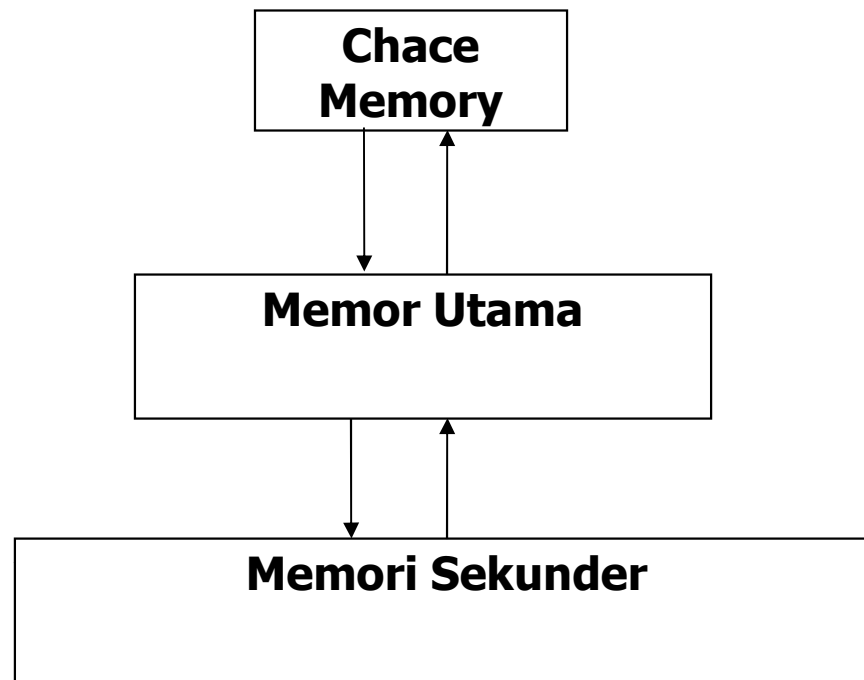
**b. Berpilah segmen (*segmentasi*)**

Pilahan yang ukuran segmen disesuaikan dengan isi segmen  
Salah satu macam pemilahan gabungan suku dan segmen adalah pemilahan suku bersegmen, dimana suku dikelompokkan ke dalam sejumlah segmen.





Gambar 7-11 : Arsitektur sistem paging



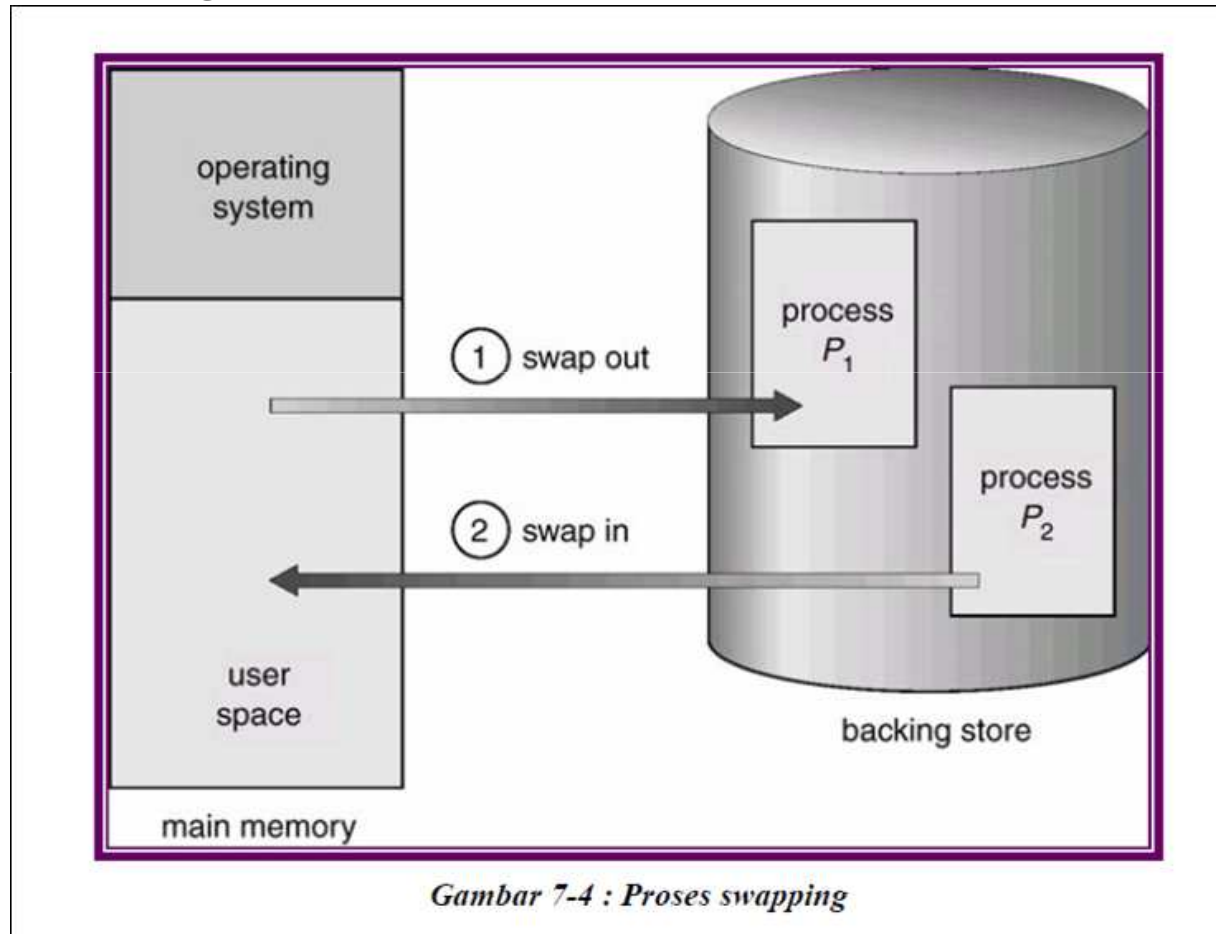
**Gambar 3.1. Hubungan Chace memory, Memori Utama dan Memori Sekunder**

Chace memory memiliki kecepatan lebih tinggi sebagai memori antara yang mempercepat proses pada memory kerja, juga sebagai transit lalulintas data selama proses dengan sumberdaya lain pada memori utama.



# SWAPPING

Pemindahan proses dari memori utama ke disk dan sebaliknya disebut *swapping*



Gambar 7-4 : Proses swapping



## **2. Manajemen memori berdasarkan keberadaan :**

- a. Dengan Swapping ( dengan kemampuan memindahkan citra/gambaran proses antara memori utama dan disk selama eksekusi)
- b. Tanpa Swapping ( tanpa memindahkan citra/gambaran proses antara memori utama dan disk selama eksekusi)

## **3. Manajemen memori Tanpa Swapping terdiri dari**

- a. Monoprogramming.

Ciri-ciri:

Hanya satu proses pd satu saat

Hanya satu proses menggunakan semua memori

Pemakai memuatkan program ke semua memori

Program mengambil kendali seluruh mesin.

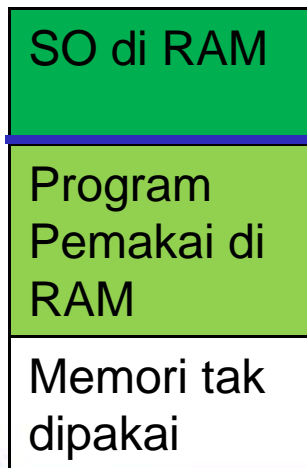


- » **Embedded system** (mikro controller) --> untuk mengendalikan alat agar bersifat intelejen dlm menyediakan satu fungsi spesifik. Dapat di[akai untuk sistem kecil yaitu sistem tempelan (menempel di sistem lain).

Keperluan dapat diprogramkan di mikroprosesor dg memori kecil (1-64kb)

Contoh: intel 8051

Sistem pada mobil mewah, di dalamnya terdapat >50 mikroprosesor yg mengendalikan fungsi spesifik (pengendalian rem, suspensi, kemudi).



- » **Proteksi pada monoprogramming sederhana** - -> Cara memproteksi rutin2 sistem operasi dari penghancuran program pemakai. Diproteksi menggunakan boundary register(register batas)

————— =Register batas



b. **multiprogramming** dengan pemartisian statis.

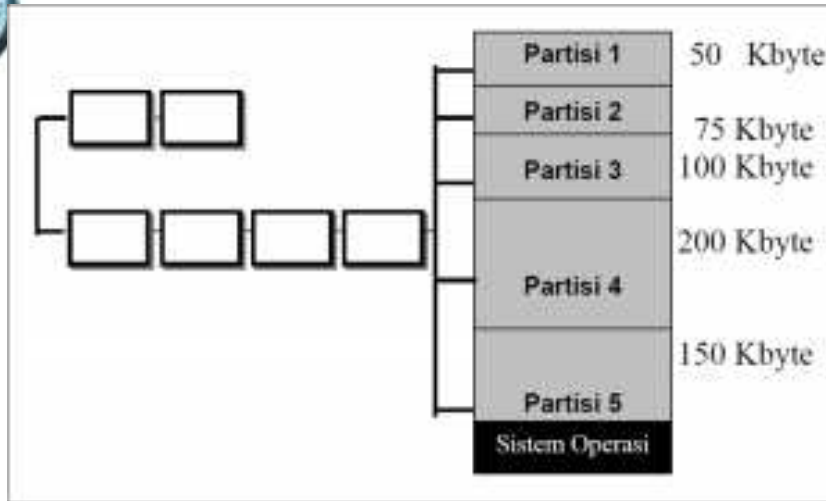
Multiprogramming digunakan untuk: mempermudah pemrogram, memberi layanan interaktif ke beberapa org scr simultan, efisiensi sumber daya, eksekusi lebh murah jika proses besar dibagi menjadi proses kecil.

>> **Strategi penempatan program ke partisi** (Partisi ukuran sama dan berbeda.

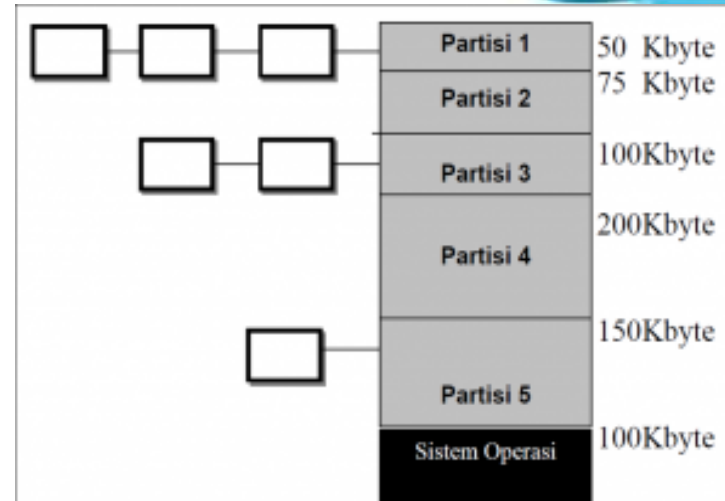
>> **Relokasi** --> Adalah masalah penempatan proses sesuai alamat fisik sehubungan alamat partisi memori dimana proses ditempatkan. Proses dapat ditempatkan pada partisi-partisi berbeda menurut keadaan sistem saat itu. Pengalamatan fisik secara absolut untuk proses tidak dapat dilakukan.

>> **Proteksi pada multi programming:** bekerja dg. Banyak proses di satu sistem scr bersamaan dikawatirkan proses menggunakan atau memodifikasi daerah yg dikuasai proses lain.

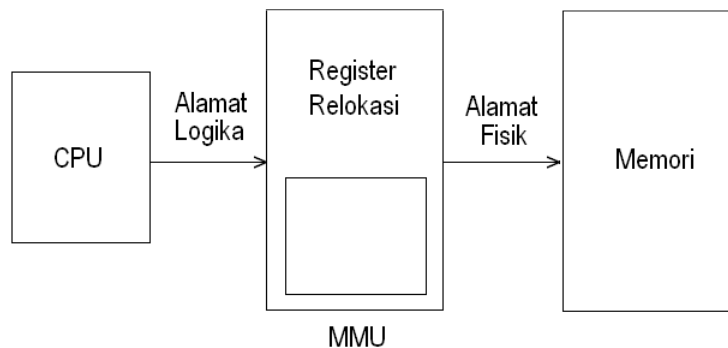
>> **Fragmentasi pada pemartisian statis** : penyiaan.  
Pemborosan memori akan terjai pada setiap organisasi penyimpanan



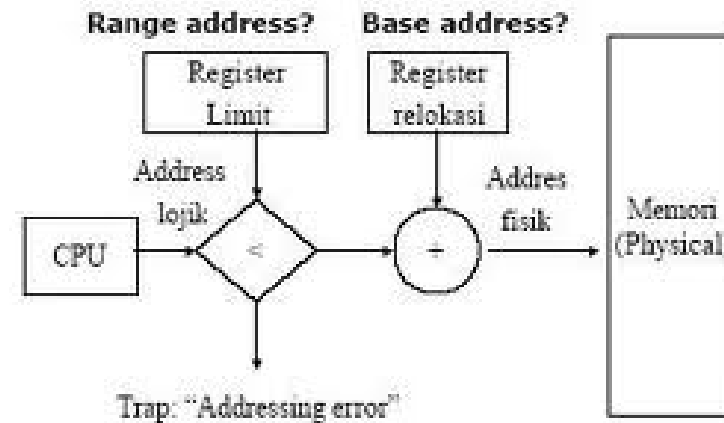
Pemartisian tetap dg satu antrian



Pemartisian tetap dg banyak antrian



Relokasi



Relokasi & proteksi Multiprogramming



#### 4. Manajemen memori pada multi programming

a. **dengan swapping**, multiprogramming sistem timesharing dapat ditingkatkan kinerjanya dg memindah proses-proses blocek ke disk dan hanya memasukkan proses ready ke memori utama.

b. **dengan pemartisian Dinamis**

- » adanya lubang-lubang kecil dimemori
- » proses tumbuh berkembang

Memori untuk SO	Memori untuk SO
Proses 0	Proses 0
Proses 1	Bebas
Proses 2	Proses 2
Proses 3	Proses 3
Proses 4	Proses 4





## 5. Pencatatan Pemakaian Memori

- a. **Pencatatan memakai peta bit** --> 0 untuk memori bebas, 1 untuk memori terpakai proses

Masalah pada peta bit adalah penetapan mengenai ukuran unit alokasi memori, yaitu :

- ☑ Unit lokasi memori berukuran kecil berarti membesarkan ukuran peta bit.
- ☑ Unit alokasi memori n berukuran besar berarti peta bit kecil tapi memori banyak disediakan pada unit terakhir jika ukuran proses bukan kelipatan unit alokasi

- b. **Pencatatan memakai penghubung/ senarai berkait**--> Sistem operasi mengelola senarai berkait (linked list) untuk segmen-segmen memori yang telah dialokasikan dan bebas. Segmen memori menyatakan memori untuk proses atau memori yang bebas (lubang). Senarai segmen diurutkan sesuai alamat blok.



## 6. Penggunaan memori

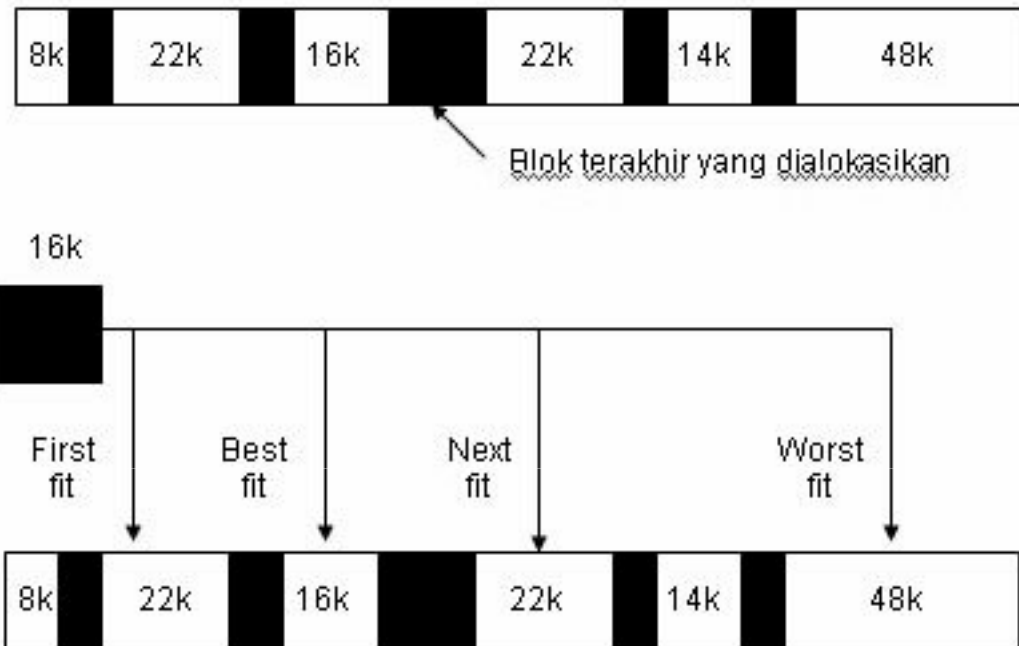
- ❑ Pencocokan ukuran informasi ke penggalan memori kerja disebut sebagai **fit**
- ❑ Bagian dari memori kerja yang tidak terpakai dan letaknya tersebar di banyak wilayah memori kerja disebut sebagai **fragmen**
- ❑ Peristiwa terjadinya fragmen disebut **fragmentasi**



## 7. Pencocokan (fit) dan fragmentasi

Beberapa jenis strategi pencocokan antara lain:

- ❑ Cocok pertama (**first fit**)  
Pencocokan terjadi menurut antrian informasi
- ❑ Cocok pertama berdaur (**cyclical first fit**)  
Pencocokan tidak harus dimulai dari urutan penggalan memori yang pertama, tetapi dapat dilakukan setelah terjadi pencocokan sebelumnya.
- ❑ Cocok terbaik (**best fit**)  
Pencocokan dilakukan sesuai dengan penggalan memori yang ukurannya pas.
- ❑ Cocok terburuk (**Worst fit**)  
Informasi akan menempati penggalan yang ukurannya terbesar.



Gambar 3.2 algoritma best, fist, next, worst fit



## 8. Fragmentasi

Menurut prosesnya terdapat dua macam fragmentasi :

### a. **Fragmentasi internal**

Kelebihan memori pada penggalan memori ketika penggalan memori itu menerima penggalan informasi yang berukuran kurang dari ukuran penggalan memori

### b. **Fragmentasi Ekternal**

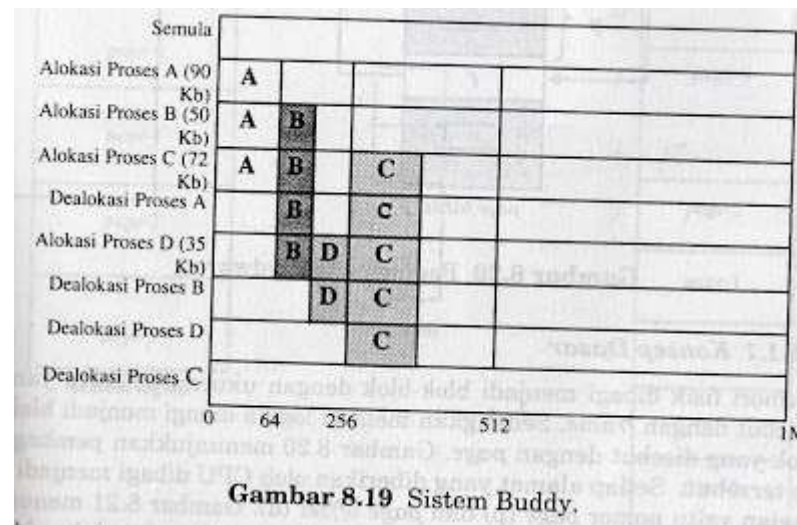
Penggalan memori bebas yang ukurannya terlalu kecil untuk dapat menampung penggalan informasi yang akan dimuat ke penggalan memori itu.

Contoh : Proses



## 9. Sistem Buddy

Sistem buddy adalah algoritma pengelolaan memori yang memanfaatkan kelebihan penggunaan bilangan biner dalam pengalamatan memori. Karakteristik bilangan biner digunakan untuk mempercepat Penggabungan lubang-lubang berdekatan ketika proses Terakhir atau dikeluarkan. Mekanisme pengelolaan sistem buddy tersebut memiliki keunggulan dan kelemahan.





## Keunggulan Sistem Buddy

1. Sistem buddy mempunyai keunggulan dibanding algoritma-algoritma yang mengurutkan blok-blok berdasarkan ukuran. Ketika blok berukuran  $2k$  dibebaskan, maka manajer memori hanya mencari pada senarai lubang  $2k$  untuk memeriksa apakah dapat dilakukan penggabungan. Pada algoritma algoritma lain yang memungkinkan blok-blok memori dipecah dalam sembarang ukuran, seluruh senarai harus dicari.
2. Dealokasi pada sistem buddy dapat dilakukan dengan cepat.



## Kelemahan Sistem Buddy

1. Utilisasi memori pada sistem buddy sangat tidak efisien.
2. Masalah ini muncul dari kenyataan bahwa semua permintaan dibulatkan ke 2k terdekat yang dapat memuat. Proses berukuran 35 kb harus dialokasikan di 64 kb, terdapat 29 kb yang disiakan. Overhead ini disebut fragmentasi internal karena memori yang disiakan adalah internal terhadap segmen-segmen yang dialokasikan





## 10. Alokasi Ruang Swap pada Disk.

Terdapat dua strategi utama penempatan proses yang dikeluarkan dari memori utama (swapout) ke disk, yaitu :

a. Ruang disk tempat swap dialokasikan begitu diperlukan

-> Ketika proses harus dikeluarkan dari memori utama, ruang disk segera dialokasikan sesuai ukuran proses.

Untuk itu diperlukan algoritma untuk mengelola ruang disk seperti untuk mengelola memori utama. Ketika proses dimasukkan kembali ke memori utama segera ruang disk untuk swap didealokasikan.



## Alokasi ruang swap (Continue)

2. Ruang disk tempat swap dialokasikan lebih dulu  
-> Saat proses diciptakan, ruang swap pada disk dialokasikan. Ketika proses harus dikeluarkan dari memori utama, proses selalu ditempatkan ke ruang yang telah dialokasikan, bukan ke tempat-tempat berbeda setiap kali terjadi swap-out. Ketika proses berakhir, ruang swap pada disk didealokasikan.



# MANAJEMEN PERANGKAT MASUKAN / KELUARAN DAN MANAJEMEN FILE

## II. Manajemen Perangkat Masukan/keluaran

**Fungsi** manajemen perangkat masukan keluaran :

- ❑ Mengirim perintah keperangkat masukan/keluaran agar menyediakan layanan.
- ❑ Menangani interupsi perangkat masuakan/keluaran
- ❑ Menangani kesalahan pada perangkat masukan/keluaran
- ❑ Menyediakan *interface* ke pemakai



## 1. Klasifikasi Perangkat I/O

### a. Klasifikasi Perangkat I/O berdasarkan **sifat aliran datanya**,

- Berorientasi blok** --> Menyimpan, Menerima dan mengirim informasi sebagai blok berukuran tetap 128 – 1024 dan memiliki alamat tersendiri (**disk, tape, cdrom**)
- Karakter** --> Perangkat yang menerima dan mengirim aliran karakter tanpa membentuk suatu struktur blok (**terminal, line printer, interface jaringan**)

### b. Klasifikasi Perangkat I/O **berdasarkan sasaran komunikasi**,

- Terbaca oleh manusia** --> Alat yang digunakan untuk komunikasi dgn manusia
- Terbaca oleh mesin** --> Alat yg digunakan untuk berkomunikasi dgn perangkat elektronik
- Untuk komunikasi** --> alat yg digunakan untuk komunikasi dengan perangkat jarak jauh



# Pemrograman Perangkat I/O

## 2. Teknik Pemrograman Perangkat I/O

### a. **Programmed I/O atau Polling System**

Ketika Alat I/O menangani Permintaan, alat menset bit status di register status perangkat, alat tidak memberitahukan ke pemroses saat tugas telah selesai sehingga prosesor harus selalu memeriksa register tersebut secara periodik dan melakukan tindakan berdasarkan status yang dibaca



b. Masukan/keluaran dikendalikan interupsi (interrupt I/O)

**I/O dikendalikan interupsi.**

Teknik I/O dituntun interupsi mempunyai mekanisme kerja sebagai berikut :

- Pemroses memberi instruksi ke perangkat I/O kemudian melanjutkan melakukan pekerjaan lainnya.
- Perangkat I/O akan menginterupsi meminta layanan saat perangkat telah siap bertukar data dengan pemroses.
- Saat menerima interupsi perangkat keras (yang memberitahukan bahwa perangkat siap melakukan transfer), pemroses segera mengeksekusi transfer data



### c. DMA (*Direct Memory Access*)

berfungsi membebaskan pemroses menunggu transfer data yang dilakukan perangkat I/O. Saat pemroses ingin membaca atau menulis data, pemroses memerintahkan DMA controller dengan mengirim informasi berikut :

- Perintah penulisan/pembacaan.
- Alamat perangkat I/O.
- Awal lokasi memori yang ditulis/dibaca.
- Jumlah word (byte) yang ditulis/dibaca.

Setelah mengirim informasi-informasi itu ke DMA controller, pemroses dapat melanjutkan kerja lain. Pemroses mendelegasikan operasi I/O ke DMA.. DMA mentransfer seluruh data yang diminta ke/dari memori secara langsung tanpa melewati pemroses. Ketika transfer data selesai, DMA mengirim sinyal interupsi ke pemroses. Sehingga pemroses hanya dilibatkan pada awal dan akhir transfer data. Operasi transfer antara perangkat dan memori utama dilakukan sepenuhnya oleh DMA lepas dari pemroses dan hanya melakukan interupsi bila operasi telah selesai.



### 3. Evolusi Fungsi

- ❖ Pemroses mengendalikan perangkat masukan/keluaran secara langsung (*Embedded System*)
- ❖ Pemroses dilengkapi pengendali masukan/keluaran (*I/O controller*)
- ❖ Perangkat dilengkapi fasilitas interupsi
- ❖ I/O controller mengendalikan memori secara langsung melalui DMA (Melibatkan Pemroses diawal dan akhir transfer)
- ❖ Pengendali masukan/keluaran menjadi terpisah
- ❖ Pengendali masukan /keluaran mempunyai memori lokal.





#### 4. Prinsip manajemen

- ❑ Efisien (*eficiency*) --> Usb Hub
- ❑ Kecukupan (*generality*) --> Siplisitas

#### 5. Hirarki Manajemen

- ❑ ***Interrupt Handler***--> Interupsi harus disembunyikan agar tidak terlihat rutin berikutnya. Device driver di blocked saat perintah I/O diberikan dan menunggu interupsi. Ketika interupsi terjadi, prosedur penanganan interupsi bekerja agar device driver keluar dari state blocked.
- ❑ ***Device Driver*** --> Semua kode bergantung perangkat ditempatkan di device driver. Tiap device driver menangani satu tipe (kelas) perangkat dan bertugas menerima permintaan abstrak perangkat lunak device independent di atasnya dan melakukan layanan permintaan.



- ❑ ***Perangkat Lunak Device-Independent*** --> Bertujuan membentuk fungsi-fungsi I/O yang berlaku untuk semua perangkat dan memberi antarmuka seragam ke perangkat lunak tingkat pemakai
- ❑ ***Perangkat Lunak Level Pemakai*** --> Kebanyakan perangkat lunak I/O terdapat di sistem operasi. Satu bagian kecil berisi pustaka-pustaka yang dikaitkan pada program pemakai dan berjalan diluar kernel.



## 6. *Buffering*

Meningkatkan efisiensi sistem operasi dan kinerja proses. Terdapat beragam cara *buffering*, antara lain:

- ❑ ***Single buffering***: proses memberiperintah u/ perangkat i/o sistem operasi menyediakan buffer memori utama milik sistem operasi u/ operasi
- ❑ ***Double buffering***: proses dapat melakukan transfer dari/ke satu buffer sementara sistem operasi mengosongkan buffer lain.
- ❑ ***Circular buffering***: Jia double buffering tidak mencukupi dapat digunakan lebih dari dua buffer.



### III. Sistem Manajemen File

Pengertian File --> Abstraksi penyimpanan dan pengambilan informasi di Disk

Sifat file : **Presisten** --> Informasi dalam file tetap ada walaupun catudaya dimatikan

**Size, Sharability** --> digunakan oleh banyak proses secara konkuren

#### 1. Sasaran dan Fungsi Sistem Manajemen File

a. **Sasaran** : memenuhi Kebutuhan Manajemen Data bagi pemakai (*retrive, insert, delete, update*), Menjamin data pada file adalah Valid, Optimasi Kinerja (*Trouhgput* dan waktu tanggap ditingkatkan ), Menyediakan dukungan masukan/keluaran beragam tipe perangkat penyimpanan, Meminimalkan potensi kehilangan data



# Fungsi manajemen File

## b. Fungsi Manajemen File

Beberapa fungsi yang diharapkan dari pengelolaan file adalah:

- Penciptaan, modifikasi dan penghapusan file.
- Mekanisme pemakaian file secara bersama.

Menyediakan beragam tipe pengaksesan terkendali, seperti:

- *Read access* (pengendalian terhadap akses membaca).
  - *Write access* (pengendalian terhadap akses memodifikasi).
  - *Execute access* (pengendalian terhadap akses menjalankan program).
- Kemampuan backup dan recovery untuk mencegah kehilangan karena kecelakaan atau dari upaya penghancuran informasi.



- 2. Arsitektur Pengelolaan File**, terdiri dari
  - a. Sistem Akses, berkaitan dengan bagaimana cara data yang disimpan pada file diakses
  - b. Manajemen File, berkaitan dgn mekanisme operasi pada file (*saving, searching, sharable, security*)
  - c. Manajemen Ruang Penyimpanan, berkaitan dengan alokasi ruang untuk file di storage
  - d. Mekanisme Integritas File, berkaitan dengan jaminan informasi pada file tak terkorupsi
- 3. Konsep Sistem File** --> Abstraksi penyimpanan dan pengambilan informasi di disk. Abstraksi ini membuat pemakai tidak dibebani rincian cara dan letak penyimpanan informasi, serta mekanisme kerja perangkat penyimpan data
- 4. Pandangan File** --> Penamaan, tipe, attribute, perintah manipulasi, operasi



**5. Direktori** --> Direktori merupakan file yang dimiliki sistem untuk mengelola struktur sistem file. File direktori merupakan file berisi informasi-informasi mengenai file-file yang termasuk dalam direktori itu.

**6. Shared File** --> Adalah file yang tidak hanya diacu oleh satu direktori (pemakai), tapi juga oleh direktori-direktori (pemakai) lain. Sistem file tidak lagi berupa pohon melainkan *Directed Acyclic Graph* (DAG).

**7. Sistem Akses File** --> Cara akses perangkat penyimpanan.

Perangkat penyimpanan berdasar disiplin pengaksesan dibagi 2 (dua), yaitu :

- a. Perangkat akses sekuen (*sequential access devices*). Proses harus membaca semua *byte* atau *record file* secara berturutan mulai dari awal, tidak dapat meloncati dan membaca diluar urutan. Contoh : tape.



b. Perangkat akses acak (*random access devices*).

Dimungkinkan dapat membaca *byte* atau *record* file di luar urutan, atau mengakses rekord berdasar kunci bukan posisinya.

**8. Organisasi File** --> Terdapat enam organisasi dasar, kebanyakan organisasi file sistem nyata termasuk salah satu atau kombinasi kategori-kategori ini.

Enam organisasi atau pengaksesan dasar adalah sebagai berikut :

a. File pile (*pile*).

b. File sekuen (*sequential file*).

c. File sekuen berindeks (*indexed-sequential file*).

d. File berindeks majemuk (*multiple indexed file*).

e. File berhash (*hashed or direct file*).

f. File cincin (*multiring file*).





## TUGAS 2 -→ SEBELUM 19 OKTOBER

- Masing-masing kelompok membuat BLOG kelompok
- Di dalam blog ada minimal 3 posting/ artikel:
  1. Perkenalan, nim nama anggota kelompok (sertakan foto lebih baik)
  2. Posting ilustrasi buatan kelompok masing-masing
  3. Posting penjelasan mengenai Enam organisasi atau pengaksesan dasar
  4. URL atau alamat blog di kirim melalui komentar di blog <http://idpmengajar.wordpress.com>