

MANAJEMEN MEMORI

Memory manager :

Salah satu bagian sistem operasi yang mempengaruhi dalam menentukan proses mana yang diletakkan pada antrian.

Jenis Memori

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

Alamat Memori

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

Isi Memori

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

Fungsi manajemen memori :

- 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 dua yaitu :

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

MEMORI TANPA SWAPPING OR PAGING

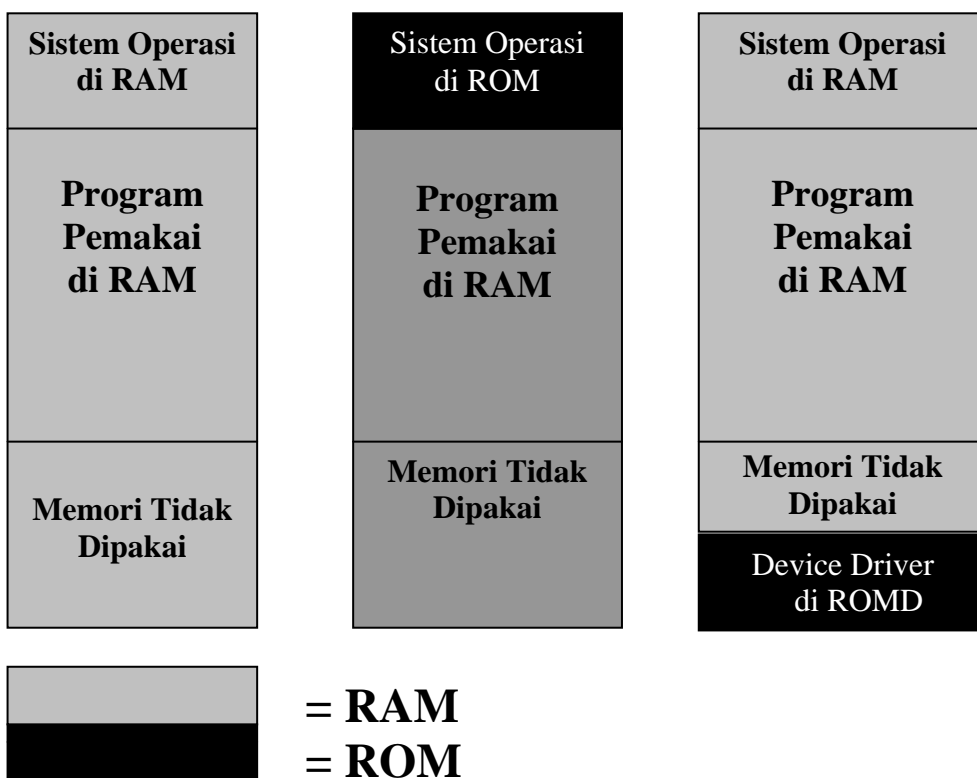
➤ Yaitu manajemen memori tanpa pemindahan citra proses antara memori utama dan disk selama eksekusi.

➤ 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



Contoh : IBM PC menggunakan cara ketiga di mana device driver ROM ditempatkan pada blok 8K tertinggi dari address space 1M. Program pada ROM disebut BIOS (Basic Input Output System).

Gambar 1. Tiga Cara Organisasi Memori Satu Proses Tunggal

(Bambang Hariyanto, hlm 154)

□ **Multiprogramming Dengan Pemartisian Statis**

Terbagi dua :

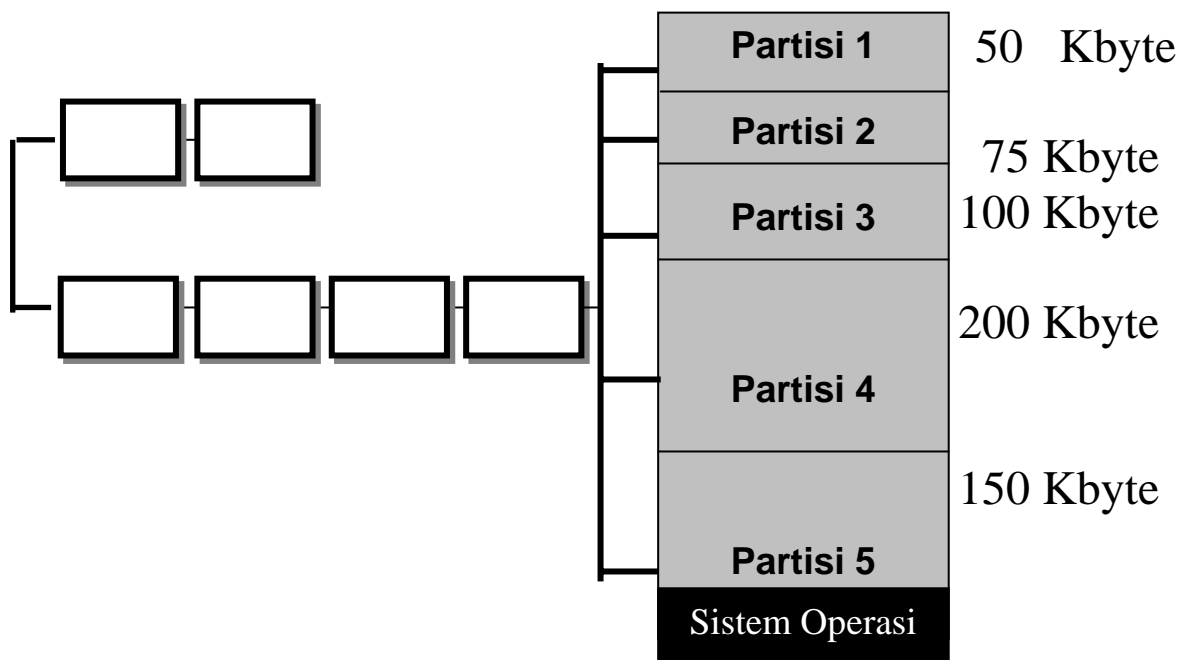
1. Pemartisian menjadi partisi-partisi berukuran sama, yaitu ukuran semua partisi memori adalah sama
2. Pemartisian menjadi partisi-partisi berukuran berbeda, yaitu ukuran semua partisi memori adalah berbeda.

Strategi Penempatan Program Ke Partisi

➤ **Satu Antrian Tunggal Untuk Semua Partisi**

Keuntungan: Lebih fleksibel serta implementasi dan operasi lebih minimal karena hanya mengelola satu antrian.

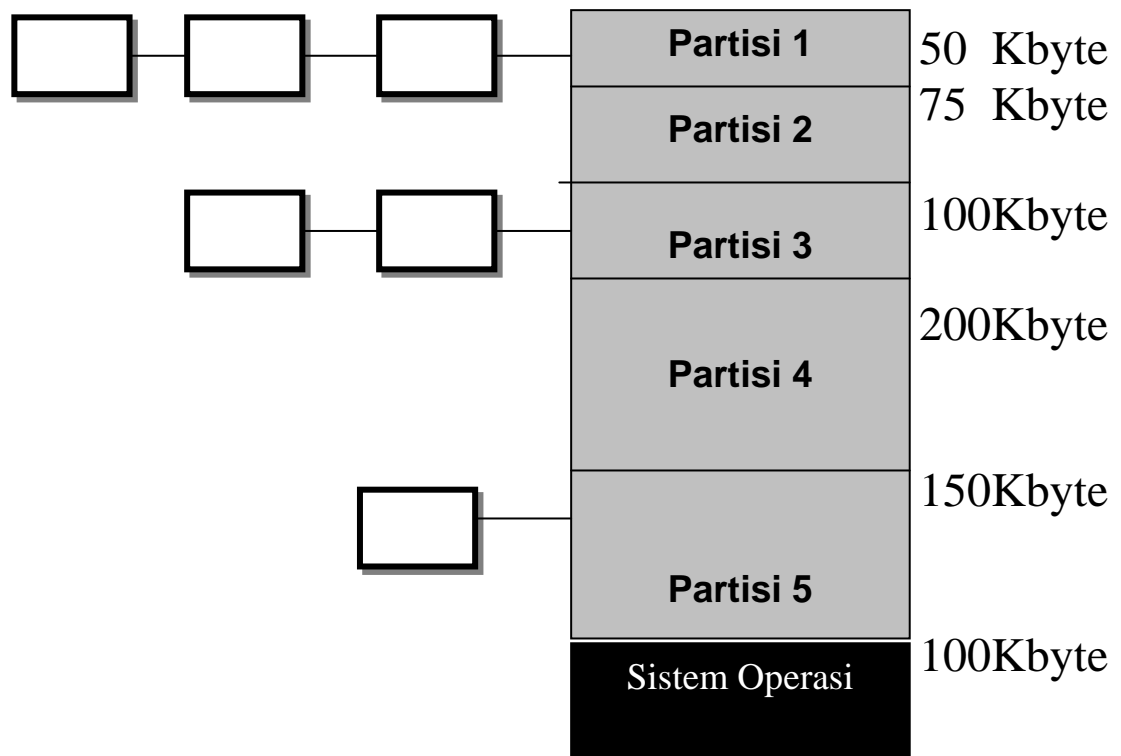
Kelemahan : Proses dapat ditempatkan di partisi yang banyak diboroskan, yaitu proses kecil ditempatkan di partisi sangat besar.



Gambar 2. Multiprogramming dengan Pemartisian Tetap dengan Satu Antrian

(Bambang Hariyanto, hlm 159)

- Satu Antrian Untuk Tiap Partisi (banyak antrian Untuk Seluruh Partisi)
Keuntungan : Meminimalkan pemborosan memori
Kelemahan : Dapat terjadi antrian panjang di suatu partisi sementara antrian partisi - partisi lain kosong



Gambar 3. Multiprogramming dengan Pemartisian Tetap dengan Banyak Antrian
 (Bambang Hariyanto, hlm. 158)

MULTIPROGRAMMING DENGAN SWAPPING

Swapping : pemindahan proses dari memori utama ke disk dan kembali lagi.

1. Multiprogramming dengan Pemartisian Dinamis

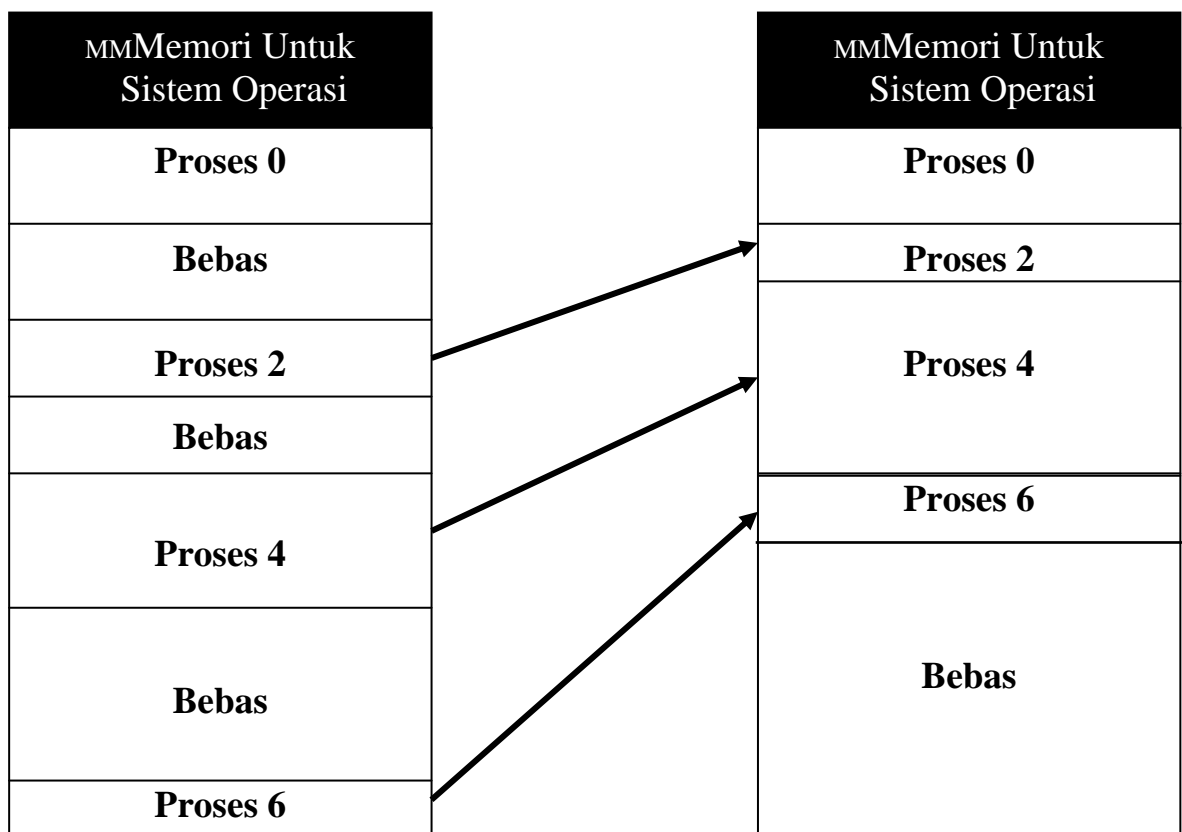
Jumlah , lokasi dan ukuran proses di memori dapat beragam sepanjang waktu secara dinamis.

Kelemahan:

- Dapat terjadi lubang-lubang kecil memori di antara partisi-partisi yang dipakai.
- Merumitkan alokasi dan dealokasi memori

Solusi:

Lubang-lubang kecil di antara blok-blok memori yang digunakan dapat diatasi dengan pemadatan memori yaitu menggabungkan semua lubang kecil menjadi satu lubang besar dengan memindahkan semua proses agar saling berdekatan.



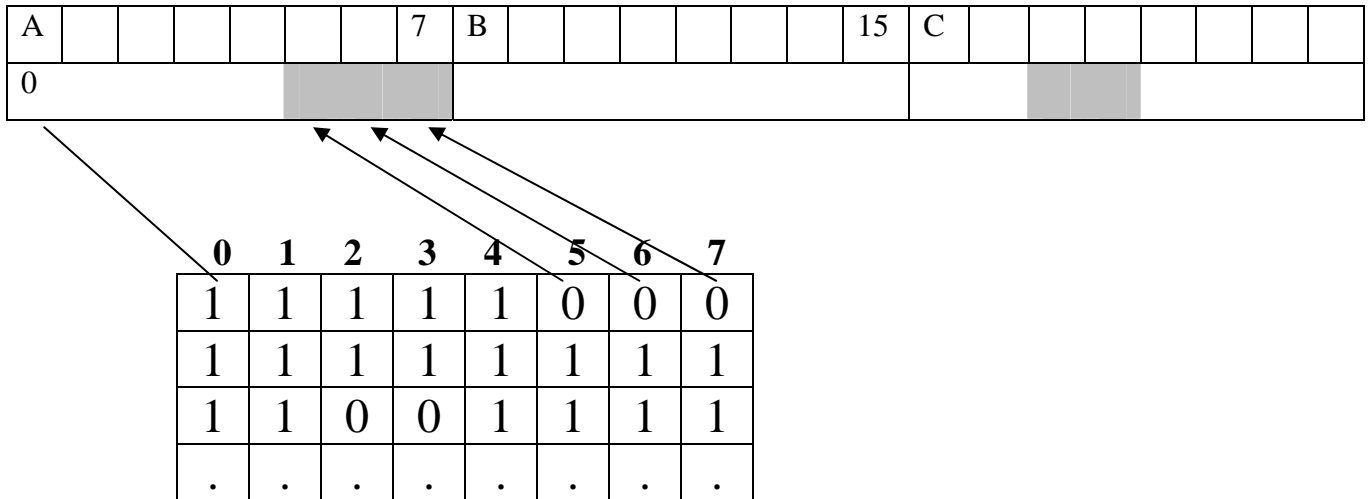
Gambar 4. Lubang-lubang Memori dan Pemadatan Memori
(Bambang Hariyanto, hlm. 165)

2. Pencatatan Pemakaian memori

- Pencatatan memakai peta bit (Bit Map)
- Pencatatan memakai linked list

Pencatatan memakai peta bit

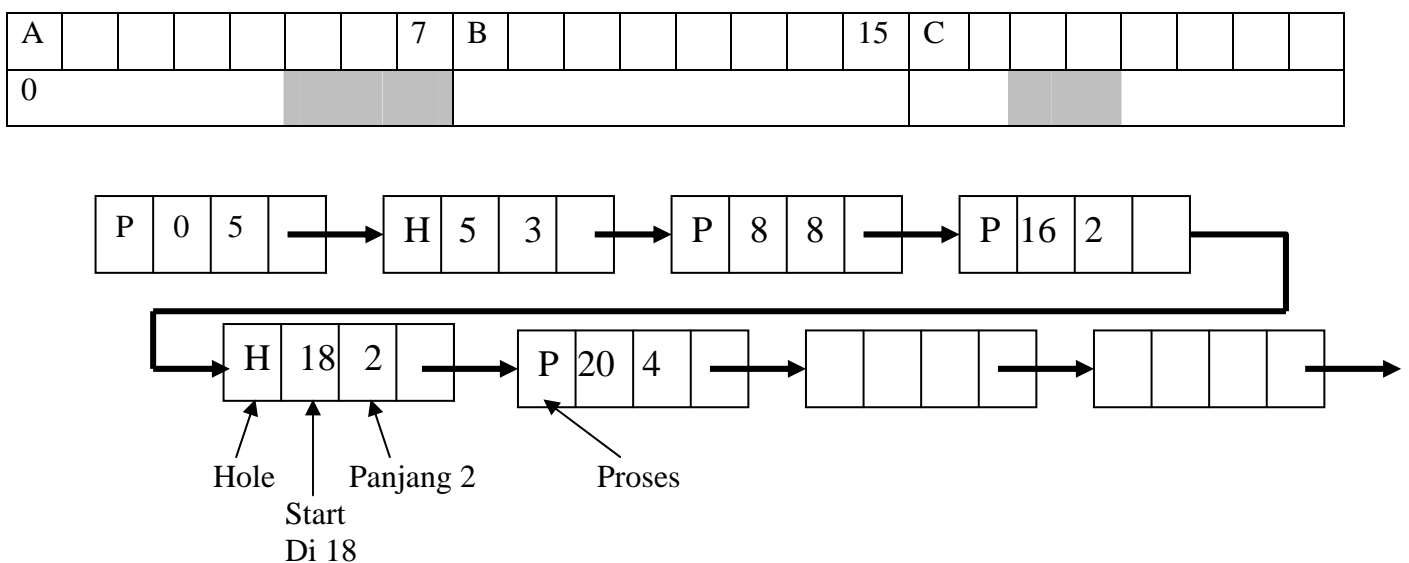
- Nilai 0 pada peta bit berarti unit masih bebas
- Nilai satu berarti unit digunakan



Gambar 5. Peta Bit untuk Pengelolaan Pemakaian Memori
(Bambang Hariyanto, hlm. 166)

Pencatatan memakai Linked List

- P= Proses, H=Hole(lubang)



Gambar 6. Pengelolaan Pemakaian dengan Linked List
(Bambang Hariyanto, hlm. 167)

3. Strategi Alokasi Memori

- First fit algorithm : memory manager men-scan list untuk menemukan hole yg cukup untuk menampung proses yg baru. Proses akan menempati hole pertama yg ditemuinya yg 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 untuk 8K, dst.

Sistem Buddy

Memori di susun dalm senari blok-blok bebas berukuran 1,2,4,8,16 byte dst, sampai kapasitas memori.

						Jmlh Hole	
Semula	1024					1	
Permintaan 70 Kb	A	128	256		512	3	
Permintaan 35 Kb	A	B	64	256	512	3	
Permintaan 80 Kb	A	B	64	C	128	512	3
A didealokasi	128	B	64	C	128	512	4
Permintaan 60 Kb	128		D	C	128	512	4
B didealokasi	128		D	C	128	512	4
D didealokasi	256		C	128	512	3	
C didealokasi	1024					1	

Gambar 7. Pengelolaan Memori dengan Sistem Buddy

(Tanenbaum, Bab 3, hlm 86)

Dari berbagai cara alokasi tsb. Di atas, sebuah hole yg 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.

Alokasi Ruang Swap pada Disk

(Penempatan proses pada disk setelah di-swap-out dari memori)

- Ruang disk tempat swap dialokasikan begitu diperlukan
- Ruang disk tempat swap dialokasikan lebih dahulu.

Algoritma untuk pengaturan ruang swap pada disk sama dengan untuk memori utama. Perbedaannya adalah ruang pada disk harus dialokasikan sebagai kelipatan bilangan bulat dari disk block.