

BAB III

MANAJEMEN MEMORI

MATERI

1. Memori manajer
2. Manajemen memori dengan swapping dan paging

STANDAR KOMPETENSI

Mengetahui tentang memori manajer dan mengerti tentang manajemen memori dengan swapping dan paging

CAPAIAN PEMBELAJARAN

1. Mahasiswa mengetahui tentang memori manajer
2. Mahasiswa mengerti tentang manajemen memori dengan swapping dan paging

1. Memory manajer :

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

Tanda pengenal untuk pencarian letak memori adalah alamat, lintasan / trek, sector pada suatu disk.

- **Jenis Memori**

- **Memori Kerja**

Tugas utamanya adalah menampung pekerjaan itu pada saat sebelum dan sesudah pekerjaan itu dilaksanakan oleh prosesor dan menampung

berbagai hal yang diperlukan prosesor, contohnya system operasi, system bahasa, catatan.

✓ *Contoh memori kerja untuk memori tetap adalah :*

- ROM : Read Only Memory → memori baca saja
- PROM : Programmable ROM → dapat diprogram tetapi tidak dapat dihapus lagi
- EPROM : *Electrically PROM* → dapat diisi melalui listrik, dan dapat dihapus
- EEPROM : *Erasable EPROM* → dapat diisi dan dihapus dengan listrik, maka ciri utamanya adalah isi tetap ada / tidak mudah dihapus meskipun daya listrik computer terputus.

✓ *Contoh memori kerja untuk memori bebas adalah :*

- RAM : Random Access Memory → dapat diisi dan dapat dibaca, maka ciri utamanya adalah mengenal asas pemuktahiran yaitu dapat diisi dengan informasi terbaru dan isi akan hilang jika catu daya padam.

Memori kerja terdiri dari sel memori yang berisi 1 kata sandi, misalnya sistem 8 bit menggunakan 8 bit kata sandi, sistem 1 kbyte menggunakan 1024 byte kata sandi.

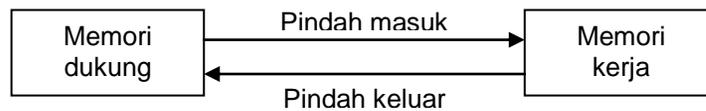
✓ *Cache memory*

Memori berkapasitas kecil tetapi berkecepatan tinggi, yang dipasang diantara prosesor dan memori utama. Instruksi dan data yang sering diakses oleh prosesor ditempatkan dalam chace sehingga dapat lebih cepat diakses oleh prosesor. Hanya bila data / instruksi yang diperlukan tidak tersedia dalam chace barulah prosesor mencarinya dalam memori utama.

○ **Memori Dukung / backing store**

Contohnya : Floppy, Harddisk, CD, dll.

Untuk mendukung memori kerja, umumnya berbentuk disk sehingga berlaku juga asas pemuktakhiran. Setiap trek dan sektor dapat menyimpan sejumlah byte dari memori kerja. Memori kerja dicapai melalui alamat memori dan register data memori. Dan untuk mencapai informasi di memori dukung, isinya harus dipindahkan dulu ke memori kerja (memori dukung = memori semu = virtual memori).



1 pindahan = 1 blok, makin kecil ukuran memori kerja, makin sering terjadi pindahan.

- **Alamat Memori**

- **Alamat memori mutlak (alamat fisik)**

Sel memori pada memori kerja adalah sumber daya berbentuk fisik, sehingga untuk mencapai sel memori ini digunakan kata pengenal. Maka disebutlah alamat fisik dan karena nomor alamat fisik ini bersifat mutlak (nomor setiap sel adalah tetap), maka disebut juga alamat mutlak.

- **Alamat memori relatif (alamat logika)**

Alamat memori yang digunakan oleh program / data berurutan / berjulat. Jika kita menggunakan alamat 1, maka kitapun menggunakan alamat 2,3, ... dan untuk 1 informasi jika alamat awalnya 0 dan alamat lainnya relatif terhadap alamat awal 0 ini, maka dinamakan alamat relatif. Dan alamat tersebut adalah logika dari untaian alamat yang menyimpan informasi maka dikenal alamat memori logika.

Contoh : alamat awal relatif 0, alamat awal fisik 14726, maka selisihnya = relokasinya = $14726 - 0 = 14726$.

Alamat relatif	Alamat mutlak	Relokasi
0	14726	14726
1	14727	14726
2	14728	14726
3	14729	14726
...

- **Isi Memori**

- ***Sistem bahasa pemrograman***

Translator tergantung pada program yang digunakan dan memiliki run time subroutine yang mengatur program dari bahasa pemrogramannya untuk melakukan translator.

- ***Sistem utilitas***

Merupakan bagian dari sistem operasi komputer, biasanya berbentuk berkas dalam memori arsip.

- ***Inti sistem operasi***

Terdiri dari inti, kernel, nucleus adalah sistem operasi yang menetapkan dalam memori kerja. Contohnya adalah pada saat “boot / booting / bootstrap”.

- ***Sistem operasi***

Program sistem operasi diletakkan dalam memori, maka banyak kegiatan yang juga terletak dalam memori.

- ***Pengendali alat***

Terdiri dari penggerak alat / device controller dan pelaksana / device driver yang berbentuk program dan menempati ruang memori tertentu.

- ***Berkas pemakai***

Terdiri dari berkas program dan berkas data, yang ditampung dalam memori dan pada waktunya diteruskan ke prosesor untuk diolah.

- **Fungsi manajemen memori**

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

- **Pemuatan informasi ke memori**

Pemuat / loader, menyalin informasi ke memori kerja.

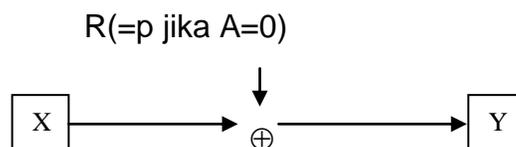
- ***pemuatan mutlak***

jika 2 macam alamat memori dibuat sama antara yang tercantum dalam program dan yang ditempati dalam memori kerja. Contohnya pada program, alamat awal 5235 dan subroutine = 5468 diletakkan pada alamat fisik yang sama.

- ***pemuatan relokasi***

alamat program tidak selalu sama dengan alamat fisik.

Contohnya alamat pangkal / fisik = 14200, alamat relative program 327, maka alamat mutlaknya = $14200 + 327 = 14527$



Keterangan:

R : relokasi

P : alamat pangkal / fisik

A : alamat awal / logika

Y : alamat fisik dari X

X : alamat suatu informasi tertentu (logika)

$$\text{Maka : } R = P - A \qquad Y = X + R = X + P - A$$

Jika informasi dengan alamat relative = 0, maka $A = 0$.

Contoh : suatu informasi beralamat awal 2100 dan informasi ini dimuat relokasikan ke alamat pangkal 23600. Hitung besar relokasi dan hitung pula pada memori kerja dimana bagian informasi yang beralamat 2453 terletak ?

$$\text{Maka : } A = 210 \qquad P = 23600 \qquad X = 2453$$

$$R = P - A = 23600 - 2100 = 21500$$

$$Y = X + R = 2453 + 21500 = 23953$$

- ***pemuatan sambung***
penyambungan antara informasi yang satu dengan yang lain terjadi pada saat pemuatan. Sehingga membutuhkan informasi sebelum melakukan pemuatan sambung tersebut, contohnya informasi alamat akhir dari bagian informasi yang akan disambung tersebut.
- ***pemuatan dinamik / tumpang / overlay***
terjadi pada program tunggal ketika ukuran program melampaui ukuran ruang memori kerja. Dan agar dapat tertampung maka dibagi / dipenggal menjadi beberapa segmen.
 - Sambung / linker akan menyatukan 1 informasi dengan informasi lainnya.
 - Lama muat, ditentukan oleh kecepatan baca pada memori dukung yang berbentuk disk.
- **Pengutipan / pembacaan isi memori**
 - ***Pengutipan oleh prosesor***
Untuk melaksanakan pekerjaan yang dalam informasi itu, yang merupakan tugas dan proses dan dilakukan sel memori per sel memori.
 - ***Pengutipan oleh memori dukung / arsip***

Untuk menyimpan informasi tersebut di dalam memori dukung / arsip sehingga mereka tidak hilang ketika catu daya dimatikan. Isi memori kerja disimpan ke dalam disk memori dukung / disket memori arsip. Kemudian di dalam disk / disket, informasi direkam ke dalam sector, setelah 1 sektor penuh, maka berpindah ke sektor lainnya. Pengutipan ini dilakukan blok demi blok yang biasanya merupakan kelipatan bulat dari ukuran sektor.

- ***Pengutipan oleh pencetak / printer***

Untuk mencetak isi memori ke atas kertas, pengutipan secara blok demi blok.

- ***Pengutipan oleh alat peripheral lainnya***

Contoh pada komunikasi data melalui jaringan, pengutipan mengikuti suatu protocol tertentu, misalnya protocol OSI (open system Interconnection) dari ISO (international standard organization) yang terdiri dari 7 lapis / layer.

- **Akses memori langsung / DMA (Direct Memory Access)**

Melakukan pemindahan informasi dari dan ke memori kerja tanpa campur tangan langsung dari prosesor. Prosesor hanya diinterupsi saat akses tersebut dimulai dan selesai.

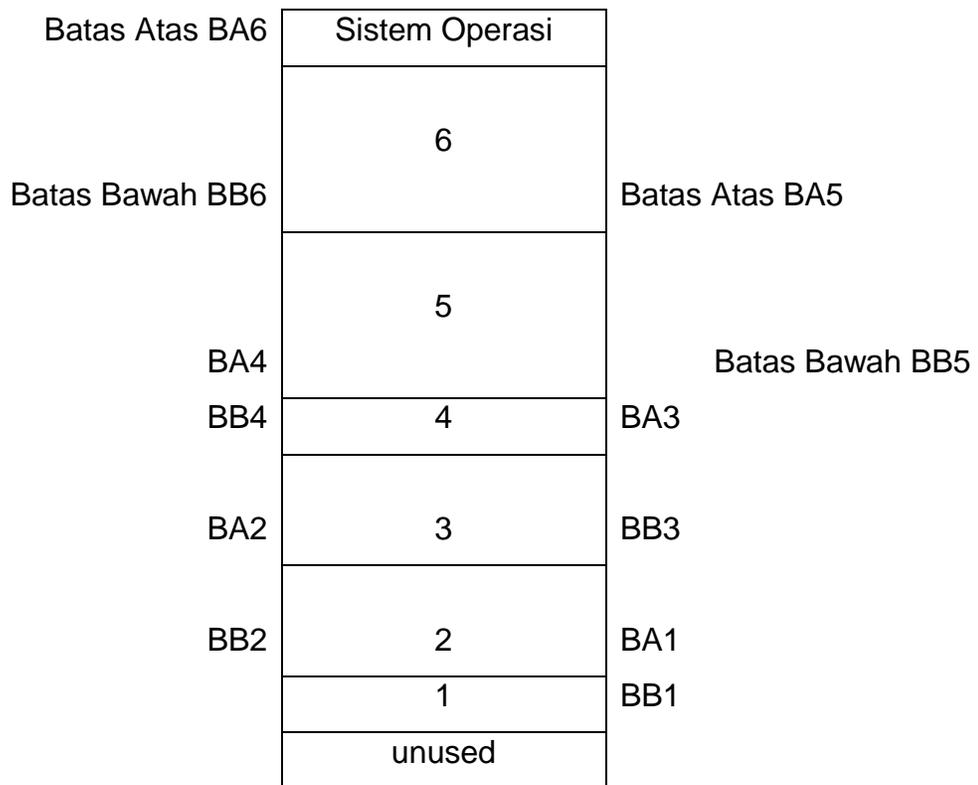
Contohnya alamat pagar untuk pemenggalan memori kerja untuk wilayah program monitor, program dan data user.

- ***Pagar pada memori kerja***

Wilayah memori telah kita penggal menjadi 2 bagian yaitu di atas alamat pagar dan di bawah alamat pagar yang letaknya terpisah.

- ***Pemenggalan memori kerja***

Memenggal memori ke dalam banyak penggalan dan setiap penggalan dibatasi 2 pagar, yaitu batas bawah dan batas atas.



Untuk setiap penggalan memori kerja, memerlukan 2 register yaitu untuk mencatat batas atas dan untuk mencatat batas bawah. Dan setiap alamat memerlukan dua kali perbandingan untuk mencapai penggalan.

- **Relokasi ke penggalan memori kerja**

Dapat memilih penggalan yang menjadi tujuan relokasi dan harus diperiksa panjang informasi dan ukuran penggalan memorinya.

Karena jika masuk pada penggalan yang kecil, informasi tidak dapat tertampung.

Contoh : Jika alamat awalnya = 0, alamat pangkal P maka $A = 0$ dan $P = BB$. Sehingga $R = P - A = BB$

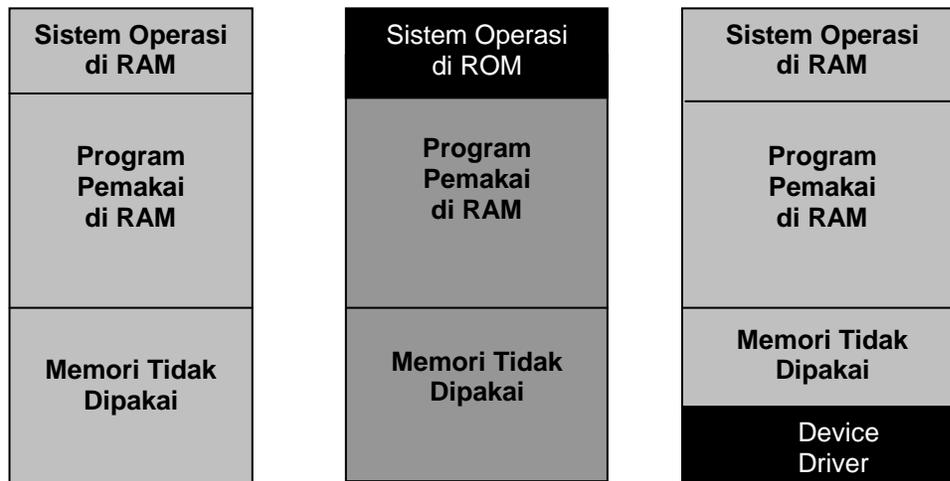
Jika alamat X tidak melampaui batas (BA-BB) maka pemuatan secara relokasi dapat dilaksanakan yaitu dengan menambahkan alamat pangkal (BB) dari penggalan memori kerja.

2. Manajemen Memori dengan Swapping dan Paging

- **Memori Tanpa Swapping atau Paging**

Yaitu manajemen memori tanpa pemindahan citra proses antara memori utama dan disk selama eksekusi, Terdiri dari :

- **Monoprogramming**



= RAM

= ROM

Tiga Cara Organisasi Memori Satu Proses Tunggal

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).

Ciri-ciri :

- Hanya satu proses pada satu saat
- Hanya satu proses menggunakan semua memori

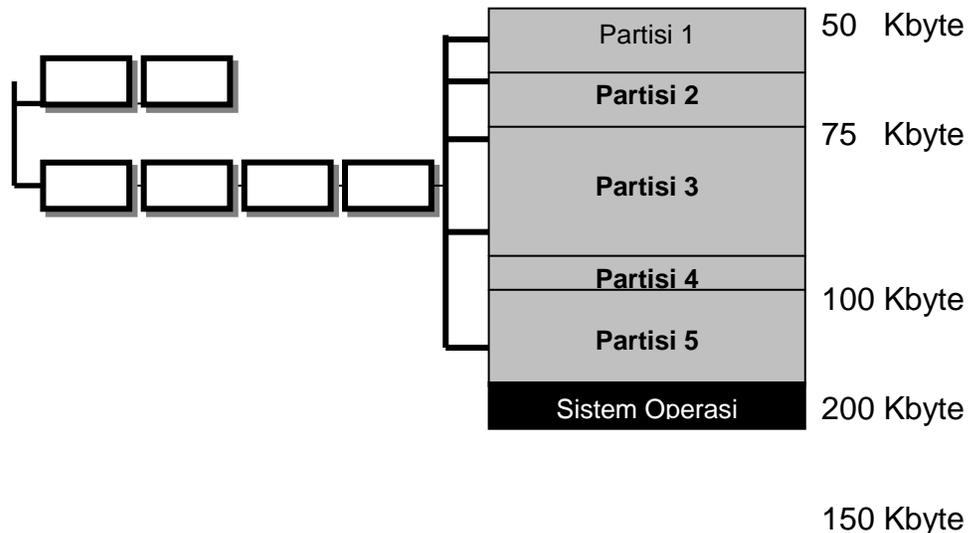
- Pemakai memuatkan program ke seluruh memori dari disk atau tape
- Program mengambil kendali seluruh mesin
- **Multiprogramming Dengan Pemartisian Statis**
Terbagi dua :
 - Pemartisian menjadi partisi-partisi berukuran sama, yaitu ukuran semua partisi memori adalah sama
 - Pemartisian menjadi partisi-partisi 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.

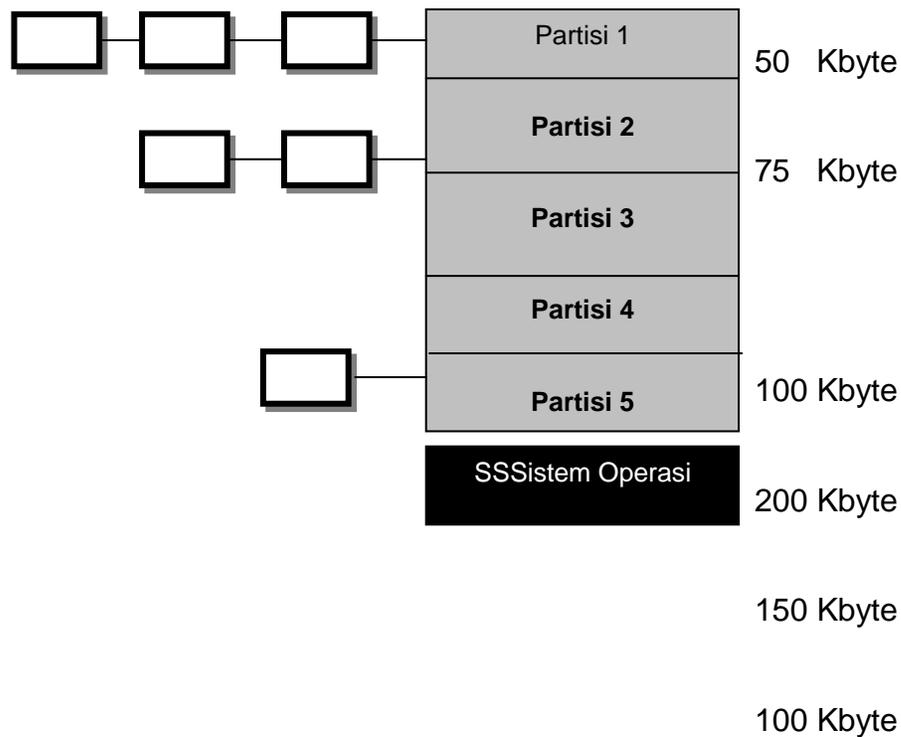


Multiprogramming dengan Pemartisian Tetap dengan Satu Antrian

- **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



Multiprogramming dengan Pemartisian Tetap dengan Banyak Antrian

- **Multiprogramming Dengan Swapping**

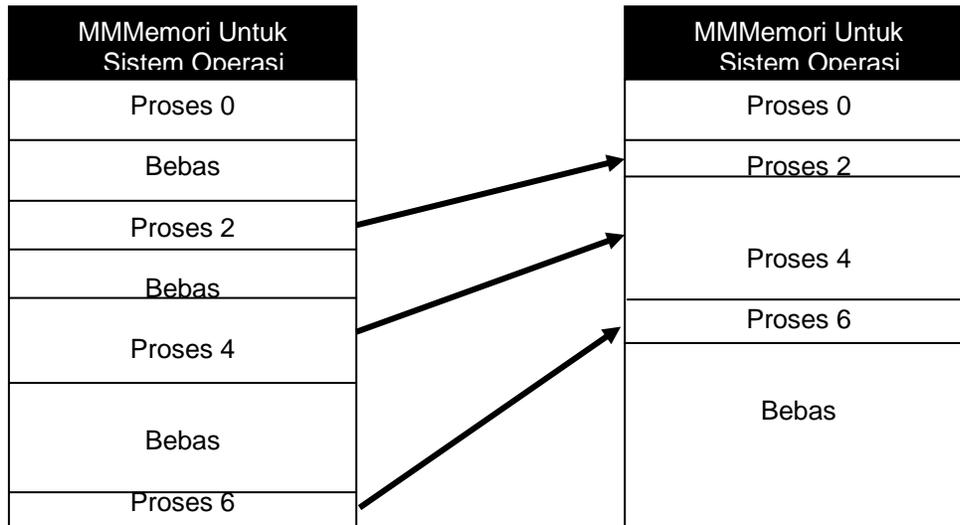
Swapping : pemindahan proses dari memori utama ke disk dan kembali lagi.

- **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



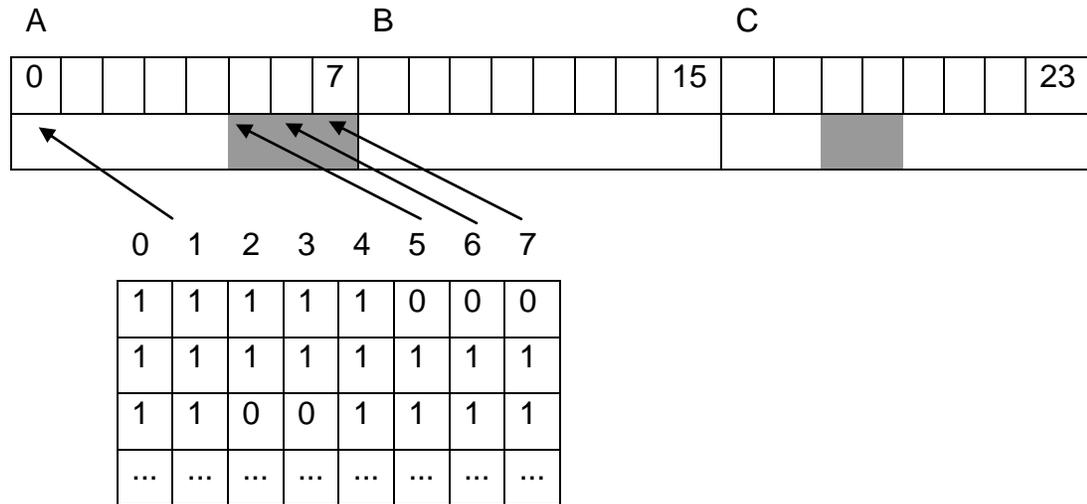
Lubang-lubang Memori dan Pemadatan Memori

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.

- **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 / terisi

Setelah penggalan memori rampung dipakai, maka keadaan bit akan berubah dari 1 ke 0, begitupun sebaliknya. Jika jumlah penggalan memori

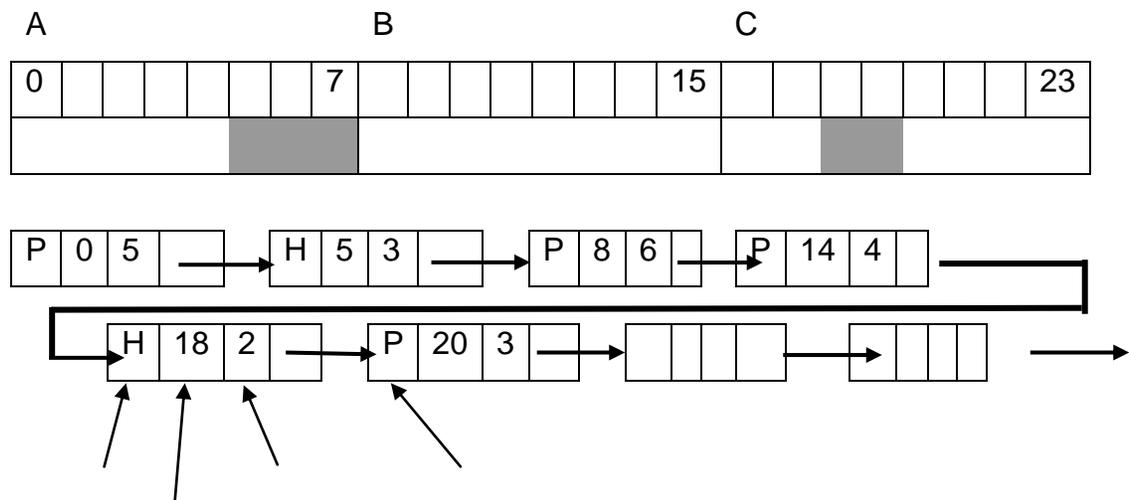
kerja benar, maka bit menjadi panjang dan pemeriksaan keadaan penggalan memori kerja akan lama.



Peta Bit untuk Pengelolaan Pemakaian Memori

✓ Pencatatan memakai Linked List

P= Proses, H=Hole(lubang)



Hole Panjang=2 Proses
Start
Di 18

Pengelolaan Pemakaian dengan Linked List

- **Strategi Alokasi Memori**

- ✓ Pencocokan / FIT

Karena ukuran penggalan memori tidak selalu sama besar, maka dibutuhkan pencocokan ukuran dari memori dukung dengan ukuran penggalan memori di memori kerja.

- Cocok pertama / First fit algorithm

Pencocokan terjadi menurut antrian informasi, informasi yang terdepan dicocokkan terlebih dahulu dan menurut urutan penggalan, penggalan yang terdepan dicocokkan terlebih dahulu.

- Cocok pertama berdaur / Cyclical first fit

Pencocokan tidak harus dimulai dari urutan penggalan memori pertama.

Contoh : informasi 2K telah menempati penggalan 4 K, pencocokan cocok pertama berdaur bagi informasi 1K tidak lagi mencoba penggalan 4K tetapi 3K dan karena penggalan memori ini dapat menampung informasi tersebut, maka pada cocok pertama berdaur informasi 1K menempati memori 3K.

- Cocok terbaik / best fit

Mencari penggalan memori yang dapat menampung informasi yang paling pas / tidak ada memori di penggalan yang tersisa / sisanya yang paling kecil.

Contohnya informasi pertama 2K mencari penggalan informasi yang terpas yaitu pada penggalan memori 2K juga / tidak ada sisa.

- Cocok terburuk / worst fit

Kebalikan dari cocok terbaik yaitu akan menempati penggalan yang ukurannya terbesar sehingga banyak ruang memori pada penggalan itu yang tidak terpakai.

- **Sistem Buddy**

Memori di susun dalam senarai blok-blok bebas berukuran 1,2,4,8,16 Kbyte dst, sampai kapasitas memori.

Permintaan : A = 70 Kb; B = 35 Kb; C = 80 Kb; D = 60 Kb

1024					
A	128		256		512
A	B	64	256		512
A	B	64	C	128	512
128	B	64	C	128	512
128		D	C	128	512
128		D	C	128	512
256			C	128	512
1024					

Pengelolaan Memori dengan Sistem Buddy

Dari berbagai cara alokasi tersebut 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*.

- **Fragmentasi internal**

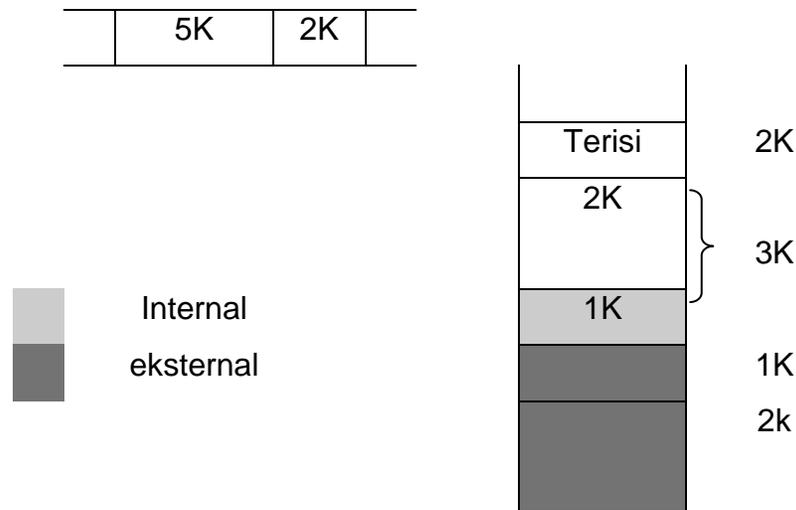
Suatu informasi menempati penggalan yang lebih besar ukurannya akan menyebabkan adanya sisa dalam penggalan memori tersebut. Sisa tersebut dinamakan fragmentasi internal.

- **Fragmentasi eksternal**

Penggalan memori bebas yang ukurannya terlalu kecil untuk dapat menampung penggalan informasi yang akan dimuat ke penggalan memori itu. Misalnya informasi 5K akan menempati penggalan memori yang belum terisi yaitu 1K dan 2K tetapi karena ukurannya tidak sesuai maka

pemuatan informasi ke penggalan memori tidak dapat dilaksanakan, maka penggalan memori 1K dan 2K(=3K) menjadi fragmentasi eksternal.

Contoh :



- **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.

CONTOH SOAL

1. Jelaskan apa yang dimaksud dengan alamat logika dan alamat fisik.

Alamat yang dihasilkan oleh CPU berupa alamat logika, sedangkan yang masuk ke dalam memori adalah alamat fisik. Pada compile time dan load time, alamat fisik dan logika identik. Sebaliknya, perbedaan alamat fisik dan logika terjadi pada execution time.

Kumpulan semua alamat logika yang dihasilkan oleh program adalah ruang alamat logika/ruang alamat virtual. Kumpulan semua alamat fisik yang berkorespondensi dengan alamat logika disebut ruang alamat fisik.

Pada saat program berada di CPU, program tersebut memiliki alamat logika, kemudian oleh MMU dipetakan menjadi alamat fisik yang akan disimpan di dalam memori.

Ilustrasinya sebagai berikut, nilai pada register ini akan ditambah dengan setiap alamat yang dibuat oleh user process yang kemudian dikirim ke memori. Contohnya register relokasi berada di 14000, alamat logika di 346, maka langsung dipetakan menjadi alamat fisik di 14346.

2. Apa perbedaan dari first-fit, best-fit dan worst-fit.

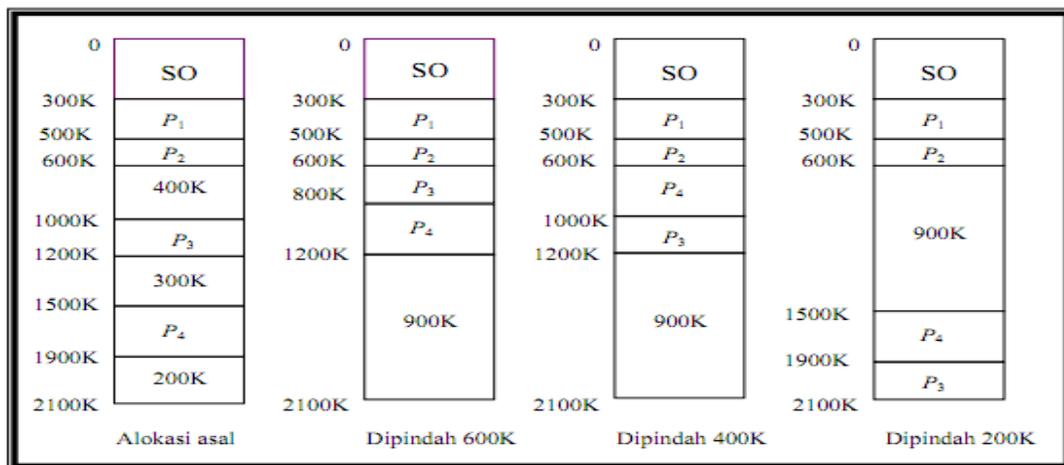
Diantara ketiganya first-fit dan best-fit lebih baik dibandingkan worst-fit dalam hal menurunkan waktu dan utilitas penyimpanan. Tetapi first-fit dan best-fit lebih baik dalam hal utilitas penyimpanan tetapi first-fit lebih cepat.

3. Jelaskan solusi untuk masalah fragmentasi eksternal.

Dengan teknik pemadatan (compaction) yaitu memadatkan sejumlah lubang kosong menjadi satu lubang besar sehingga dapat digunakan

untuk proses. Pemadatan tidak selalu dapat dipakai. Agar proses dapat dieksekusi pada lokasi baru, semua alamat internal harus direlokasi.

Pemadatan hanya dilakukan pada relokasi dinamis dan dikerjakan pada waktu eksekusi. Karena relokasi membutuhkan pemindahan program dan data dan kemudian mengubah register basis (atau relokasi) yang mencerminkan alamat basis baru. Terdapat beberapa cara pemadatan seperti pada gambar di bawah ini:



EVALUASI

1. Jelaskan apa yang dimaksud dengan memori kerja dan berikan contohnya
2. Jelaskan apa yang dimaksud dengan alamat memori dan berikan contohnya
3. Jelaskan apa yang dimaksud dengan isi memori dan berikan contohnya
4. Sebutkan fungsi dari manajemen memori
5. Jelaskan bagaimana pemuatan informasi ke memori
6. Jelaskan bagaimana Pengutipan / pembacaan isi memori
7. Jelaskan bagaimana Akses memori langsung / DMA (Direct Memory Access)

8. Jelaskan tentang monoprogramming dan berikan contohnya
9. Jelaskan tentang Strategi Penempatan Program Ke Partisi dan sebutkan untung ruginya
10. Jelaskan apa yang dimaksud dengan Multiprogramming Dengan Swapping
11. Jelaskan apa yang dimaksud dengan System Buddy

DAFTAR PUSTAKA :

- Harvey M Deitel dan Paul J Deitel. 2005. *Java How To Program*. Sixth Edition. Prentice Hall.
- Bambang Hariyanto. 1997. *Sistem Operasi*. Buku Teks Ilmu Komputer. Edisi Kedua. Informatika. Bandung.
- John L Hennessy dan David A Patterson. 2002. *Computer Architecture . A Quantitative Approach*. Third Edition. Morgan Kaufman. San Francisco.
- Randall Hyde. 2003. *The Art of Assembly Language*. First Edition. No Strach Press.
- Kenneth H Rosen. 1999. *Discrete Mathematics and Its Application*. McGraw Hill.
- Ronald L Krutz dan Russell D Vines. 2001. *The CISSP Prep Guide Mastering the Ten Domains of Computer Security*. John Wiley & Sons.
- Sri Kusumadewi. 2000. *Sistem Operasi*. Edisi Dua. Graha Ilmu. Yogyakarta.
- Robert Love. 2005. *Linux Kernel Development*. Second Edition. Novell Press.
- Larry L Peterson dan Bruce S Davie. 2000. *Computer Networks A Systems Approach*. Second Edition. Morgan Kaufmann.
- Riri Fitri Sari dan Yansen. 2005. *Sistem Operasi Modern*. Edisi Pertama. Andi. Yogyakarta.

- Betha Sidik. 2004. *Unix dan Linux*. Informatika. Bandung.
- Abraham Silberschatz, Peter Galvin, dan Greg Gagne. 2002. *Applied Operating Systems*. Sixth Edition. John Wiley & Sons.
- Avi Silberschatz, Peter Galvin, dan Grag Gagne. 2005. *Operating Systems Concepts*. Seventh Edition. John Wiley & Sons.
- William Stallings. 2001. *Operating Systems: Internal and Design Principles*. Fourth Edition. Edisi Keempat. Prentice-Hall International. New Jersey.
- Andrew S Tanenbaum dan Albert S Woodhull. 1997. *Operating Systems Design and Implementation*. Second Edition. Prentice-Hall.
- Andrew S Tanenbaum. 2001. *Modern Operating Systems*. Second Edition. Prentice-Hall.