Organisasi Sistem Komputer

OSK 6 – Dukungan Sistem Operasi

Muh. Izzuddin Mahali, M.Cs.

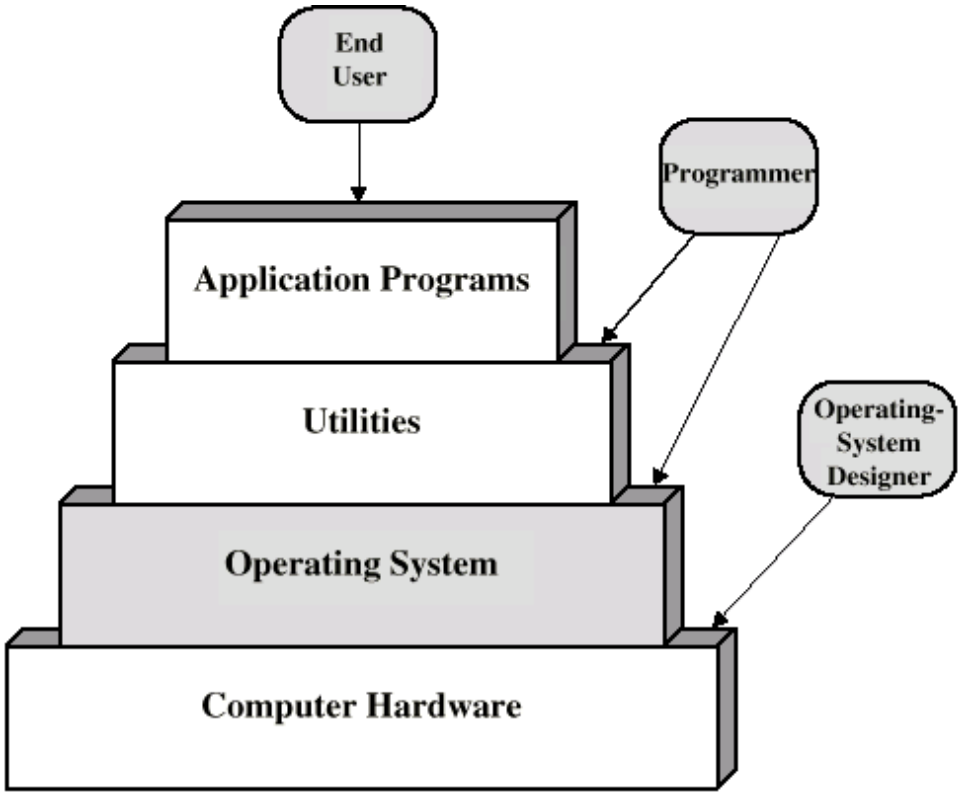
Apakah Sistem Operasi Itu ?

- Sebuah program yang mengontrol eksekusi program aplikasi dan berperan sebagai interface (antar-muka) antara pengguna komputer dengan hardware. Tujuannya:
  - **Convenience**

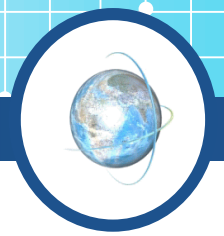
    Membuat komputer lebih mudah digunakan, lebih mudah diperintah oleh pengguna (bukan programmer)
  - **Efficiency**

    Mengatur penggunaan resources (komponenkomponen yang digunakan dalam eksekusi proses) pada komputer dengan efisien.

Letak Sistem Operasi Dalam Hirarki Sistem Komputer



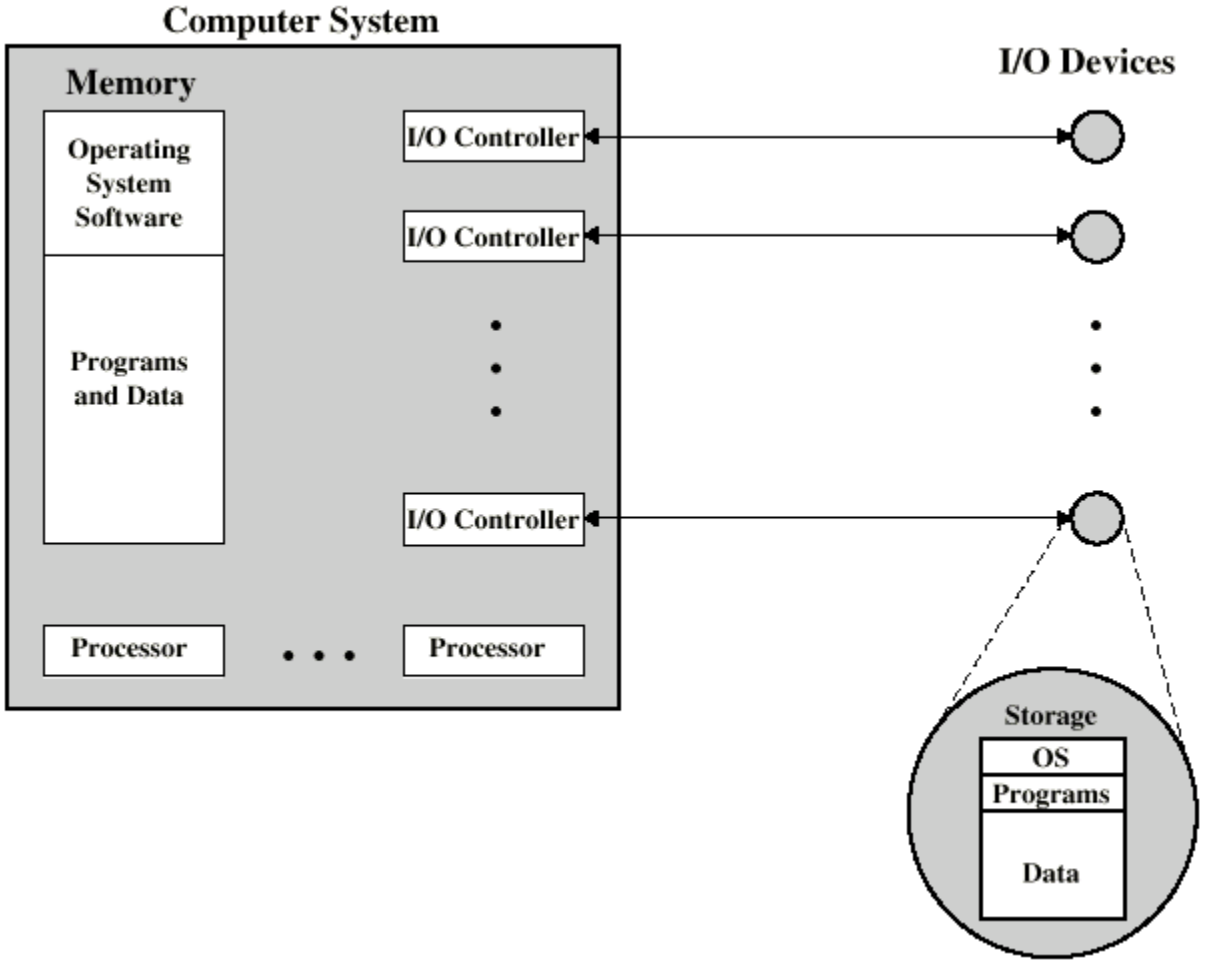
Layanan Sistem Operasi



- Pembuatan program
- Eksekusi program
- Akses ke periferal / eksternal device
- Akses terstruktur ke file di dalam periferal
- Akses ke seluruh perangkat dan resource sistem
- Deteksi error dan handlingnya
- Pengaturan sistem

- Oleh karena itu sistem operasi dapat dimanfaatkan untuk membantu mengatur organisasi resources (memory, prosesor, disk, dsb) agar bekerja lebih

O/S as a Resource Manager



Kontrol Program

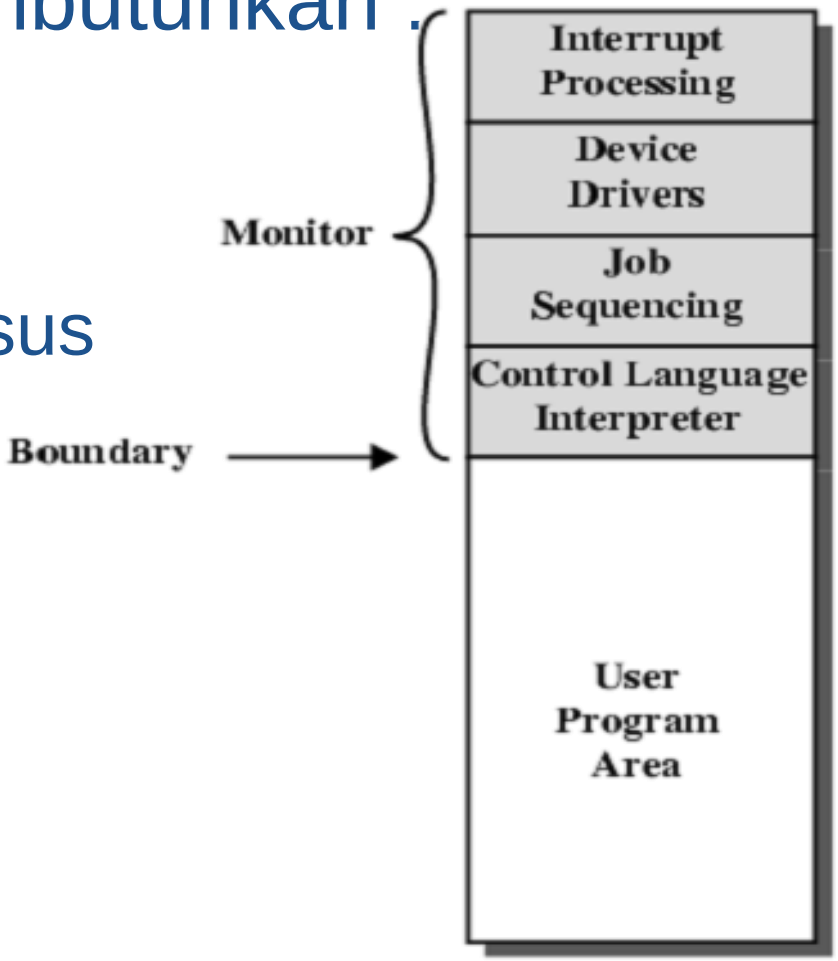
- Jaman dahulu (1940an s/d 1950an), komputer tanpa S/O, program (rangkaian proses) berinteraksi langsung dengan hardware melalui bahasa mesin
- Operator komputer ingin bisa memasukkan banyak program sekaligus tanpa harus ada operator standby: tiap satu program selesai, memasukkan program berikutnya.
- Dikembangkanlah simple batch system untuk mengatur jalannya program (inilah S/O awal)
- Bentuk modern: DOS

Kontrol Program – Batch?

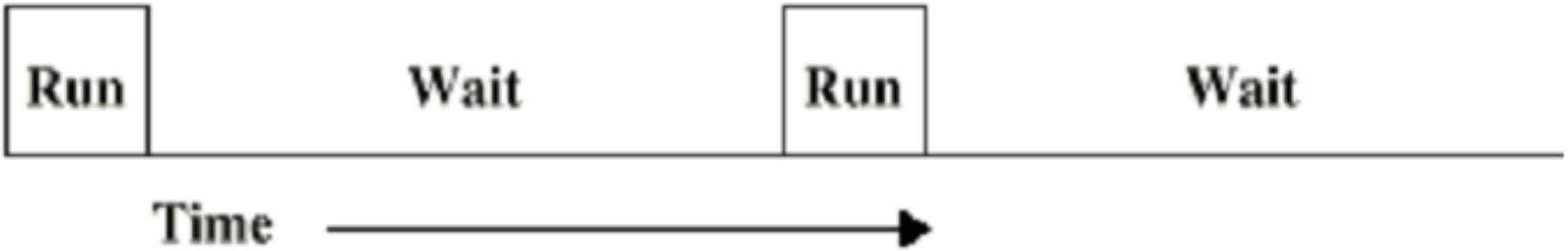
- Batch artinya berurutan, sequentially queued, Batch operating system berbentuk sebuah program stay resident di memory.
- Program/job disusun dulu oleh user dalam punched card atau magnetic tape, kemudian diberikan kepada operator
- Operator memasukkan program ke sistem
- Tiap program/job dibaca oleh monitor, disimpan ke memory
- Program/job dijalankan oleh monitor sesuai dengan urutan masuknya (batched)

Kontrol Program - Batch

- Resident Monitor membutuhkan :
  - Memory Protection
  - Timer dan Interrupt
  - Batasan Instruksi Khusus

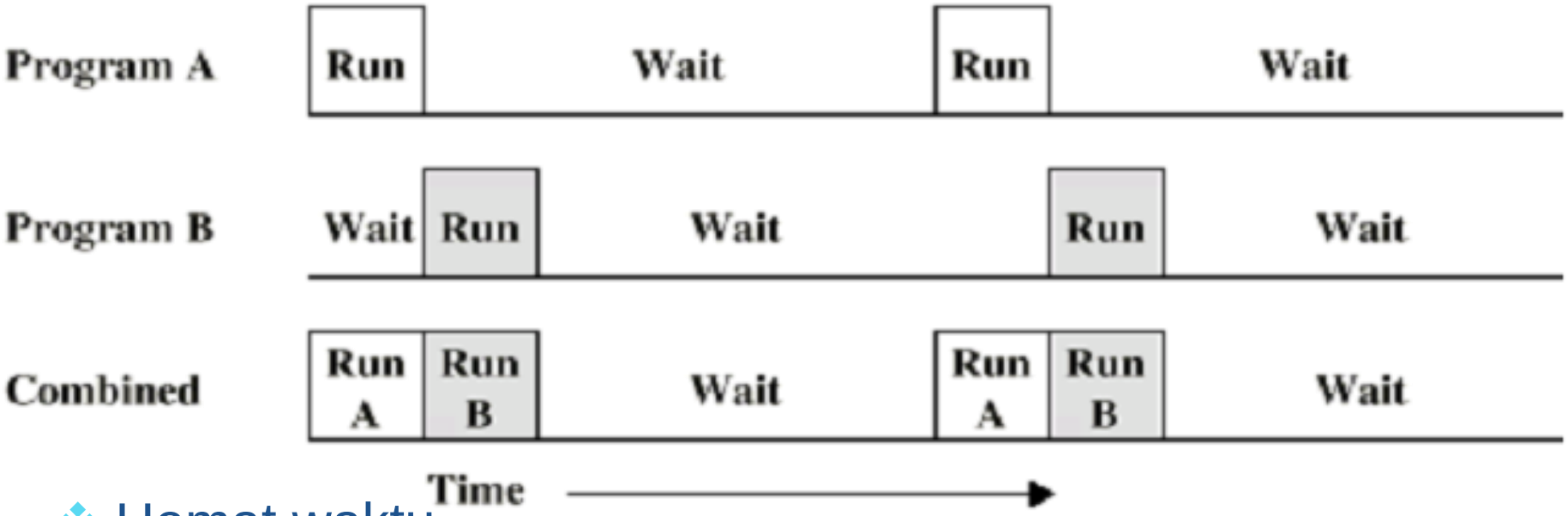


Kontrol Program – Batch Awal



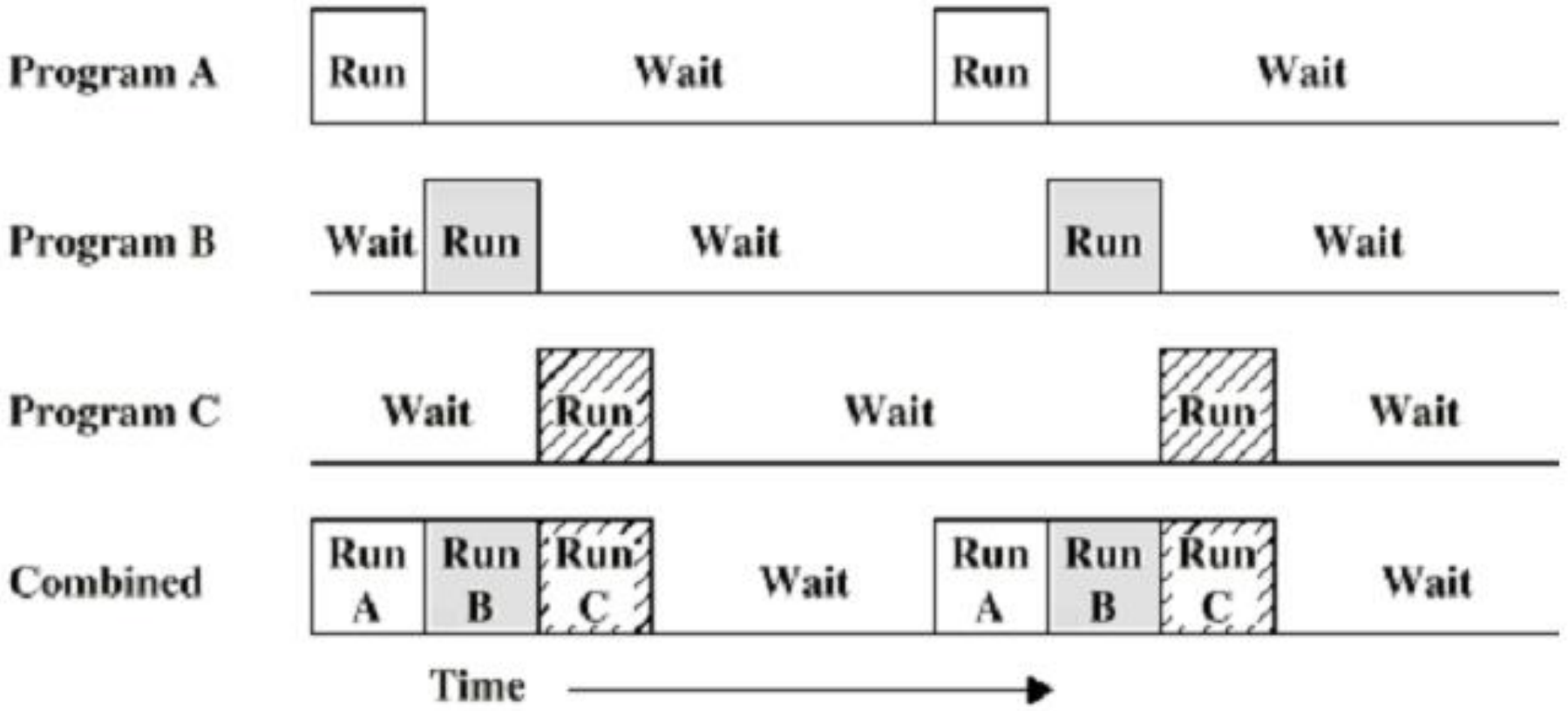
- Boros waktu
- Tidak bisa menjalankan lebih dari satu program bersamaan – **uniprogramming**

Kontrol Program – Pengembangan

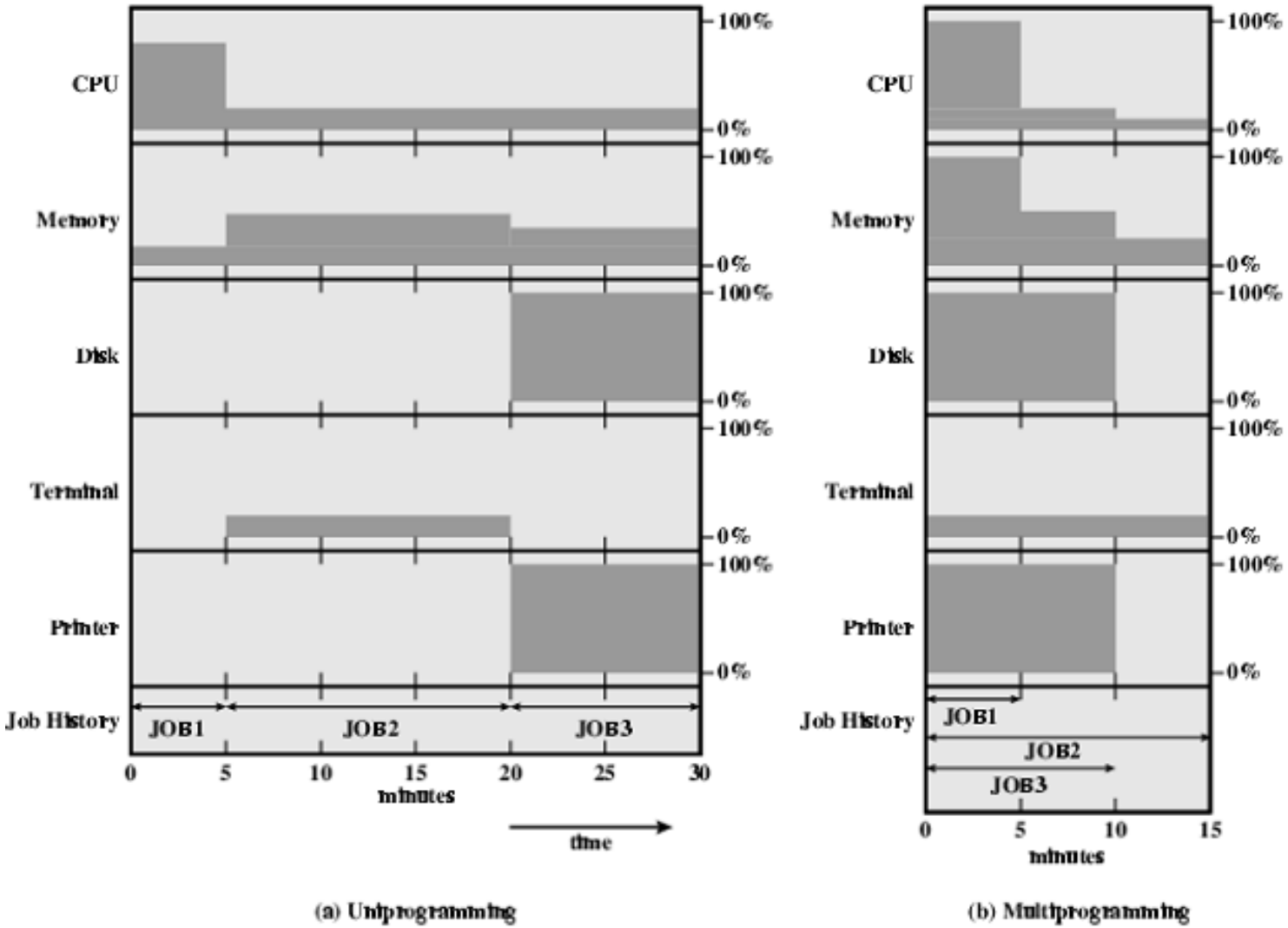


- Hemat waktu
- Lebih boros resource (pada penggunaan CPU)
- bisa menjalankan lebih dari satu program secara(seolah-olah) bersamaan – **multiprogramming**

Kontrol Program – Pengembangan



Kontrol Program – Penggunaan Resource



Penjadwalan

- Untuk mengatur susunan program/job pada multiprogramming, digunakanlah penjadwalan / scheduling
- Sistem operasi yang kita gunakan saat ini (Windows/Linux/MacOS/Solaris dkk) sudah tidak murni batched, namun sudah menggunakan teknik scheduling yang lebih maju, di mana urutan masuk program di memory tidak lagi penting untuk menentukan urutan eksekusi, tapi ditandai oleh adanya status (ready, waiting, dsb) dan event (interrupt) dari tiap proses

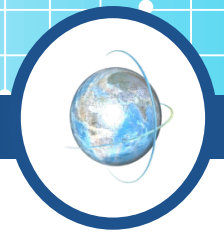
Penjadwalan Proses

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

- Sedangkan **proses** sendiri merupakan unit kerja terkecil yang secara individu memiliki sumberdaya atau unit pemilikan sumberdaya.

Tugas Penjadwalan :

- Memutuskan proses yang harus berjalan
- Memutuskan kapan dan selama berapa lama proses itu berjalan



❖ Adil (*fairness*)

Adil adalah proses –proses diperlakukan sama yaitu mendapat jatah waktu pemroses yang sama dan tak ada proses yang tak kebagian layanan pemroses sehingga mengalami (starvation).

❖ Efisiensi

Efisiensi atau utilisasi pemroses dihitung dengan perbandingan (rasio) waktu sibuk pemroses.

- Waktu Tanggap (*response time*)

Sistem Interaktif

Waktu tanggap dalam sistem interaktif didefinisikan sebagai waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukkan atau transaksi sampai hasil pertama muncul di layar (terminal).

Waktu tanggap ini disebut *terminal response time*.

Sistem Realtime

Pada sistem waktu nyata (*real-time*), waktu tanggap di definisikan sebagai waktu dari saat kejadian (internal atau eksternal) sampai instruksi pertama rutin layanan yang dimaksud dieksekusi , disebut *event respons time*.

Sasaran penjadwalan adalah meminimalkan waktu tanggap.

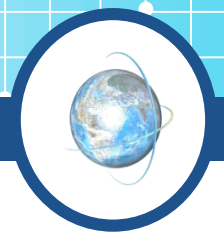
❖ *Turn Around Time*

waktu yang dihabiskan dari ssat program atau *job* mulai masuk ke sistem sampai proses diselesaikan sistem.

❖ *Throughput*

Throughput adalah jumlah kerja yang dapat diselesaikan dalam satu unit waktu.

Tipe-tipe Penjadwalan



- Penjadwalan jangka pendek (short-termscheduller)

Penjadwalan ini bertugas menjadwalkan alokasi pemroses diantara proses-proses ready di memori utama.

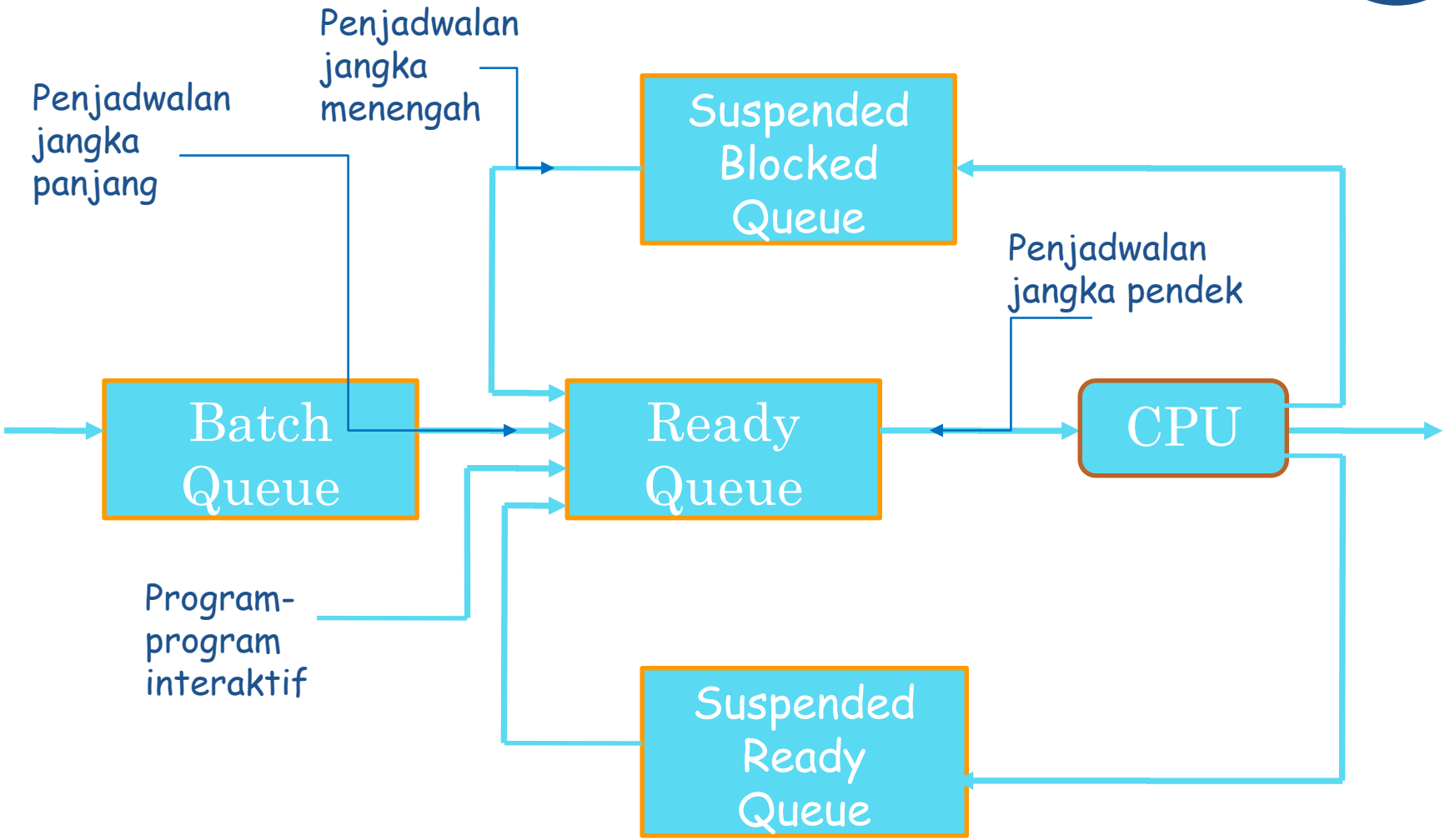
- Penjadwalan jangka menengah (medium termscheduller)

Penjadwalan jangka menengah adalah menangani proses-proses swapping (aktivitas pemindahan proses yang tertunda dari memory utama ke memory sekunder).

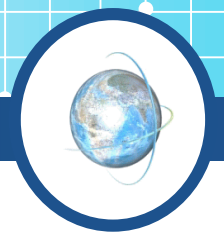
- Penjadwalan jangka panjang (long-termscheduller)

Penjadwalan jangka panjang bekerja terhadap antrian batch (proses – proses dengan penggunaan sumberdaya yang intensif) dan memilih batchberikutnya yang harus di eksekusi.

Tipe-tipe Penadwalan



Strategi Penjadwalan



- Penjadwalan *Nonpreemptive*

Begitu proses diberi jatah waktu pemroses maka pemroses tidak dapat diambil alih oleh proses lain sampai proses itu selesai

- Penjadwalan *Preemptive*

Saat proses diberi jatah waktu pemroses maka pemroses dapat diambil alih oleh proses lain sehingga proses disela sebelum selesai dan harus dilanjutkan menunggu jatah waktu pemroses tiba kembali pada proses itu

Algoritma Penjadwalan

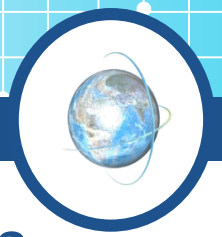
Algoritma – algoritma yang menerapkan strategi *nonpreemptive :*

- **FIFO** (*First-in, First-out*).
- **SJF** (*Shortest Job First*).
- **HRN** (*Highest Ratio Next*).
- **MFQ** (*Multiple Feedback Queues).*

Algortima – algoritma yang menerapkan strategi *preemptive :*

- **RR** (*Round-Robin*).
- **SRF** (*Shortest-Remaining-First*).
- **PS** (*Priority Schedulling*).
- **GS** (*Guaranteed Schedulling*).

Manajemen Memory

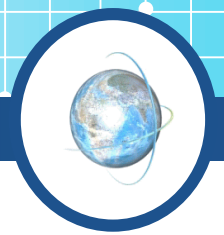


- Untuk uniprogramming, tidak ada masalah dengan memory
  - Memory dibagi menjadi dua
  - Satu bagian untuk sistem operasi (*monitor)*
  - Satu bagian untuk program yg sedang dieksekusi
- Muncul masalah pada multiprogramming:
  - Jatah memory untuk program harus dibagi antara beberapa program yang sedang dieksekusi
  - Perlu adanya pengaturan penggunaan ruang memory tersebut

Manajemen Memory - Swapping

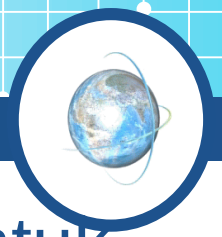
- Kadang ukuran memory yang tersedia tidak mencukupi untuk mengeksekusi semua instruksi dalam sebuah program
- Ketika memory belum kosong, proses ditampung dahulu di sebuah medium (misal hard disk)
- Saat ruang memory sudah tersedia, load proses tersebut ke memory
- Saat proses sudah selesai, buang seluruhnya dari memory
- Kalau ada proses yang blocked (stuck), pindahkan ke medium, kemudian load proses lain yang dalam kondisi ready

Manajemen Memory – Swapping?



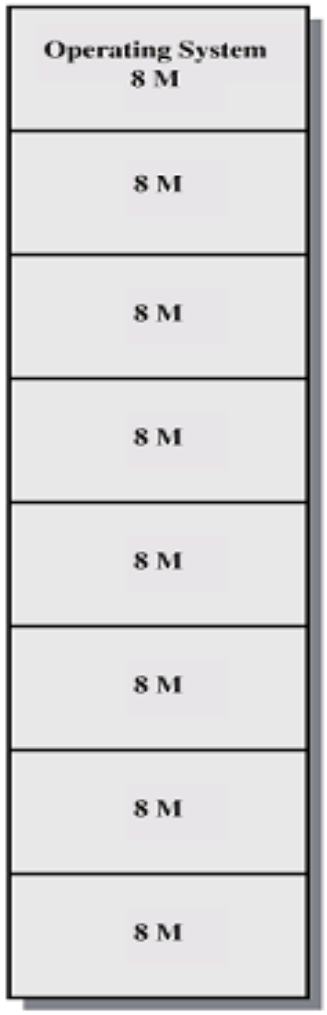
- Swapping adalah sebuah proses yang melibatkan modul I/O, memindahkan data dari storage device ke memory → melewati bus.
- Berarti swapping adalah sebuah proses yang lambat, untuk itu diperlukan metode yang tepat untuk mengatur penempatan proses ke memori, agar lebih efisien:
- Partitioning, dan
- Paging

Manajemen Memory – Partitioning



- Membagi memory menjadi bagian-bagian untuk menampung proses (termasuk milik Sistem Operasi)
- Fixed-sized partitions
  - Ukuran boleh sama atau tidak tiap partisi.
  - Proses dimasukkan di tempat kosong yang ukurannya paling mendekati
  - Ada memory yang terbuang
  - Memunculkan ide adanya variable-sized partitions → ukuran partisi disesuaikan tempat yang akan dipakai

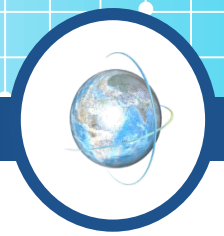
Fixed-sized partitions



(a) Equal-size partitions

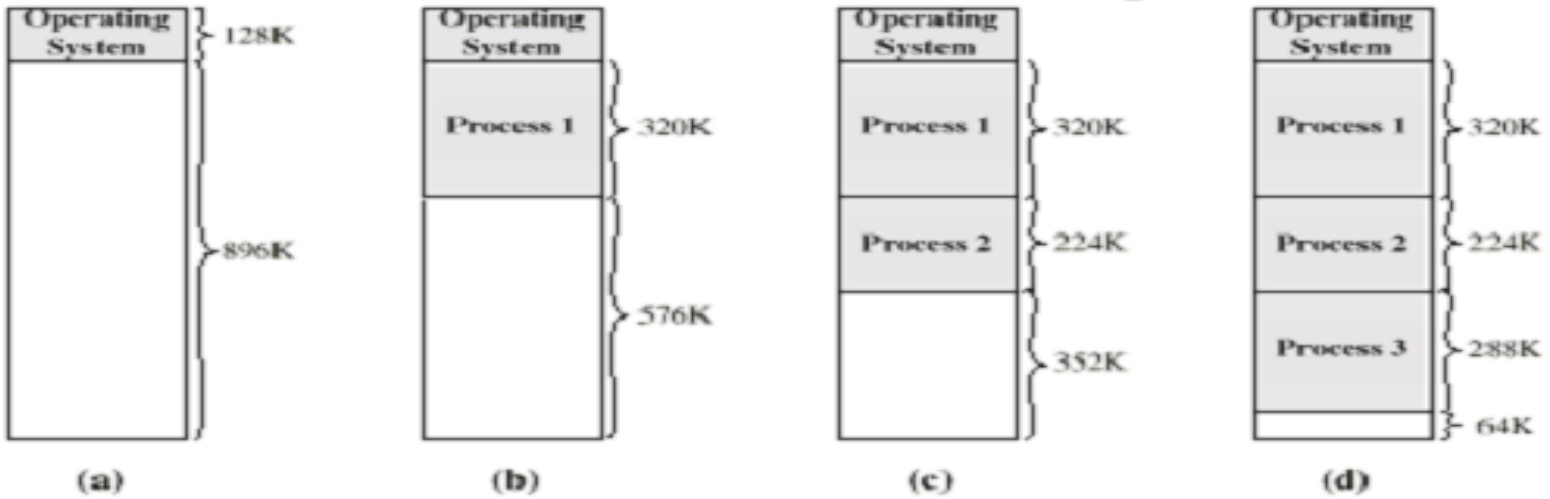
(b) Unequal-size partitions

Manajemen Memory – Partitioning

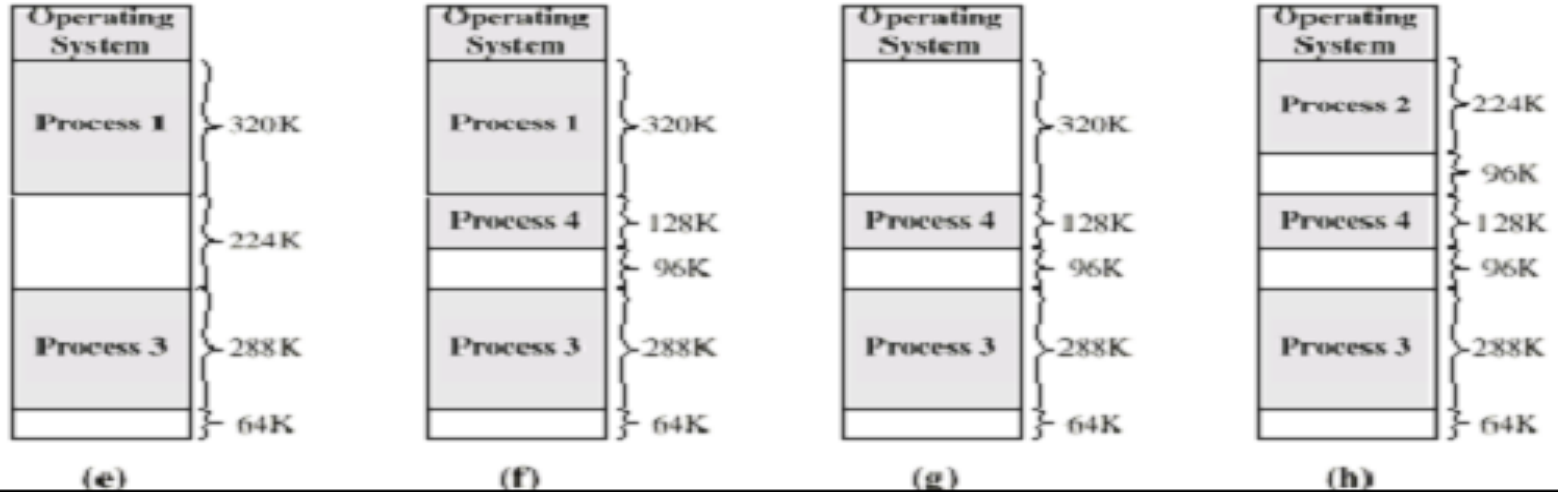


- Variable-sized partitions
  - Proses dimasukkan di tempat kosong yang ukurannya pas sesuai yang dibutuhkan
  - Ada memory yang terbuang juga
  - Akan terjadi banyak fragmen memory yang dipakai dan yang kosong
  - Untuk mendapatkan ruang yang cukup bisa dilakukan pemampatan (defragmentation) dengan mengelompokkan ruang-ruang yang terisi di depan. Tapi ini justru menambah waktu.
  - Muncullah ide untuk paging

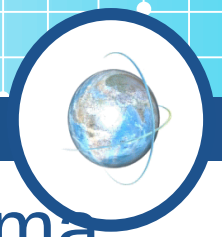
Manajemen Memory – Partitioning



(Ini pun punya kelemahan)



Manajemen Memory – Paging



- Bagi memory menjadi potongan2 kecil sama besar – disebut page frames
- Bagi program (proses) menjadi potongan2 kecil sama besar – disebut pages
- Hitung jumlah page frames yang dibutuhkan proses tersebut, dan masukkan ke dalamnya
- Serangkaian pages tidak harus masuk ke page frames yang contiguous (berlanjut)
- Gunakan page table untuk melacak lanjutan page frame
- Sistem operasi menyimpan daftar page frame yang sedang kosong (bisa digunakan)

Manajemen Memory – Paging

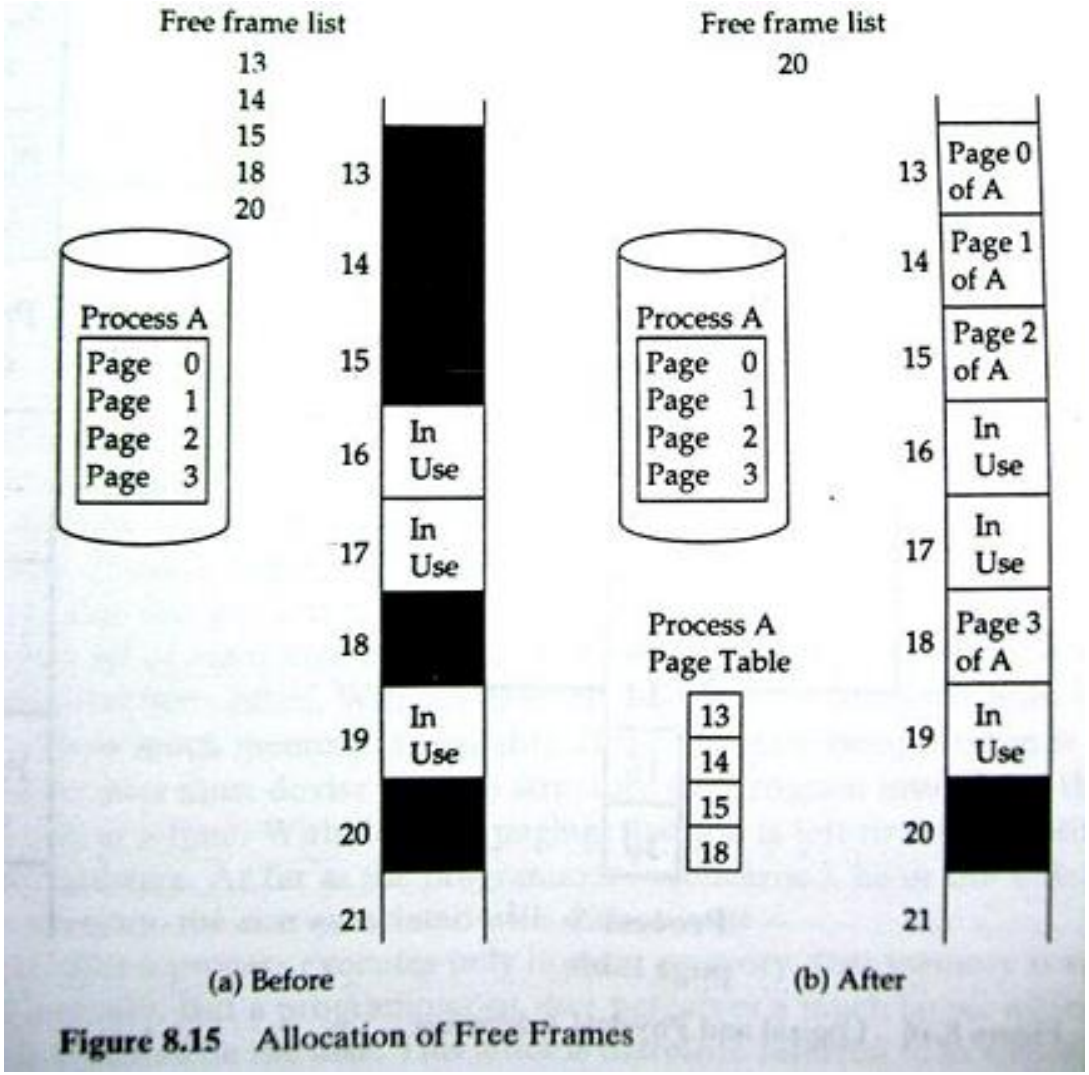
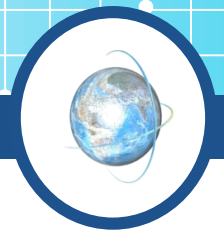


Figure 8.15 Allocation of Free Frames

Manajemen Memory – Paging

- Virtual memory pada Windows dan Linux menerapkan swapping dibantu paging.
- Pada kondisi tertentu, metode virtual memory Windows tidak hanya menggunakan hard disk sebagai tempat swapping tapi juga tempat paging (memory pura2), dan page frames dapat ditambah seperlunya bila dibutuhkan, selama space hard disk masih ada → tetap lambat.
- Ada kalanya sistem operasi mencapai taraf thrashing, yaitu saat memory terlalu kecil untuk dipakai sebuah program, sehingga S/O hanya swapping terus menerus tanpa sempat memproses. Solusinya hanya menambah memory atau mematikan proses lainnya.



selesai