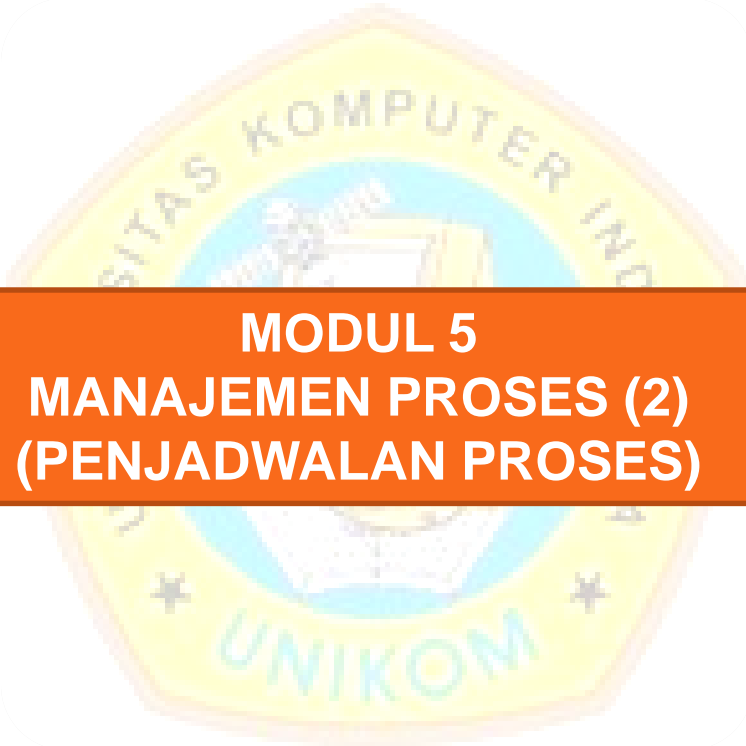




SISTEM OPERASI

Semester 4 Sistem Informasi



MODUL 5 MANAJEMEN PROSES (2) (PENJADWALAN PROSES)

PROSES



Pengelolaan siklus hidup proses :

- **Penciptaan Proses**
- **Penghentian Proses**
- **Pengalihan Proses**

PENCIPTAAN PROSES



Kondisi penyebab penciptaan proses :

- **Request user menjalankan aplikasi melalui shell**
- **Proses lain yang sedang berjalan**

PENCIPTAAN PROSES



Tahap yang dilalui pada penciptaan proses :

1. SO membuat ID proses
2. Membuat rekaman data dan disisipkan pada suatu struktur data di memori utama → proses table
3. Process image disimpan di memori utama, terdiri dari kode program, stack, PCB
4. SO menginisialisasi PCB
5. Menyisipkan PCB ke antrian ready dan merubah status menjadi ready

PENGHENTIAN PROSES



Kondisi penyebab penghentian proses :

- **Proses telah selesai dieksekusi (normal)**
- **Proses dihentikan secara paksa karena terjadi kesalahan**

PENGHENTIAN PROSES



Tahap yang dilalui pada penghentian proses secara normal:

1. Mengembalikan output ke parent process
2. Menghapus rekaman proses di semua antrian dan table proses
3. Menghapus PC dan process image
4. Membebaskan sumber daya yang digunakan

Setelah proses selesai, SO akan melakukan penjadwalan untuk proses selanjutnya

PROCESS SWITCHING



Kondisi umum process switching :

1. Proses yang sedang running sudah habis jatah waktu eksekusi
2. Pergantian proses diseleksi menggunakan rutin scheduler
3. Pengalihan eksekusi menggunakan rutin dispatcher

PROCESS SWITCHING



**Penyebab terjadinya process switching :
Aksi External (Interupsi), Aksi Internal (Trap, System Call)**

- 1. Interupsi (disebabkan oleh aksi eksternal)**
 - **Interupsi waktu (Clock Interrupt) → memeriksa jatah waktu penggunaan prosesor**
 - **Interupsi I/O (I/O Interrupt) → selesainya operasi I/O atau adanya data baru dari peranti I/O**
 - **Interupsi Kesalahan Memori (Memory Fault Interrupt)**

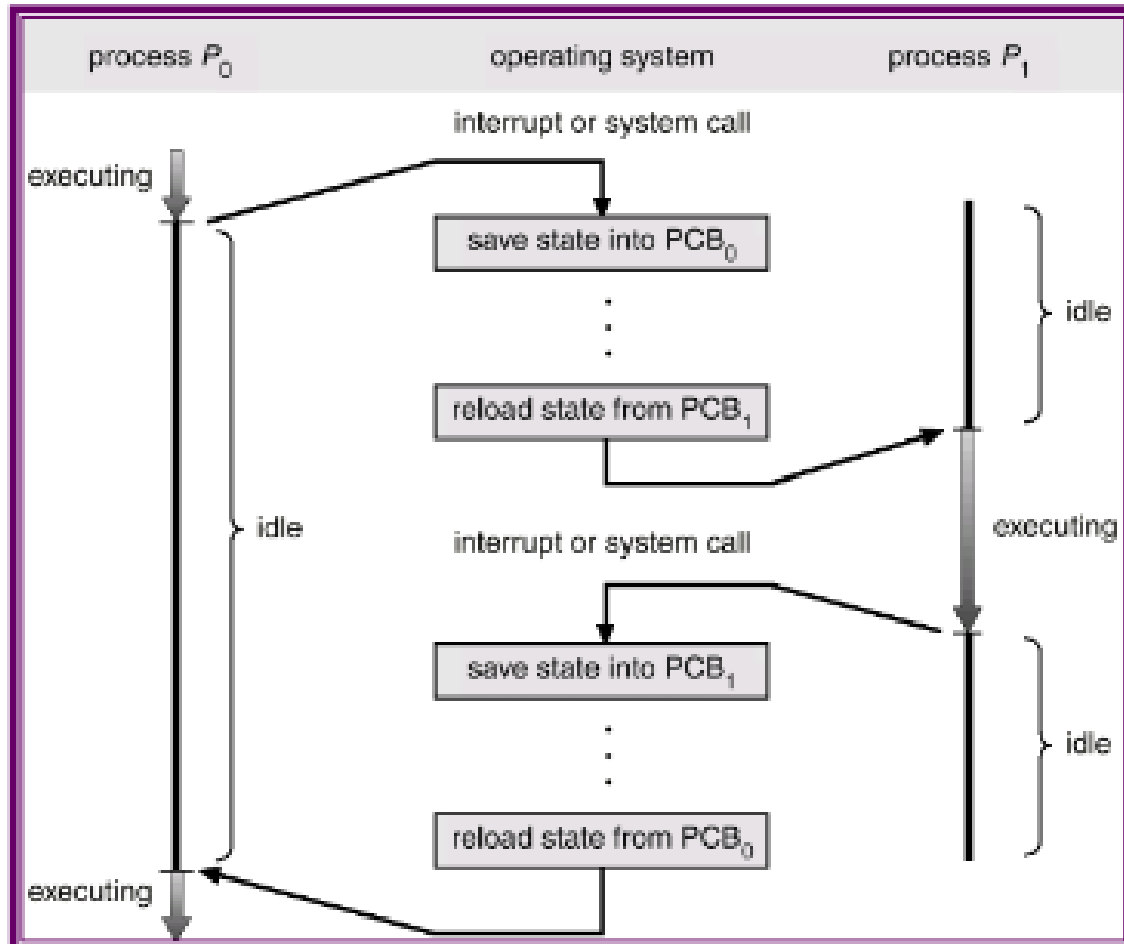
PROCESS SWITCHING



2. Interupsi (disebabkan oleh aksi internal)

- **Trap** → kesalahan eksekusi dari proses yang sedang running. Jika fatal akan mengalihkan ke proses berikutnya, jika tidak fatal akan dilakukan recovery proses dan melanjutkan proses tersebut
- **System Call** → pemanggilan layanan SO oleh proses yang sedang running

PROCESS SWITCHING



PENJADWALAN PROSES



Penjadwalan proses → kumpulan kebijaksanaan dan mekanisme SO yang mengatur urutan dan jangka waktu eksekusi proses-proses yang aktif.

Bertugas memilih proses dan menentukan durasi penggunaan prosesor

PENJADWALAN PROSES



Komponen Penjadwalan :

- Antrian Penjadwalan (Scheduling Queue)
- Penjadwal (Scheduler)
- Dispatcher → **Suatu rutin SO yang berfungsi untuk melakukan pengalihan eksekusi**

PENJADWALAN PROSES



- Antrian Penjadwalan (Scheduling Queue)

Jenis-jenis antrian penjadwalan :

- 1. Job Queue.** Semua proses yang masuk pada suatu sistem akan diletakkan ke dalam job queue
- 2. Ready Queue.** Sedangkan proses-proses yang ada di memori utama dan menunggu untuk dieksekusi diletakkan pada suatu list yang disebut dengan ready queue.
- 3. Device Queue.** Deretan proses yang sedang menunggu peralatan I/O tertentu disebut dengan device queue.

PENJADWALAN PROSES



■ Penjadwal (Scheduler)

Penjadwal → rutin program dengan algoritma tertentu yang menyeleksi proses yang akan dieksekusi prosesor

Jenis scheduler :

- a. Penjadwalan jangka pendek (short term scheduler) → menyeleksi proses-proses mana yang harus diawa ke ready queue
- b. Penjadwalan jangka menengah (medium-term scheduler) → menyeleksi proses yang akan di swap dari dan ke memori utama jika memori utama tidak cukup
- c. Penjadwalan jangka panjang (longterm-scheduler) → memilih proses yang siap dieksekusi, dan mengalokasikan CPU ke salah satu dari proses-proses tersebut

PENJADWALAN PROSES



- **Dispatcher**

Dispatcher → rutin SO yang berfungsi untuk melakukan pengalihan eksekusi dari proses yang running ke proses yang terseleksi oleh short term scheduler

PENJADWALAN PROSES



Kriteria Penjadwalan Proses

- **Keadilan (fairness) → pembagian jatah waktu CPU yang adil, mencegah starvation**
- **Efisiensi (processor utilization) → memastikan CPU digunakan selama masih ada proses berstatus ready**
- **Waktu tanggapan (respon time) → durasi antara user menginput dengan output / feedback dari SO**
- **Waktu tunggu (waiting time) → durasi waktu yang dibutuhkan proses dalam antrian ready**
- **Turn around time → durasi suatu proses dieksekusi dari awal hingga akhir**
- **Throughput → rata-rata proses yang dapat diselesaikan per satuan waktu**

PENJADWALAN PROSES



Strategi Dasar Penjadwalan Proses

- **Non-preemptive**
 - ✓ Proses lain tidak dapat mengambil alih eksekusi prosesor ketika proses lainnya sedang berjalan
 - ✓ Pengalihan terjadi jika proses sebelumnya selesai (normal/abnormal)
 - ✓ Jika proses yang sedang running mengalami crash atau looping tak berhingga, proses lain tidak akan tereksekusi
 - ✓ Terdapat pada sistem batch atau sekuensial

PENJADWALAN PROSES

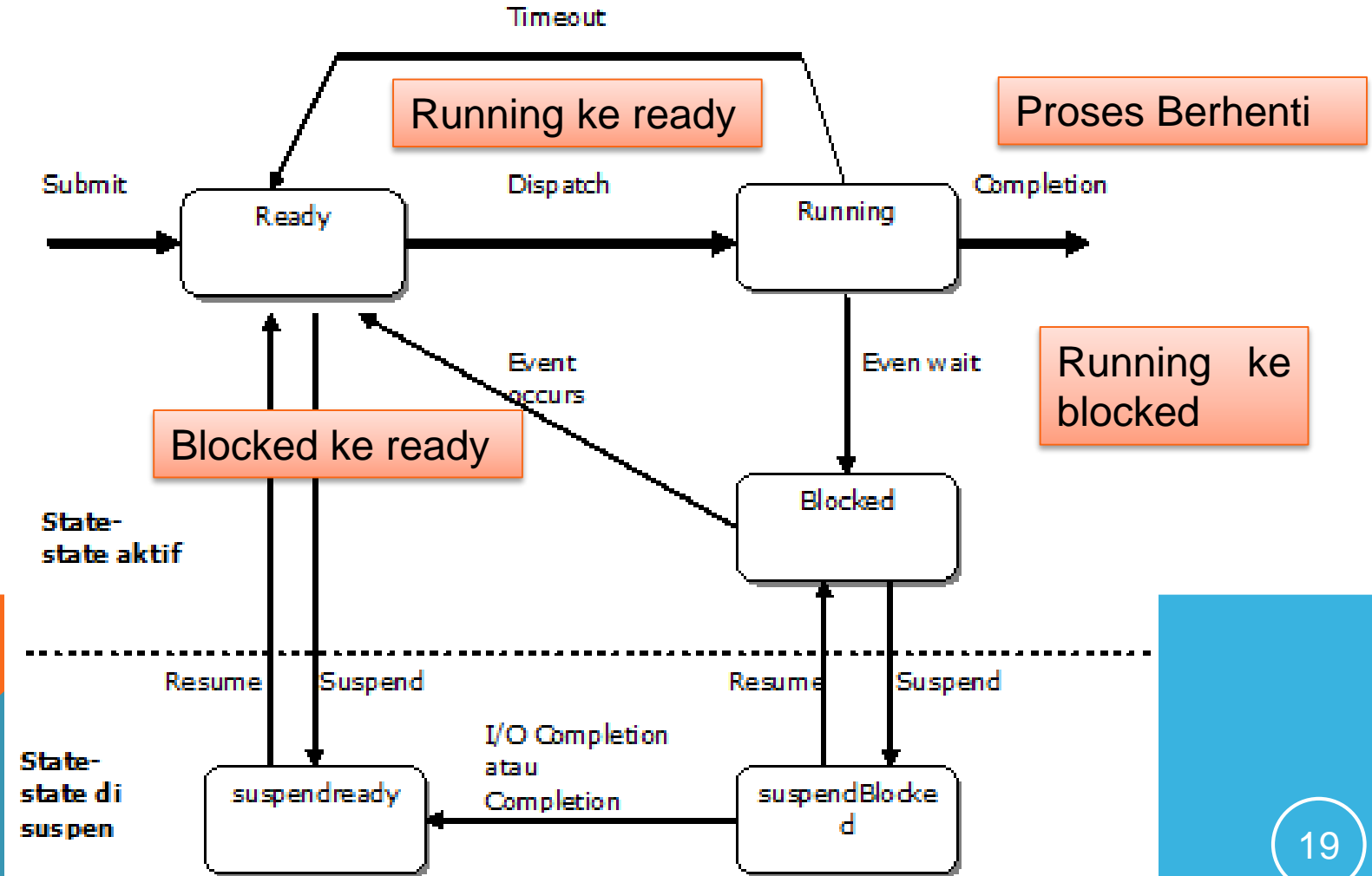


- **Preemptive**
 - ✓ **SO dan Proses lain dapat mengambil alih eksekusi prosesor ketika proses lainnya sedang berjalan**
 - ✓ **Digunakan pada sistem yang membutuhkan tanggapan prosesor secara cepat**
 - ✓ **Terdapat pada sistem konkurensi**

PENJADWALAN PROSES



Pemicu Terjadinya Penjadwalan



SINKRONISASI



Definisi → mekanisme untuk memastikan operasi berjalan secara sinkron pada proses-proses konkuren yang saling mempengaruhi

Penyebab → Race Condition

Akibat → Data tidak konsisten

Critical Resource → Sumber daya yang berada dalam kondisi race condition

Critical Section → kode program yang mengakses critical resource

SINKRONISASI



- **Mutual Exclusion**

Pengertian Mutual Exclusion adalah ketika suatu proses (P0) sedang menggunakan critical section, maka tidak boleh ada proses lain (P1) yang menggunakan critical section di saat bersamaan.

- **Progress**

Artinya ketika tidak ada proses yang menggunakan critical section dan ada proses-proses yang ingin menggunakan critical section tersebut, maka permintaan tersebut harus dipenuhi

- **Bounded Waiting**

Maksud dari Bounded Waiting adalah setiap proses yang menunggu menggunakan critical section, maka proses-proses yang menunggu tersebut dijamin suatu saat akan menggunakan critical section. Dijamin tidak ada yang mengalami starvation.



TERIMA KASIH