

**MATA KULIAH SISTEM OPERASI  
KODE MATA KULIAH CCS210**

**DISUSUN OLEH  
NIZIRWAN ANWAR & TEAM**

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS ESA UNGGUL  
JAKARTA  
2018**

MATERI

**“PROCESS A MANAGEMENT”  
(MANAJEMEN PROSES)**

## 1. Pendahuluan

Proses adalah keadaan ketika sebuah program sedang di eksekusi. Saat komputer berjalan, terdapat banyak proses yang berjalan secara bersamaan. Sebuah proses dibuat melalui *system call create-process* yang membentuk proses turunan (*child process*) yang dilakukan oleh proses induk (*parent process*). Proses turunan tersebut juga mampu membuat proses baru sehingga semua proses ini pada akhirnya membentuk pohon proses.

Ketika sebuah proses dibuat maka proses tersebut dapat memperoleh sumber-daya seperti waktu CPU, memori, berkas, atau perangkat I/O. Sumber daya ini dapat diperoleh langsung dari sistem operasi, dari proses induk yang membagi-bagikan sumber daya kepada setiap proses turunannya, atau proses turunan dan proses induk berbagi sumber-daya yang diberikan sistem operasi.

Dalam proses pengelolaan memori, dengan tujuan, yaitu untuk mengurangi dan memperkecil CPU Idle-Time adalah suatu jangka waktu, dimana CPU tidak bekerja walaupun ada satu/lebih proses yang sedang berjalan.

Hal ini terjadi karena proses I/O memerlukan waktu relatif lebih lama dari pada proses CPU. Dalam konsep memori, dapat kita bagi menjadi 3 bagian, yaitu :

- a. Untuk sistem operasi
- b. Untuk user/pemakai
- c. Tidak dipergunakan (blank)

### Contoh Soal ;

Asumsikan pada tipe mesin IBM 360/65, terdapat suatu program/job, mempunyai 3 proses dimana setiap job memakai 70% waktu, dari waktu tetap dan 30% untuk proses CPU.

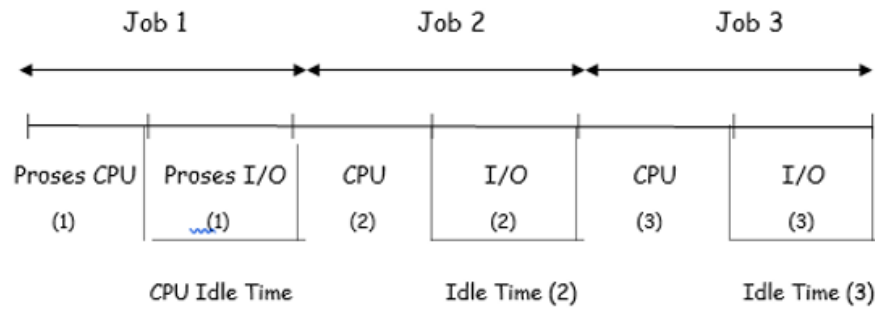
Apabila kapasitas memori mesin tersebut	:	256 Kb.
Sistem operasinya sebesar	:	32 Kb.
Dengan besar job yang berbeda	Job 1	: 15 Kb.
	Job 2	: 30 Kb.
	Job 3	: 50 Kb.

Berapa sisa/blank memori dan gambarkan skema memorinya?

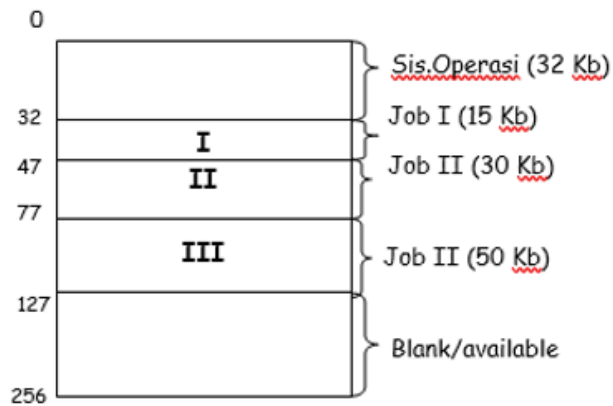
### Jawaban nya ;

secara empiris waktu total

$$\text{Waktu Total} = \sum \text{Waktu CPU} + \sum \text{Waktu I/O}$$



gambarkan skema memorinya adalah



sis/blank memori =  $256 - 127 = 129$  Kb.

Sedangkan untuk proses perhitungan waktu tunggu CPU, rumusnya adalah :

➤ **MONOPROGRAMMING**

$$W = \frac{I_i}{I_i + C_i} \times 100\%$$

dimana

W = Waktu menunggu (untuk Mono)

li = Waktu menunggu I/O

$C_i =$  Waktu total CPU

waktu total ;

Waktu total CPU ( $C_i$  total) =  $\sum$  waktu CPU  $n$  +  $\sum$  waktu I/O  $n$

➤ **MULTIPROGRAMMING**

$$W1 = \frac{\left(\frac{W}{1-W}\right)^n / n! \sum_{i=0}^n \left(\frac{W}{1-W}\right)^i}{i!}$$

dimana

$W1 =$  Waktu menunggu (multiprogramming)

$n =$  Jumlah proses (derajat multiprogramming)

$W =$  Waktu menunggu I/O (monoprogramming)

$i =$  Waktu I/O

**2. Manajemen Proses**

Proses perlu dikelola karena dalam sebuah proses membutuhkan beberapa sumber daya untuk menyelesaikan tugasnya. Sumber daya tersebut dapat berupa CPU time, memori, berkas-berkas, dan perangkat-perangkat I/O.

Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen proses seperti ;

- a. Pembuatan dan penghapusan proses pengguna dan sistem proses.
- b. Menunda atau melanjutkan proses.
- c. Menyediakan mekanisme untuk proses sinkronisasi.
- d. Menyediakan mekanisme untuk proses komunikasi.
- e. Menyediakan mekanisme untuk penanganan deadlock.

### Modul Online 3

Sebagaimana proses bekerja, maka proses tersebut merubah state (keadaan statis/ asal). Status dari sebuah proses didefinisikan dalam bagian oleh aktivitas yang ada dari proses tersebut. Tiap proses mungkin (**Silberschatz**) adalah satu dari keadaan berikut ini:

- a. *New*, Proses sedang dikerjakan/ dibuat.
- b. *Running*, Instruksi sedang dikerjakan.
- c. *Waiting*, Proses sedang menunggu sejumlah kejadian untuk terjadi seperti sebuah penyelesaian I/O
- d. *Ready*, Proses sedang menunggu untuk ditugaskan pada sebuah prosesor.
- e. *Terminated*, Proses telah selsesai melaksanakan tugasnya/ mengeksekusi.

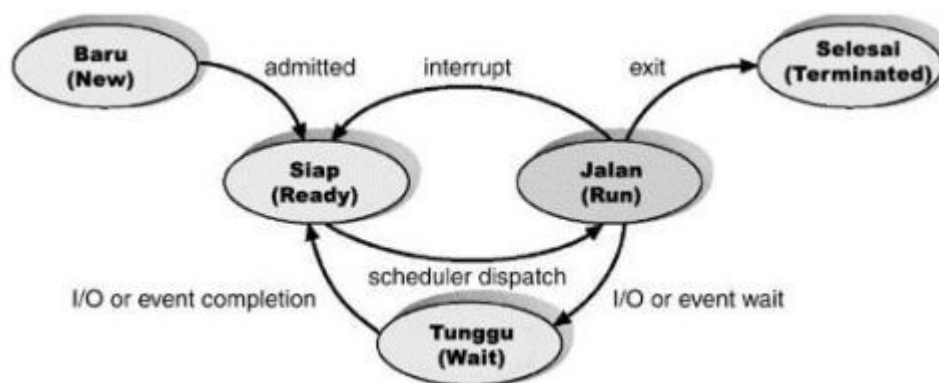
Dan manajemen proses menurut **Tanenbaum**

- a. *Running* : pemroses sedang mengeksekusi proses
- b. *Ready* : proses siap, tapi pemroses belum bersedia mengeksekusi proses ini
- c. *Blocked* : proses menunggu kejadian (event) untuk melengkapi tugasnya.

Contoh: Proses Menunggu

- Selesainya operasi perangkat I/O
- Tersedianya memori
- Tibanya pesan jawaban

dalam bentuk gambar diilustrasikan di bawah ini



Gambar Status Proses Sistem Operasi

Keterangan tambahan dari status manajemen proses system operasi

- Hanya ada satu proses yang dapat berjalan pada prosesor mana pun pada satu waktu!
- Banyak proses yang dapat berstatus ready atau waiting

terdapat 3 (tiga) kemungkinan bila proses memiliki

- **Status Running**
  - ❖ Proses telah selesai → terminated
  - ❖ Waktu yang disediakan habis → interrupt → ready
  - ❖ Sebuah event terjadi (ada permintaan I/O) → proses waiting
- **Status Proses Suspension**
  - ❖ Suspension (penundaan) → memindahkan sementara proses2 tertentu → mereduksi beban sistem selama beban puncak
  - ❖ Suspended process → proses lain meresume
  - ❖ Suspension jangka Panjang → resources proses dibebaskan
  - ❖ Memori utama → dibebaskan → dimanfaatkan proses lain
  - ❖ Resuming proses (pengaktifan kembali) → menjalankan dari titik (instruksi) waktu suspend terjadi.
- **Status Suspend & Resume**

- ❖ Sistem berfungsi buruk/gagal → di-suspend → di-resume
- ❖ Contoh: Proses pencetakan → kertas habis → di-suspend → kertas ditambah → di-resume
- ❖ Pemakai ragu/khawatir hasil proses → suspend proses → berfungsi → resume proses
- ❖ Fluktuasi jangka pendek beban system → suspend proses → normal → resume proses

Hal-hal berkaitan manajemen proses antara lain ;

- a. Multiprogramming
- b. Multiprocessing
- c. Distributed Processing

### 3. Multiprogramming

Multiprogramming adalah suatu metode yang memungkinkan dua buah program atau lebih dijalankan secara serentak dalam sebuah komputer dan berbagai sumber daya dalam waktu yang berlainan. Konsep multiprogramming bisa diibaratkan sebagai pekerja yang tak hanya menangani satu pekerjaan, tetapi juga menangani pekerjaan lain, sehingga tak ada waktu yang terbuang.

Konsep dasar dari multiprogramming ini adalah: suatu proses akan menggunakan CPU sampai proses tersebut dalam status wait (misalnya meminta I/O) atau selesai. Pada saat wait, maka CPU akan nganggur (idle). Untuk mengatasi hal ini, maka CPU dialihkan ke proses lain pada saat suatu proses sedang dalam wait, demikian seterusnya. ([www.info-teknologi.com/konsep-dasar-penjadwalan-cpu/](http://www.info-teknologi.com/konsep-dasar-penjadwalan-cpu/)).

Proses perhitungan waktu tunggu CPU pada multiprogramming, rumusnya adalah sebagai berikut

Pada sistem multiprogramming, sistem operasi harus menyediakan mekanisme untuk manajemen memori, penjadwalan CPU dan manajemen disk. Sistem operasi multiprogram menyediakan supply untuk I/O routine. Sistem harus dapat mengalokasikan memory untuk beberapa job. Beberapa job yang sudah siap dieksekusi akan dipilih oleh system, job mana yang akan dieksekusi oleh CPU. Perangkat apa saja yang diperlukan oleh setiap job juga harus dialokasikan oleh sistem.

Multitasking adalah istilah teknologi informasi dalam bahasa Inggris yang mengacu kepada sebuah metode dimana banyak pekerjaan atau dikenal juga sebagai proses diolah dengan menggunakan sumberdaya CPU yang sama. Dalam kasus sebuah komputer dengan prosesor tunggal, hanya satu instruksi yang dapat bekerja dalam



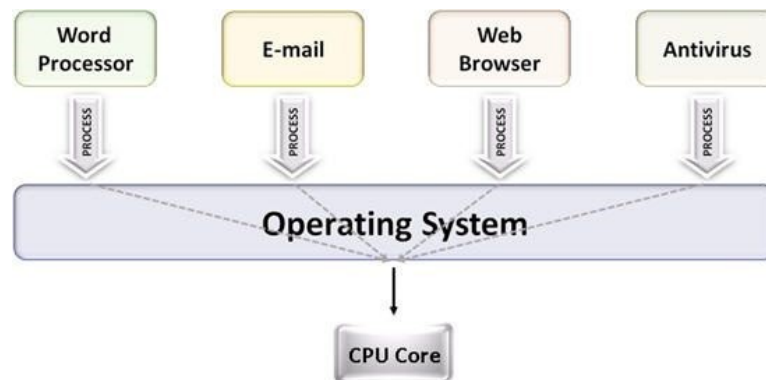
### Modul Online 3

satu waktu, berarti bahwa CPU tersebut secara aktif mengolah instruksi untuk satu pekerjaan tersebut.

Banyak proses yang dijalankan bersamaan, masing-masing proses mendapat bagian memori dan kendali sendiri. Program yang dijalankan bersifat

- a. Tidak bergantung (Independent), Proses terpisah satu dari lainnya & tidak berpengaruh
- b. Satu program pada satu saat (one program at any instant), Pada satu waktu hanya satu proses yang dilayani pemroses, menggunakan interleave bukan overlap diantara program-program

Oleh karena perpindahan dari satu proses ke proses dilakukan secara cepat bagi bagi pemakai seolah-olah bekerja secara paralel. Hal ini dikenal dengan paralel semu (*pseudoparallelism*)



Gambar Multiprogramming

Contoh : PC, workstation dengan MS Windows 3.1, NT, OS/2, MacOS 7 atau sejenisnya

#### 4. Multiprocessing

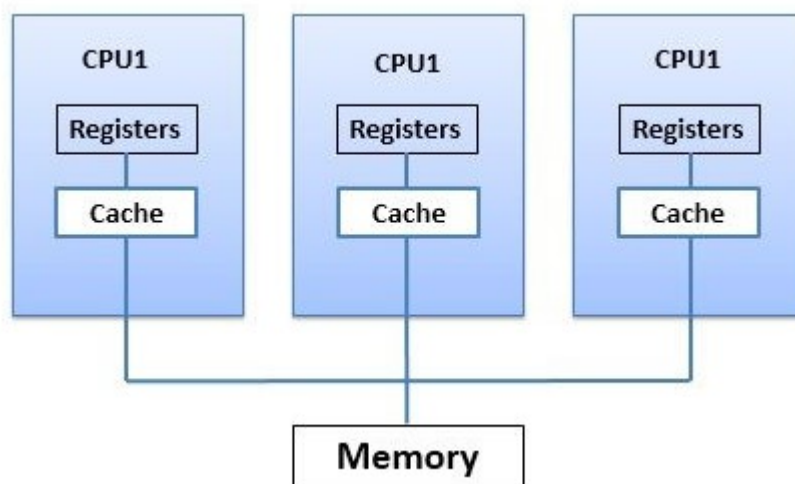
Multiprocessing adalah istilah teknologi informasi dalam bahasa Inggris yang merujuk kepada kemampuan pemrosesan komputer yang dilakukan secara serentak. Hal ini dimungkinkan dengan menggunakan dua CPU atau lebih dalam sebuah sistem komputer. Istilah ini juga dapat merujuk kepada dukungan sebuah sistem untuk mendukung lebih dari satu prosesor dan mengalokasikan tugas kepada prosesor-prosesor tersebut.

Multiprocessing juga kadang merujuk kepada kemampuan eksekusi terhadap beberapa proses perangkat lunak dalam sebuah sistem secara serentak, jika

dibandingkan dengan sebuah proses dalam satu waktu, meski istilah multiprogramming lebih sesuai untuk konsep ini. Multiprocessing sering diimplementasikan dalam perangkat keras (dengan menggunakan beberapa CPU sekaligus), sementara multiprogramming sering digunakan dalam perangkat lunak. Sebuah sistem mungkin dapat memiliki dua kemampuan tersebut, salah satu di antaranya, atau tidak sama sekali.

Multiprocessing dapat dibagi ke dalam beberapa kelas, yakni:

- a. Berdasarkan simetrinya, multiprocessing
  - (1) Asymmetric Multiprocessing (ASMP)
  - (2) Symmetric Multiprocessing (SMP)
  - (3) Non-uniform memory access (NUMA) multiprocessing
  - (4) Clustering
  
- b. Berdasarkan jumlah instruksi dan datanya (lihat Taksonomi Flynn)
  - (1) SISD (Single Instruction on Single Data Stream)
  - (2) SIMD (Single Instruction on Multiple Data Stream)
  - (3) MISD (Multiple Instruction on Single Data Stream)
  - (4) MIMD (Multiple Instruction on Multiple Data Stream)
  
- c. Berdasarkan kedekatan antar prosesor
  - (1) Loosely coupled
  - (2) Tightly coupled



Gambar Multiprocessing

Contoh : *mainframe* dan minikomputer besar (dulu), *server & desktop* (sekarang), perangkat mobile

## 5. Distributed Processing

Mengerjakan semua proses pengolahan data secara bersama antara komputer pusat dengan beberapa komputer yang lebih kecil dan saling dihubungkan melalui jalur komunikasi. Setiap komputer tersebut memiliki prosesor mandiri sehingga mampu mengolah sebagian data secara terpisah, kemudian hasil pengolahan tadi digabungkan menjadi satu penyelesaian total. Jika salah satu prosesor mengalami kegagalan atau masalah yang lain akan mengambil alih tugasnya.

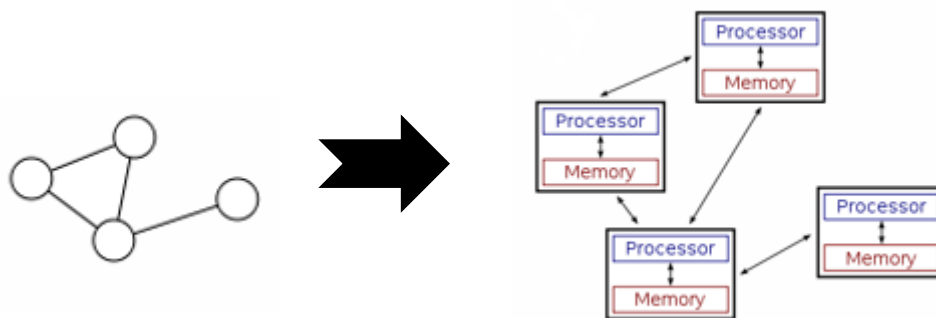
Tujuan dari pengolahan terdistribusi adalah menyatukan kemampuan dari sumber daya (sumber komputasi atau sumber informasi) yang terpisah secara fisik, ke dalam suatu sistem gabungan yang terkoordinasi dengan kapasitas yang jauh melebihi dari kapasitas individual komponen-komponennya.

Tujuan lain yang ingin dicapai dalam komputasi terdistribusi adalah transparansi. Kenyataan bahwa sumber daya yang dipakai oleh pengguna sistem terdistribusi berada pada lokasi fisik yang terpisah, tidak perlu diketahui oleh pengguna tersebut. Transparansi ini memungkinkan pengguna sistem terdistribusi untuk melihat sumber daya yang terpisah tersebut seolah-olah sebagai satu sistem komputer tunggal, seperti yang biasa digunakannya.

Salah satu masalah yang dihadapi dalam usaha menyatukan sumber daya yang terpisah ini antara lain adalah skalabilitas, dapat atau tidaknya sistem tersebut dikembangkan lebih jauh untuk mencakup sumber daya komputasi yang lebih banyak.

Arsitektur umum yang memungkinkan proses sistem terdistribusi antara lain:

- a. Client-Server, client menghubungi server untuk pengambilan data, kemudian server memformatnya dan menampilkannya ke pengguna.
- b. Arsitektur 3-tier: Kebanyakan aplikasi web adalah 3-Tier.
- c. Arsitektur N-tier: N-Tier biasanya menunjuk ke aplikasi web yang menyalurkan lagi permintaan kepada pelayanan enterprise. Aplikasi jenis ini paling berjasa bagi kesuksesan server aplikasi.
- d. Tightly coupled: biasanya menunjuk kepada satu set mesin yang sangat bersatu yang menjalankan proses yang sama secara paralel, membagi tugas dalam bagian-bagian, dan kemudian mengumpulkan kembali dan menyatukannya sebagai hasil akhir.
- e. Peer-to-peer: sebuah arsitektur di mana tidak terdapat mesin khusus yang melayani suatu pelayanan tertentu atau mengatur sumber daya dalam jaringan. Dan semua kewajiban dibagi rata ke seluruh mesin, yang dikenal sebagai peer.
- f. Service oriented di mana sistem diatur sebagai satu set pelayanan yang dapat diberikan melalui antar-muka standar.
- g. Mobile code: berdasarkan prinsip arsitektur mendekatkan pemrosesan ke sumber data
- h. Replicated repository: Di mana repository dibuat replikanya dan disebar ke dalam sistem untuk membantu pemrosesan online/offline dengan syarat keterlambatan pembaharuan data dapat diterima.



Gambar Distributed Processing

## 6. Operasi-Operasi Pada Proses

- Penciptaan proses (create a process)
- Penghancuran/terminasi proses (destroy a process)
- Penundaan proses (suspend a process)
- Pelanjutan kembali proses (resume a process)
- Pengubahan prioritas proses
- Mem-blocked proses
- Membangunkan proses
- Menjadwalkan proses
- Komunikasi dengan proses lain

### Aktivitas yang terjadi ketika penciptaan proses ;

- Menamai (memberi **identitas**) proses
- Menyisipkan proses pada **senarai proses**/tabel proses
- Menentukan **prioritas** awal proses
- Menciptakan **PCB (process control block)**
- Mengalokasikan **resource** awal bagi proses
- Proses baru → struktur data → mengelola & mengalokasikan ruang alamat.

### Kejadian yang dapat menyebabkan penciptaan proses

- Sebagai tanggapan atas pemberian suatu kerja (*job*)
- Seorang pemakai *log on* ke sistem
- SO menciptakan proses untuk memenuhi layanan (satu fungsi pada program pemakai)

- Proses menciptakan anak proses (*child process*)

### Penciptaan Proses

- Proses induk (*parent process*) dapat menciptakan proses anak (*child process*)
- Proses anak dapat menciptakan proses anak-nya.
- Saling berbagi (sharing) sumber daya (resource)
  - ❖ Induk dan anak membagi semua resource.
  - ❖ Anak membagi subset resource milik induk.
  - ❖ Tidak ada resource yg dibagi induk dan anak.
- Eksekusi
  - ❖ Induk dan anak mengeksekusi secara bersamaan.
  - ❖ Induk menunggu hingga anak selesai (terminate).

model pada UNIX dan MS DOS (studi kasus)

- ***Pada UNIX***
  - ❖ System call ***fork*** menciptakan proses baru
  - ❖ Menciptakan salinan identik dengan proses induk.
  - ❖ Induk melanjutkan kerjanya bersama dengan proses anak
- ***Pada DOS***
  - ❖ *Load* berkas biner ke memori -> eksekusi jadi proses anak
  - ❖ Akibatnya menunda proses induk -> proses anak selesai eksekusi

## Penghancuran Proses

- Melibatkan pembebasan dari sistem
- *Resource* yang dipakai dikembalikan ke sistem
- Proses dihancurkan dari senarai/tabel sistem dan PCB ikut dihapus
- Kerumitan muncul bila proses telah menciptakan proses2 lain
  - ❖ Otomatis proses anak hancur → proses induk hancur
  - ❖ Proses anak independen → proses induk

Aktivitas	Proses
Selesaiya proses secara normal	Proses mengeksekusi panggilan layanan So untuk menandakan bahwa proses telah berjalan secara lengkap.
Batas waktu telah terlewati	Proses telah berjalan melebihi batas waktu total yang dispesifikasikan. Terdapat banyak kemungkinan untuk tipe waktu yang diukur, termasuk waktu total yang dijalani ("walk clock time") jumlah waktu yang dipakai untuk eksekusi, dan jumlah waktu sejak pemakai terakhir kali memberi masukan (pada proses interaktif) .
Memori tidak tersedia	Proses memerlukan memori lebih banyak daripada yang dapat disediakan oleh sistem.
Pelanggaran terhadap batas memori	Proses mencoba mengakses lokasi memori yang tidak diijinkan untuk diakses
Terjadi kesalahan karena pelanggaran proteksi	Proses berusaha menggunakan sumber daya atau file yang tidak diijinkan dipakainya, atau proses mencoba menggunakannya tidak untuk peruntukannya, seperti menulis file read only
Terjadi kesalahan aritmatika	Proses mencoba perhitungan terlarang, seperti pembagian dengan nol, atau mencoba menyimpan angka yang lebih besar daripada yang ddaapat diakomodasi oleh H/W
Waktu telah kadaluwarsa	Proses telah menunggu lebih lama daripada maksimum yang telah ditentukan untuk terjadinya suatu kejadian spesfiik
Terjadi kegagalan masukan/keluaran	Kesalahan muncul pada masukan atau keluaran, seprti ketidakmampuan menemukan file, kegagalan membaca atau mennulis setelah sejumlah maksimum percobaan yang ditentukan (misalnya area rusak didapatkan pada tape,atau operasi tidak valid seperti membaca dari line printer)

Intruksi yang tidak benar	Proses berusaha mengeksekusi instruksi yang tidak ada (sering sebagai akibat pencabangan ke daerah data dan berusaha mengeksekusi data tersebut)
Terjadi usaha memakai instruksi yang tidak diijinkan	Proses berusaha mengeksekusi instruksi yang disimpan untuk SO
Kesalahan penggunaan data	Bagian data adalah tipe yang salah atau tidak diinisialisasi
Diintervensi oleh SO atau operator	Untuk suatu alasan, operator atau sistem operasi mengakhiri proses (misalnya terjadi deadlock)
Berakhirnya proses induk	Ketika parent berakhir. So mungkin dirancannng secara otomatis mengakhiri semua anak proses dari parent itu
Atas permintaan proses induk	Parent process biasanya mempunyai otoritas mengakhiri suatu anak proses

## 7. Life-Cycle Manajemen Proses

Proses yang baru diciptakan akan segera mempunyai state ready.

Proses berstate running menjadi blocked karena sumber daya yang diminta belum tersedia atau meminta layanan perangkat masukan/keluaran sehingga menunggu kejadian muncul. Proses menunggu kejadian alokasi sumber daya atau selesainya layanan perangkat masukan/keluaran (event wait).

Proses berstate running menjadi ready karena penjadwal memutuskan eksekusi proses lain karena jatah waktu untuk proses tersebut telah habis (time-out).

Proses berstate blocked menjadi ready saat sumber daya yang diminta/diperlukan telah tersedia atau layanan perangkat masukan/keluaran selesai (event occurs).

Proses berstate ready menjadi running karena penjadwal memutuskan penggunaan pemroses untuk proses itu karena proses yang saat itu running berubah statenya (menjadi ready atau blocked) atau telah menyelesaikan sehingga disingkirkan dari sistem Proses menjadi mendapatkan jatah pemroses.

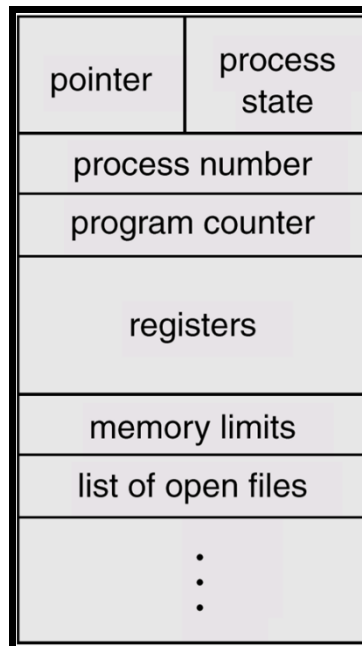
### Citra proses

Memiliki struktur berisi PCB, stack pemakai (user stack), ruang alamat proses eksklusif dan ruang alamat yang dipakai bersama proses lain. Struktur ini ditunjukkan pada tabel sebelumnya. Pada tabel diatas, struktur citra proses digambarkan kontigu (berturutan) di satu ruang alamat. Implementasi penempatan citra proses yang sesungguhnya bergantung skema manajemen memori yang digunakan dan organisasi struktur kendali sistem operasi.

### PCB (Program Control Block)



PCB berisikan banyak bagian dari informasi yang berhubungan dengan sebuah proses yang spesifik, yaitu:

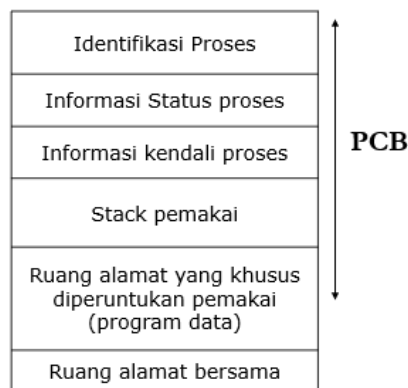


Gambar Blok Level PCB

### Elemen PCB

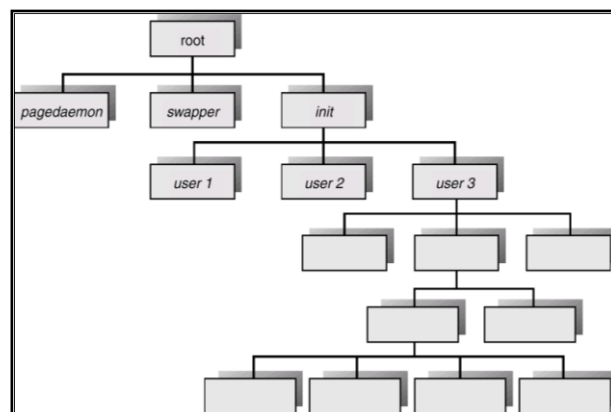
Informasi Kendali Proses
<b>Informasi penjadwakan dan status</b>
Informasi-informasi yang digunakan untuk menjalankan fungsi penjadwalan antara lain : <ul style="list-style-type: none"> <li>▪ Status proses : mendefinisikan keadaan/status proses (running, ready, blocked, dsb)</li> <li>▪ Prioritas : menjelaskan prioritas proses</li> <li>▪ Informasi berkaitan dengan penjadwalan : berkaitan dengan informasi penjadwalan seperti lama menunggu, lama proses terakhir dieksekusi, dsb.</li> <li>▪ Kejadian : identitas kejadian yang ditunggu proses.</li> </ul>
<b>Penstrukturan data</b>
Satu proses dapat dikaitkan dengan proses lain dalam satu antrian atau ring, atau struktur lainnya. PCB harus memiliki pointer untuk mendukung struktur ini.
<b>Komunikasi antar proses</b>
Beragam flag, sinyal dan pesan dapat diasosiasikan dengan komunikasi antara dua proses yang terpisah. Informasi ini disimpan dalam PCB

<b>Kewenangan proses</b>
Proses dapat mempunyai kewenangan berkaitan dengan memori dan tipe instruksi yang dapat dijalankan
<b>Manajemen memori</b>
Bagian ini berisi pointer ke tabel segmen atau page yang menyatakan memori maya (virtual memori) proses.
<b>Kepemilikan dan utilisasi sumber daya</b>
Sumber daya yang dikendalikan proses harus diberi tanda, misalnya : <ul style="list-style-type: none"> <li>▪ Berkas yang dibuka</li> <li>▪ Pemakaian pemroses</li> <li>▪ Pemakaian sumber daya lainnya</li> <li>▪ Informasi ini diperlukan oleh penjadwal</li> </ul>



Gambar Proses Alokasi Memori

## 8. Hirarki Manajemen Proses Sistem UNIX



Gambar Hirarki ProsesUNIX

