

<b>Judul</b>	<b>DESKRIPSI PROSES DAN STRUKTUR DATA PROSES</b>	
<b>Penyusun</b>	<b>Distribusi</b>	<b>Perkuliahan</b>
<b>Nixon Erzed</b>	<b>FASILKOM</b> UNIVERSITAS ESA UNGGUL	Pertemuan – V ON LINE

**Tujuan :**

Mahasiswa memahami pengertian proses sebagai entitas dinamis yang memerlukan layanan sistem komputer dan bagaimana proses diterapkan oleh sistem operasi.

**Materi:**

- Kebutuhan Pengendalian Proses
- Proses Sistem Operasi
- Diagram State Dasar
- Diagram State 5 keadaan
- Program Control Block
- Senarai Proses
- Citra Proses
- Implementasi Proses

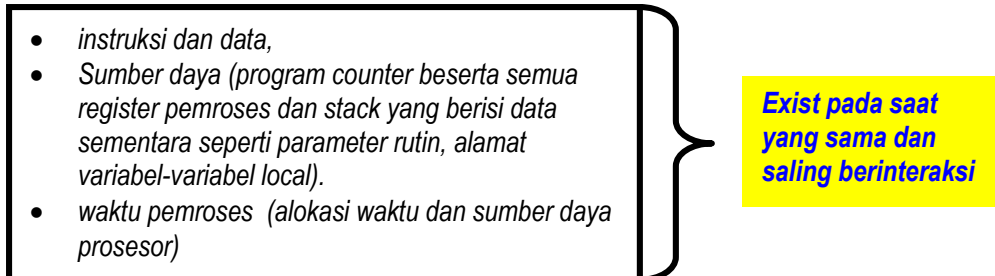
**Referensi :**

1. Modern Operating System 3th Edition Andrew S Tanembaun 2009
2. Operating System, Internals and design Principles, William Stallings 7<sup>th</sup> Ed. 2012
3. Operating System Concepts, Abraham Silberschatz, 9th Ed, 2012
4. Sistem Operasi, Bambang Haryanto, Rev.5 2012
5. Arsitektur dan Organisasi Komputer, William Stalling, Prehalindo

# PROSES

## 1. Pengertian Proses

Proses adalah entitas dinamis. Proses berisi instruksi dan data, program counter beserta semua register pemroses dan stack yang berisi data sementara seperti parameter rutin, alamat variabel-variabel lokal.



## Kebutuhan Utama Pengendalian Proses

Pada konteks multitasking, kebutuhan utama pengendalian proses oleh sistem operasi dapat dinyatakan dengan mengacu ke proses, yaitu:

1. Saling melanjutkan (interleave)
2. Mengikuti kebijaksanaan tertentu
3. Mendukung komunikasi antar proses dan penciptaan proses

### 1. Saling Melanjutkan (Interleave)

Sistem operasi harus interleave (saling melanjutkan) eksekusi proses-proses agar memaksimalkan penggunaan pemroses sambil masih memberi waktu tanggap yang memadai.

**Prosesor melayani banyak proses → jangan sampai prosesor idle sementara terdapat proses yang memerlukan layanan**

Proses harus dikelola → identifikasi ada/tidak ada, proses complete, dan lain-lain.

### 2. Mengikuti Kebijakan Tertentu

Sistem operasi harus mengalokasikan daya ke proses-proses mengikuti kebijakan yang ditentukan sambil menghindari deadlock.

### 3. Mendukung Komunikasi Antar Proses dan Penciptaan Proses

Sistem operasi perlu mendukung komunikasi antar proses dan penciptaan proses oleh proses pemakai lainnya sehingga membantu menstrukturkan aplikasi.

Pada sistem dengan banyak proses aktif, proses-proses dapat berada dalam suatu keadaan saling tergantung

→ misal proses P1 adalah predesesor proses P5, sehingga output dari proses P1 harus dapat disampaikan kepada P5

→ Misal P1 merequest penciptaan P15 → P15 adalah proses anak dari P1

## 2. Diagram State PROSES

Pada sistem dengan banyak proses aktif, proses-proses pada satu saat berada dalam beragam tahap eksekusinya. Proses mengalami beragam state selama siklus hidupnya sebelum berakhir dan keluar dari sistem. Sistem operasi harus dapat mengetahui state masing-masing proses dan merekam semua perubahan yang terjadi secara dinamis. Informasi ini untuk penjadwalan dan memutuskan alokasi sumber daya.

Diagram state proses → menggambarkan keadaan proses-proses pada satu saat selama berada dalam masa aktifnya (submit s/d completion).

Proses mengalami beragam state selama siklus hidupnya sebelum berakhir dan keluar dari sistem (completion).

Sistem operasi harus dapat mengetahui state masing-masing proses dan merekam semua perubahan yang terjadi secara dinamis. Informasi ini untuk penjadwalan dan memutuskan alokasi sumber daya.

Diagram State Dasar (3 state/keadaan) → proses aktif

Diagram State Lanjut (5 state/keadaan) → proses aktif + proses suspend

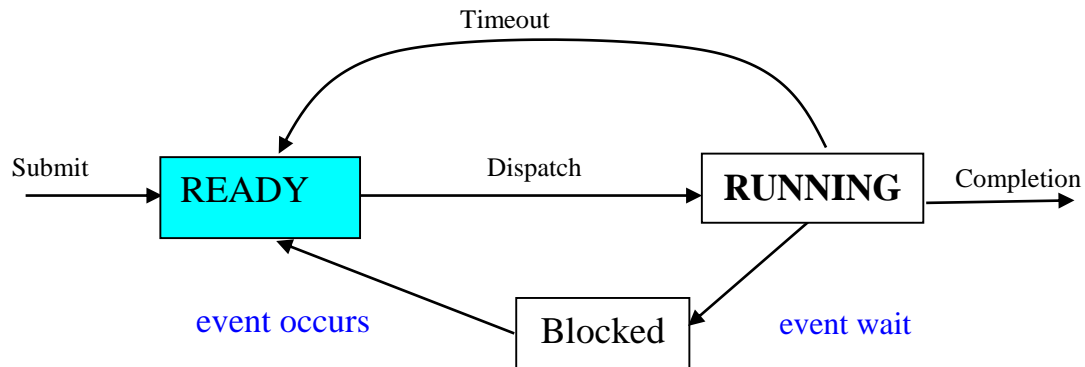
### Diagram State Dasar (Tiga Keadaan Proses Aktif)

Proses melewati serangkaian [state diskrit](#). Beragam kejadian dapat menyebabkan perubahan state proses. Diagram state dasar terdiri dari :

→ 3 keadaan	→ 6 transisi	
<ul style="list-style-type: none"> <li>○ ready,</li> <li>○ running,</li> <li>○ blocked</li> </ul>	<ul style="list-style-type: none"> <li>○ submit</li> <li>○ dispatch</li> <li>○ time out</li> </ul>	<ul style="list-style-type: none"> <li>○ event wait</li> <li>○ event occurs</li> <li>○ completion</li> </ul>

Status	Deskripsi
<b>Running</b>	Pemroses sedang mengeksekusi instruksi proses itu
<b>Ready</b>	Proses siap (ready) dieksekusi, tapi waktu pemroses belum tersedia untuk eksekusi proses ini → antri
<b>Blocked</b>	Proses menunggu kejadian untuk melengkapi tugasnya contoh: <ul style="list-style-type: none"> <li>• Selesaiya operasi perangkat masukan/keluaran Contoh : Statement sortir tabel data → operasi pembacaan tabel ditangani oleh DMAC → sementara menunggu job sortir akan blocked</li> <li>• Tersedianya memori Pada eksekusi variabel dinamis → perlu tambah ruang memory → ketika memory belum tersedia → proses akan diblock</li> <li>• Tibanya pesan jawaban</li> <li>• Dsb</li> </ul>

Hubungan ketiga state dasar digambarkan dengan diagram state.



**Submit :**

Proses yang baru diciptakan akan segera mempunyai state ready.

**Completion**

Ketika proses Running telah menyelesaikan eksekusi terhadap intruksi yang terakhir, maka proses tersebut akan dihancurkan

**Dispatch :**

Proses berstate ready menjadi running karena penjadwal memutuskan penggunaan pemroses untuk proses itu karena :

Prosesor menjadi kosong → proses yang saat itu running berubah state-nya (menjadi ready atau blocked) atau telah menyelesaikan sehingga disingkirkan dari sistem. Proses yang memenuhi aturan/kebijakan penjadwalan, sehingga diputuskan mendapatkan jatah waktu pemroses .

**Event wait :**

Proses berstate running menjadi blocked karena :

- sumber daya yang diminta belum tersedia, atau
- meminta layanan perangkat masukan/keluaran sehingga menunggu kejadian muncul, atau
- menunggu selesainya layanan perangkat masukan/keluaran
- 

**Time Out :**

Proses berstate running menjadi ready karena penjadwalan memutuskan eksekusi proses lain **karena jatah waktu untuk proses tersebut telah habis** → pada implementasi time-sharing

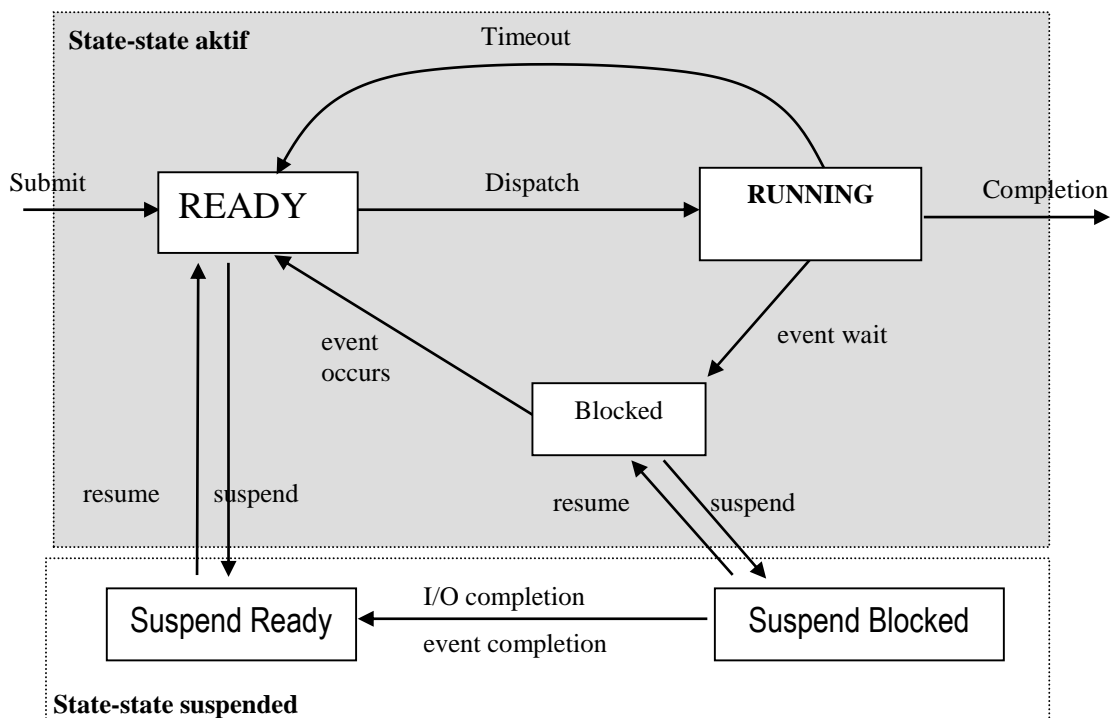
**Event Occurs :**

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

## Diagram State 5 Keadaan

Terdapat kebutuhan *suspended (me-nonaktifkan)* proses-proses aktif untuk mengurangi beban system pada periode sibuk. *Suspended* proses dimaksudkan untuk memperbaiki kinerja system.

- Untuk mengendalikan beban sistem dan menghindarkan penguasaan sumber daya oleh proses-proses yang mungkin tidak dapat direalisasikan dalam waktu tertentu, sistem operasi dapat menonaktifkan (*suspend*) proses-proses tersebut.
- Proses-proses yang non aktif tidak akan berlanjut sampai ada proses lain yang mengaktifkannya (*resume*).



Proses *suspended* → akan di-swap dari ruang memory ke disk (area perluasan memory)

Pendekatan penentuan proses-proses yang akan di-swap (*suspended*) :

1. Proses-proses *blocked* yang kecil kemungkinannya untuk mendapatkan event/I/O yang ditunggu (relative terhadap lama menunggu/*blocked*)
2. proses-proses yang berdasarkan logika penjadwalan, belum akan *Running* pada waktu dekat
3. Proses dengan sumber daya besar dan prioritas bukan utama

*Suspend* dan *resume* merupakan intervensi oleh proses dengan kewenangan lebih tinggi atau oleh administrator.

Proses-proses yang di-suspen akan dipindahkan ke media penyimpanan sekunder, sehingga ruang-ruang alamat yang dikuasainya dapat dialokasikan untuk proses lain.

*Suspended* dilakukan ketika pada beban puncak, admin ingin memperbaiki kinerja sistem

- beberapa proses yang dipilih akan *suspended* dan dipindahkan ke ruang memory sekunder (*extended area*).

## STRUKTUR DATA PROSES

Untuk mendukung implementasi multitasking, struktur data proses terdiri dari PCB dan tabel-tabel proses.

### 1. PCB (Program Control Block)

Tabel identitas proses (berisi semua informasi tentang proses yang diperlukan oleh sistem operasi untuk mengelola proses)

- setiap proses memiliki PCB masing-masing
  - Ketika proses diciptakan → PCB dibuat
  - Ketika proses dihancurkan → PCB dihapus

Informasi di PCB dapat dikelompokkan menjadi tiga kelompok, yaitu:

1. Informasi identifikasi proses.
2. Informasi status pemroses.
3. Informasi kendali proses.

#### **Informasi Identifikasi Proses**

Identifikasi proses berkaitan dengan identitas proses yang unik. Dengan identifier ini, proses dikaitkan ke tabel-tabel lain.

Id yang disimpan pada PCB → id proses, id proses induk, id proses anak

Identifikasi Proses
Identifier
Identifier numerik yang meliputi: <ul style="list-style-type: none"><li>• Identifier proses</li><li>• Identifier proses yang menciptakan</li><li>• Identifier pemakai</li></ul>

#### **Informasi Status Pemroses**

- Informasi ini esensinya terdiri dari isi register-register pemroses. Saat proses berstatus running, informasi-informasi ini berada di register-register.
- Ketika proses diinterupsi, semua informasi register harus disimpan agar dapat dikembalikan saat proses dieksekusi kembali.
- Jumlah dan ragam register yang terlibat bergantung arsitektur komputer.

Informasi Status Pemroses
Register-register yang terlihat pemakai  Register-register yang dapat ditunjuk instruksi bahasa assembly untuk diproses pemroses

<p>Register-register kendali dan status</p> <p>Register-register yang digunakan untuk mengendalikan operasi pemroses, antara lain:</p> <ul style="list-style-type: none"> <li>• Program counter</li> <li>• PSW</li> <li>• dan sebagainya</li> </ul>
<p><b>Pointer stack</b></p> <p>Tiap proses mempunyai satu stack atau lebih. Stack digunakan untuk parameter atau alamat prosedur pemanggil dan system call. Pointer stack menunjuk posisi paling atas dari stack.</p>

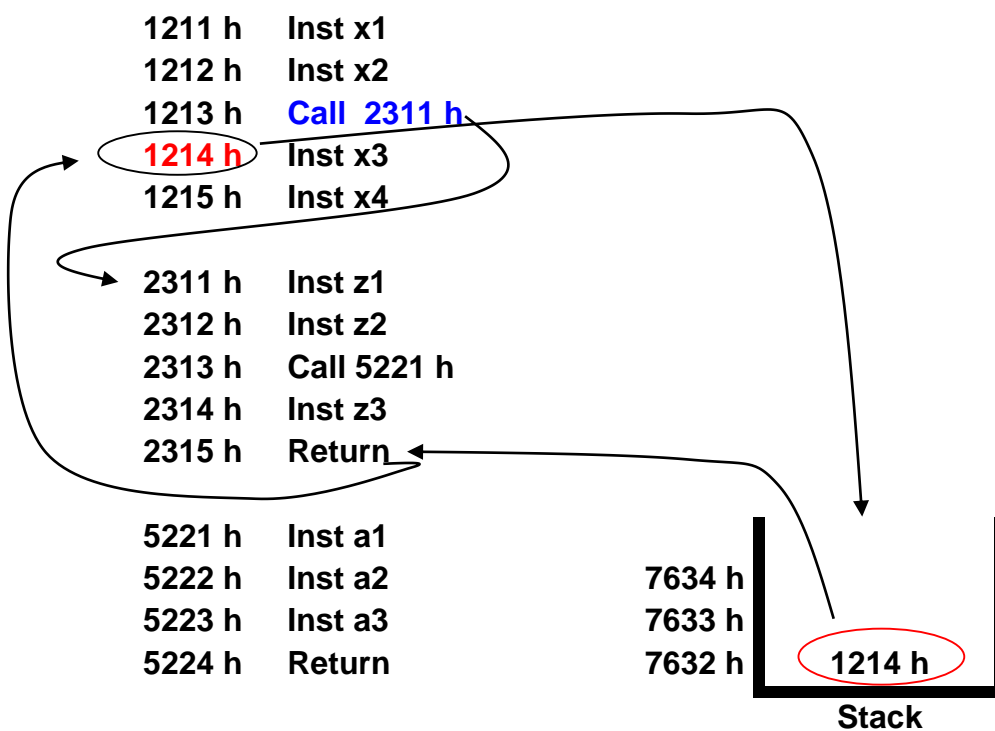
**Ilustrasi tentang bekerjanya Stack Pointer → register alamat**

	Acc
B	C
D	E
H	L
Stack Pointer	
Program Counter	

PC berisi alamat instruksi berikut yang akan dieksekusi. Ketika pertama kali program dieksekusi, alamat instruksi yang pertama akan diambil dari PCB dan load ke Register PC.

- Ketika instr. Call 2311 h dikerjakan, maka PC = 1214 h
- Eksekusi call 2311 h : menyebabkan PC harus replace dengan alamat baru (2311 H) → agar bisa return maka 1214 h harus di backup → push ke stack

Stack Pointer adalah register alamat yang ada pada prosesor dan **ruang Stack → ruang memory**



## Informasi Kendali Proses

Informasi kendali proses adalah informasi-informasi lain yang diperlukan sistem operasi untuk mengendalikan dan koordinasi beragam proses aktif.

Informasi Kendali Proses
<p>Informasi penjadwalan dan status Informasi-informasi yang digunakan untuk menjalankan fungsi penjadwalan antara lain:</p> <ul style="list-style-type: none"><li>• State 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>
<p>Penstrukturan data Satu proses dapat dikaitkan dengan proses lain dalam satu antrian atau ring, atau struktur lainnya. PCB harus memiliki pointer untuk mendukung struktur ini.</p>
<p>Komunikasi antar proses Beragam flag, sinyal dan pesan dapat diasosiasikan dengan komunikasi antara dua proses yang terpisah. Informasi ini disimpan dalam PCB.</p>
<p>Kewenangan proses Proses dapat mempunyai kewenangan berkaitan dengan memori dan tipe instruksi yang dapat dijalankan.</p>
<p>Manajemen memori Bagian ini berisi pointer ke tabel segmen atau page yang menyatakan memori maya (virtual memory) proses.</p>
<p>Kepemilikan dan utilisasi sumber daya Sumber daya yang dikendalikan proses harus diberi tanda, misalnya:</p> <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>



## 2. SENARAI PROSES

Dalam implementasi multitasking

→ proses-proses akan diorganisasikan dengan cara merangkai PCB dalam suatu senarai (linked list) → PCB adalah simpul dari senarai

→ misal dimiliki proses  $P_A$   $P_B$   $P_C$   $P_D$   $P_E$

Dengan keadaan sebagai berikut

- $P_E$  berada dalam state running
- $P_A$  dan  $P_C$  berada dalam keadaan Blocked
- $P_B$ ,  $P_D$ ,  $P_F$  dan  $P_G$  berada dalam keadaan Ready dengan urutan kedatangan sesuai urutan indeks

Ilustrasi Penempatan Proses dan Penciptaan PCB

	Alamat fisik (block memory)	Peta memory	
		Id proses	Pointer
Area Memory untuk PCB	00FFA H	$P_G$	NIL
	00DE1 H	$P_F$	00FFA H
	00AAA H	$P_E$	NIL
	00800 H	$P_D$	00DE1 H
	003A1 H	$P_C$	NIL
	00310 H	$P_B$	00800 H
	001B0 H	$P_A$	003A1 H
pointer Running	00021 H	00AAA H	
pointer Ready	00020 H	00310 H	
pointer Blocked	0001F H	001B0 H	

### 3. TABEL-TABEL PROSES

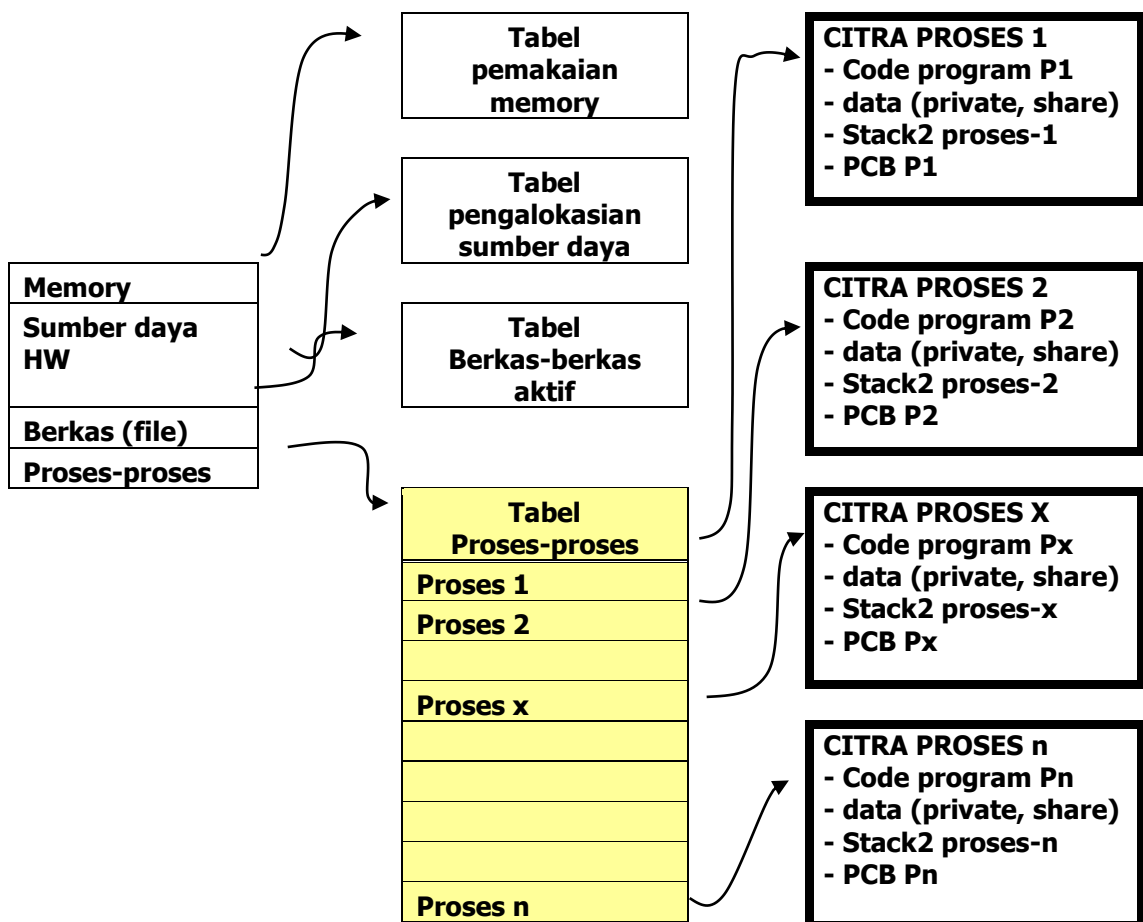
Tabel-tabel yang dimiliki sistem operasi diperlukan untuk mengelola proses-proses

Antara lain:

1. Tabel Antrian Ready
2. Tabel proses Blocked
3. Tabel Pencatatan Memory
4. Tabel alokasi sumber daya
5. dll

Struktur → tergantung karakteristik data yang disimpan, dan tujuan tabel

#### STRUKTUR TABEL-TABEL PROSES

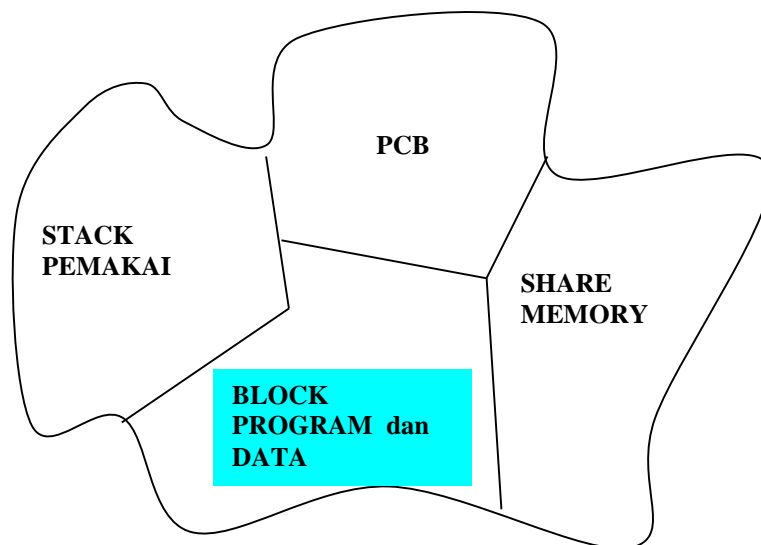


## 4. CITRA PROSES

Total ruang memory yang dialokasikan untuk melayani proses

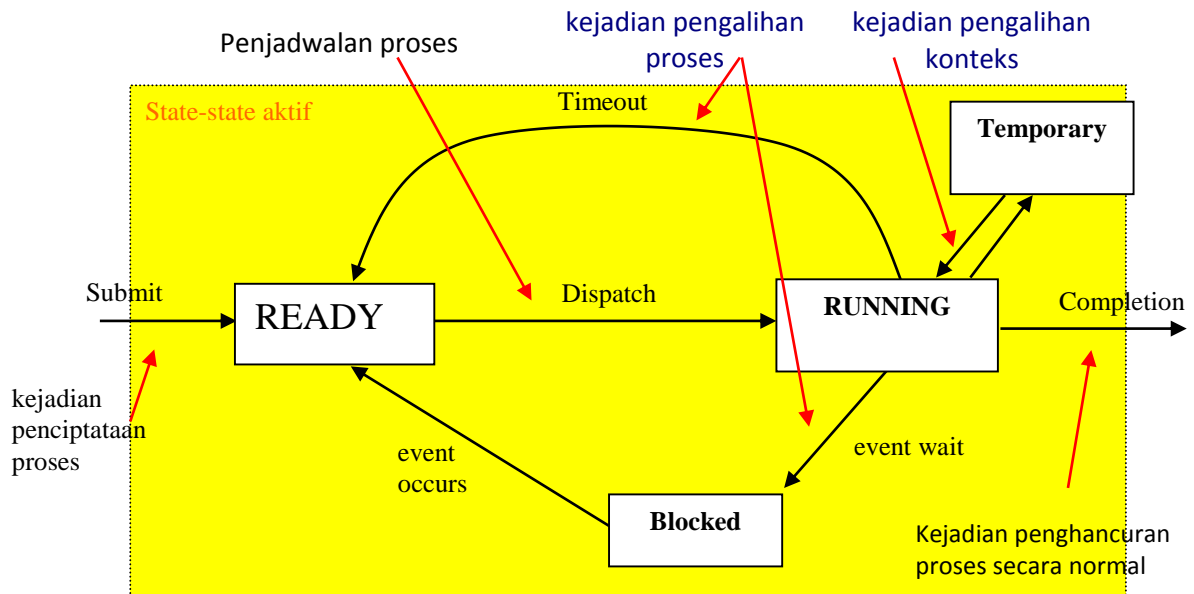
Citra proses mempunyai struktur berisi PCB, stack pemakai (user stack), ruang alamat proses eksklusif dan ruang alamat yang dipakai bersama proses lain. Struktur ini ditunjukkan gambar.

Pada gambar, 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.



## OPERASI-OPERASI PADA PROSES

Sistem operasi dalam mengelola proses, melakukan operasi-operasi terhadap proses.



Sehubungan dengan state-state proses yang tergambar pada diagram state dasar, operasi-operasi terhadap proses diantaranya adalah :

- Penciptaan proses (create a process)  
Memberikan tanggapan terhadap request pengekseskuan suatu program/proses.
- Penghancuran/terminasi proses (destroy a process)  
Menghancurkan suatu proses yang telah mengalami completion, sehingga dapat membebaskan seluruh sumber daya sistem yang teralokasi pada proses tersebut.
- Pengalihan Proses  
Mengalihkan penguasaan prosesor karena proses yang terakhir running meninggalkan prosesor dan proses dari antrian ready yang diputuskan penjadwal untuk running segera memasuki prosesor .
- Menjadwalkan (dispatcher) proses,  
Penjadwalan dapat diartikan sebagai cara menentukan/memilih satu proses dari sekumpulan proses berstate ready untuk running ketika proses terakhir yang running melepaskan penguasaannya terhadap prosesor.

Dalam pengertian yang lebih luas penjadwalan meliputi :

- Penundaan proses (suspend a process)  
Me-non aktifkan sementara suatu proses untuk memperbaiki kinerja sistem. Proses non aktif akan di-swap out ke *extended area*.
- Pelanjutan kembali proses (resume a process)  
Memulihkan kembali proses yang non aktif. Resume sebuah proses dilakukan dengan cara men-swap in dari *extended area* kembali ke memory
- Pengubahan prioritas proses
- Mem-block proses