

# Proses

By : Jajang Kusnendar  
email : [jkusnendar@gmail.com](mailto:jkusnendar@gmail.com)

# Definisi

- | Secara informal; proses adalah program dalam eksekusi. Suatu proses adalah lebih dari kode program
- | Proses juga termasuk aktivitas yang sedang terjadi, sebagaimana digambarkan oleh nilai pada program counter dan isi dari daftar prosesor/processor's register. Suatu proses umumnya juga termasuk process stack, yang berisikan data temporer (seperti parameter metoda, address yang kembali, dan variabel lokal) dan sebuah data section, yang berisikan variabel global.

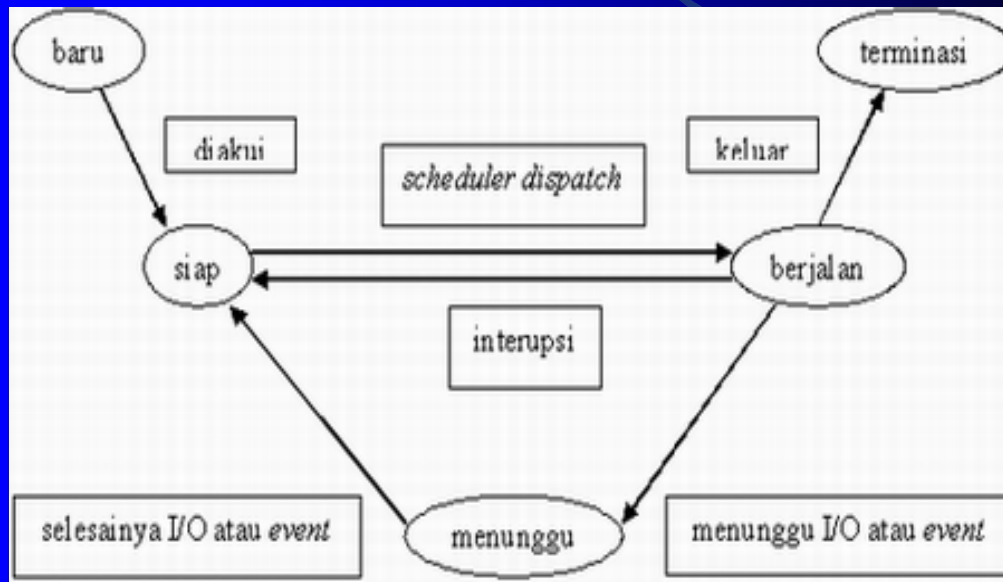
# Keadaan Proses

- | 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 adalah satu dari keadaan berikut ini:

# Keadaan Proses

- | *New*: Proses sedang dikerjakan/ dibuat.
- | *Running*: Instruksi sedang dikerjakan.
- | *Waiting*: Proses sedang menunggu sejumlah kejadian untuk terjadi (seperti sebuah penyelesaian I/O atau penerimaan sebuah tanda/ signal).
- | *Ready*: Proses sedang menunggu untuk ditugaskan pada sebuah prosesor.
- | *Terminated*: Proses telah selsesai melaksanakan tugasnya/ mengeksekusi.

# Keadaan Proses



# Proses Control Block

- I Tiap proses digambarkan dalam sistem operasi oleh sebuah *process control block* (PCB)
- I PCB berisikan banyak bagian dari informasi yang berhubungan dengan sebuah proses yang spesifik, termasuk ini:

# Keadaan Proses

- | Keadaan proses: Keadaan mungkin, *new*, *ready*, *running*, *waiting*, *halted*, dan juga banyak lagi.
- | *Program counter*: *Counter* mengindikasikan address dari perintah selanjutnya untuk dijalankan untuk proses ini.
- | CPU register: Register bervariasi dalam jumlah dan jenis, tergantung pada rancangan komputer. Register tersebut termasuk accumulator, index register, stack pointer, general-puposes register, ditambah code information pada kondisi apa pun. Besertaan dengan program counter, keadaan/ status informasi harus disimpan ketika gangguan terjadi, untuk memungkinkan proses tersebut berjalan/ bekerja dengan benar setelahnya

# Keadaan Proses

- | Informasi manajemen memori: Informasi ini dapat termasuk suatu informasi sebagai nilai dari dasar dan batas register, tabel page/ halaman, atau tabel segmen tergantung pada sistem memori yang digunakan oleh sistem operasi
- | Informasi pencatatan: Informasi ini termasuk jumlah dari CPU dan waktu riil yang digunakan, batas waktu, jumlah akun, jumlah job atau proses, dan banyak lagi.

# Keadaan Proses

- | Informasi status I/O: Informasi termasuk daftar dari perangkat I/O yang di gunakan pada proses ini, suatu daftar open berkas dan banyak lagi.
- | PCB hanya berfungsi sebagai tempat menyimpan/ gudang untuk informasi apa pun yang dapat bervariasi dari prose ke proses.

# Keadaan Proses

<i>pointer</i>	<i>state proses</i>
	nomor proses
	<i>program counter</i>
	<i>registers</i>
	batas memori
	daftar berkas yang telah dibuka
	.....

# Threads

- | Model proses yang didiskusikan sejauh ini telah menunjukkan bahwa suatu proses adalah sebuah program yang menjalankan eksekusi thread tunggal.
- | Sebagai contoh, jika sebuah proses menjalankan sebuah program Word Processor, ada sebuah thread tunggal dari instruksi-instruksi yang sedang dilaksanakan.

# Threads

- | Kontrol thread tunggal ini hanya memungkinkan proses untuk menjalankan satu tugas pada satu waktu.
- | Banyak sistem operasi modern telah memiliki konsep yang dikembangkan agar memungkinkan sebuah proses untuk memiliki eksekusi *multithreads*, agar dapat secara terus menerus mengetik dalam karakter dan menjalankan pengecek ejaan didalam proses yang sama. Maka sistem operasi tersebut memungkinkan proses untuk menjalankan lebih dari satu tugas pada satu waktu.

# Penjadwalan Proses

- | Tujuan dari multiprogramming adalah untuk memiliki sejumlah proses yang berjalan pada sepanjang waktu, untuk memaksimalkan penggunaan CPU.
- | Tujuan dari pembagian waktu adalah untuk mengganti CPU diantara proses-proses yang begitu sering sehingga pengguna dapat berinteraksi dengan setiap program sambil CPU bekerja.
- | Untuk sistem uniprosesor, tidak akan ada lebih dari satu proses berjalan. Jika ada proses yang lebih dari itu, yang lainnya akan harus menunggu sampai CPU bebas dan dapat dijadualkan kembali.

# Penjadwalan Proses

- | Ketika proses memasuki sistem, mereka diletakkan dalam antrian job. Antrian ini terdiri dari seluruh proses dalam sistem.
- | Proses yang hidup pada memori utama dan siap dan menunggu/ wait untuk mengeksekusi disimpan pada sebuah daftar bernama ready queue. Antrian ini biasanya disimpan sebagai daftar penghubung. Sebuah header ready queue berisikan penunjuk kepada PCB-PCB awal dan akhir.
- | Setiap PCB memiliki pointer field yang menunjukkan proses selanjutnya dalam ready queue.

# Penjadwalan Proses

- | Begitu proses tersebut mengalokasikan CPU dan menjalankan/ mengeksekusi, satu dari beberapa kejadian dapat terjadi.
- Proses tersebut dapat mengeluarkan sebuah permintaan I/O, lalu di tempatkan dalam sebuah antrian I/O.
- Proses tersebut dapat membuat subproses yang baru dan menunggu terminasinya sendiri.
- Proses tersebut dapat digantikan secara paksa dari CPU, sebagai hasil dari suatu interupsi, dan diletakkan kembali dalam ready queue.

# Kriteria Penjadwalan

- | Adil (fairness)
- | Efisiensi
- | Waktu Tanggap (respon time)
- | Turn Around Time= waktu yang dihabiskan dari saat program atau job mulai masuk ke sistem sampai proses diselesaikan sistem.
- | Throughput= jumlah kerja yang dihasilkan dalam satu unit waktu



# Strategi Penjadwalan

- | Non preemptive = tidak dapat disela
- | Preemptive = dapat disela

# Algoritma-algoritma

## Non Preemptive

- | FIFO (First-in, First Out)
- | SJF (Shortest Job First)
- | HRN (Highest Ratio Next)
- | MFQ (Multiple Feedback Queues)

## Preemptive

- | RR (Round Robin)
- | SRF (Shortest Remaining First)
- | PS (Priority Scheduling)
- | GS (Guaranteed Scheduling)

# Penjadual

- | Sebuah proses berpindah antara berbagai penjadualan antrian selama umur hidupnya. Sistem operasi harus memilih, untuk keperluan penjadualan, memproses antrian-antrian ini dalam cara tertentu.
- | Pemilihan proses dilaksanakan oleh penjadual yang tepat/cocok.
- | Dalam sistem batch, sering ada lebih banyak proses yang diserahkan daripada yang dapat dilaksanakan segera. Proses ini dipitakan/ disimpan pada suatu alat penyimpan masal (biasanya disket), dimana proses tersebut disimpan untuk eksekusi dilain waktu. Penjadualan long term, atau penjadual job, memilih proses dari pool ini dan mengisinya kedalam memori eksekusi.

# Alih Konteks (Context Switch)

- | Mengganti CPU ke proses lain memerlukan penyimpanan suatu keadaan proses lama (*state of old process*) dan kemudian beralih ke proses yang baru. Tugas tersebut diketahui sebagai alih konteks(*context switch*).
- | Alih konteks sebuah proses digambarkan dalam PCB suatu proses; termasuk nilai dari
- | CPU register, status proses dan informasi manajemen memori.

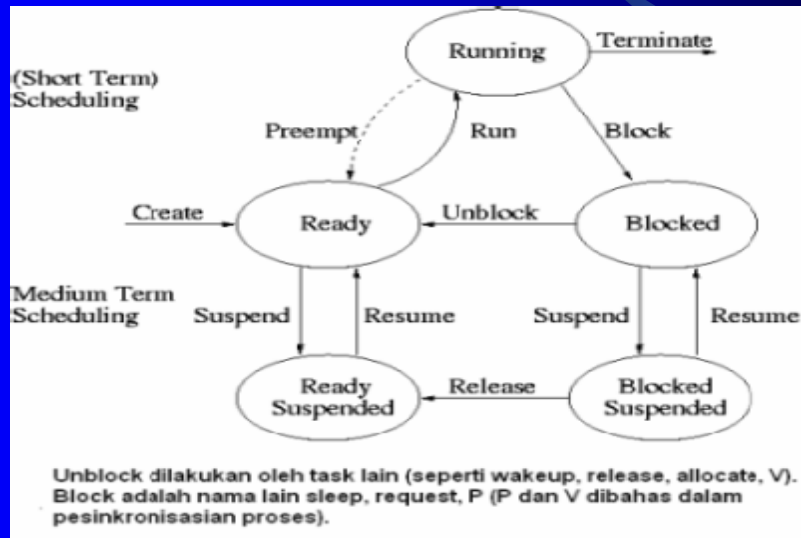
# Alih Konteks (Context Switch)

- | Ketika alih konteks terjadi, kernel menyimpan konteks dari proses lama ke dalam PCB nya dan mengisi konteks yang telah disimpan dari process baru yang telah terjadual untuk berjalan.
- | Pergantian waktu konteks adalah murni overhead, karena sistem melakukan pekerjaan yang tidak perlu.
- | Kecepatannya bervariasi dari mesin ke mesin, bergantung pada kecepatan memori, jumlah register yang harus di copy, dan keberadaan instruksi khusus (seperti instruksi tunggal untuk mengisi atau menyimpan seluruh register).
- | Tingkat kecepatan umumnya berkisar antara 1 sampai 1000 mikro detik.

# Operasi Proses

- | Proses dalam sistem dapat dieksekusi secara bersama-sama, proses tersebut harus dibuat dan dihapus secara dinamis. Maka, sistem operasi harus menyediakan suatu mekanisme untuk pembuatan proses dan terminasi proses.

# Operasi Proses



# Pembuatan Proses

- | Suatu proses dapat membuat beberapa proses baru, melalui sistem pemanggilan pembuatan proses, selama jalur eksekusi.
- | Pembuatan proses dinamakan induk proses, sebagaimana proses baru di sebut anak dari proses tersebut. Tiap proses baru tersebut dapat membuat proses lainnya, membentuk suatu pohon proses

# Pembuatan Proses

- | Ketika suatu proses membuat proses baru, dua kemungkinan ada dalam term eksekusi:
    1. Induk terus menerus untuk mengeksekusi secara bersama-sama dengan anaknya.
    2. Induk menunggu sampai sebagian dari anaknya telah diakhiri/terminasi.
- Juga ada dua kemungkinan dalam term dari *address space* pada proses baru:
1. Anak proses adalah duplikat dari induk proses.
  2. Anak proses memiliki program yang terisikan didalamnya.

# Terminasi Proses

- | Sebuah proses berakhir ketika proses tersebut selesai mengeksekusi pernyataan akhirnya dan meminta sistem operasi untuk menghapusnya dengan menggunakan sistem pemanggilan exit.
- | Pada titik itu, proses tersebut dapat mengembalikan data (keluaran) pada induk prosesnya (melalui sistem pemanggilan wait)
- | Seluruh sumber-sumber dari proses-termasuk memori fisik dan virtual, membuka berkas, dan penyimpanan I/O di tempatkan kembali oleh sistem operasi.

# Terminasi Proses

- | Ada situasi tambahan tertentu ketika terminasi terjadi. Sebuah proses dapat menyebabkan terminasi dari proses lain melalui sistem pemanggilan yang tepat (contoh abort).
- | Biasanya, sistem seperti itu dapat dipanggil hanya oleh induk proses tersebut yang akan diterminasi. Bila tidak, pengguna dapat secara sewenang-wenang membunuh job antara satu sama lain.
- | Catat bahwa induk perlu tahu identitas dari anaknya. Maka, ketika satu proses membuat proses baru, identitas dari proses yang baru diberikan kepada induknya.

# Terminasi Proses

- | Induk dapat menterminasi/ mengakhiri satu dari anaknya untuk beberapa alasan, seperti:
- | Anak telah melampaui kegunaannya atas sebagian sumber yang telah diperuntukkan untuknya.
- | Pekerjaan yang ditugaskan kepada anak telah keluar, dan sistem operasi tidak memeperbolehkan sebuah anak untuk meneruskan jika induknya berakhir.