

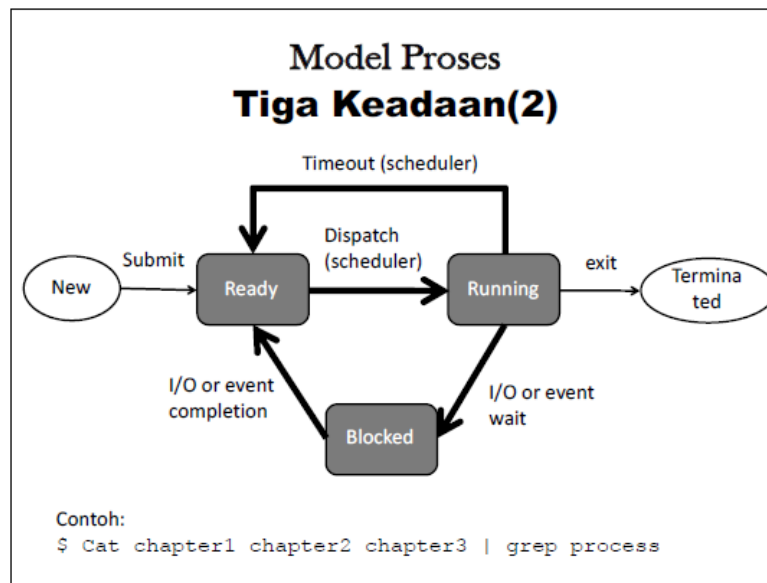
**PERINGATAN !!! PEMBAHASAN DI BAWAH INI BELUM TENTU BENAR.**

- 1) a. System Call untuk manajemen proses diperlukan untuk mengatur proses-proses yang sedang berjalan. Kita dapat melihat penggunaan system calls untuk manajemen proses pada Sistem Operasi Unix. Contoh yang paling baik untuk melihat bagaimana system call bekerja untuk manajemen proses adalah Fork.  
  
b. Contoh :
  - (1) **Process control** merupakan system calls yang mengendalikan proses-proses yang berjalan. Contohnya load, execute, create process.
  - (2) **File manipulation** adalah kumpulan system calls yang bertugas untuk melakukan manipulasi file seperti pembacaan, penulisan, penghapusan dan pengubahan. Contohnya create file, delete file, open, close.
  - (3) **Device manipulation** adalah system calls yang mengatur penggunaan peralatan yang terhubung pada mesin tersebut. Contohnya read, write, reposition.
- 2) a.
  - (1) di beberapa aplikasi, banyak kegiatan dapat berlangsung sekaligus.
  - (2) Lebih mudah untuk membuat dan menghancurkan proses daripada membuat proses.
  - (3) Performa yang lebih baik untuk aplikasi terikat input output.
  - (4) Thread sangat berguna pada sistem dengan menggunakan banyak CPU  
b. Thread di satu proses memakai bersama sumber daya yang dimiliki proses, yaitu :
  - (1) Ruang alamat.
  - (2) Proses-proses anak.  
c. Nilai yang dihasilkan oleh fungsi getpid() dalam blok fungsi ThreadRoutine dan blok fungsi main adalah sama karena dengan threading fungsi anak dan thread berbagi 1 proses yang sama, maka pid yang dimiliki adalah sama

**PERINGATAN !!! PEMBAHASAN DI BAWAH INI BELUM TENTU BENAR.**

- 3) a. Proses yang dihasilkan oleh kode tersebut sebanyak 5 proses.
- b. Urutan tampilan yang dihasilkan :
- Who am I 1  
Who am I 1  
I am Powerful 2  
Who am I 2  
Who am I 2
- c. Peran wait() adalah menunggu proses anak untuk terminasi sebelum melanjutkan ke proses parent.
- d. Child dapat memanggil FORK untuk membentuk child lainnya. Sistem call akan memanggil binary file tertentu yang ada pada memori dan mengeksekusinya sebagai child. Parent akan running kembali setelah child selesai eksekusi. Dengan demikian parent dan child tidak dapat berjalan secara parallel.

4) a.



- b. (1) Jumlah proses yang berstatus *ready* pada tiap satuan waktu **Tidak dibatasi**
- (2) Jumlah proses yang berstatus *running* pada tiap satuan waktu **Maksimal 1**
- (3) Jumlah proses yang berstatus *blocked* pada tiap satuan waktu **Tidak dibatasi**

**PERINGATAN !!! PEMBAHASAN DI BAWAH INI BELUM TENTU BENAR.**

5) a) ukuran untuk batch itu throughput maksimal(jumlah job per waktu) dan turn around time kecil(memiminalkan waktu dari mulai sampai selesai) dan cpu harus selalu sibuk. User tidak butuh respon cepat dan peralihan proses dalam batch harus sedikit sehingga overhead kecil. Sehingga algoritma yang paling cocok adalah **SJF** jika diketahui process burst nya. jika tidak diketahui lebih bagus menggunakan **FCFS**.

b) ukuran untuk sistem interaktif adalah response time dan proporsionalitas(memenuhi harapan user). Disarankan menggunakan **Algoritma Preemptive**

c)

→ **FCFS**

Nama Proses	Waktu tiba	Lama eksekusi	Waktu tunggu (= mulai eksekusi - waktu tiba)	Mulai Eksekusi (= selesai eksekusi proses sblmnya)	Selesai Eksekusi (=mulai eksekusi + lama eksekusi)	Turn-around Time (=selesai eksekusi - waktu tiba)
A	0	1	0	0	1	1
B	0	5	1	1	6	6
C	0	3	6	6	9	9
D	0	6	9	9	15	15
E	0	2	15	15	17	17
<b>Total</b>			<b>31</b>			<b>48</b>

Rata-rata WT

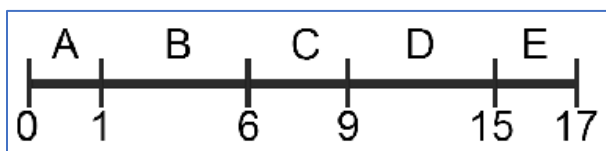
**6.2**

$\Sigma TA =$

**48**

Rerata TA =

**9.6**



PERINGATAN !!! PEMBAHASAN DI BAWAH INI BELUM TENTU BENAR.

→ SJF

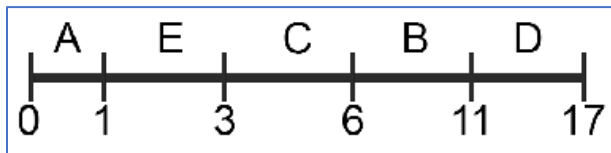
Urutan Proses	Waktu tiba	Lama eksekusi	Waktu tunggu	Mulai Eksekusi	Selesai Eksekusi	TA
A	0	1	0	0	1	1
E	0	2	1	1	3	3
C	0	3	3	3	6	6
B	0	5	6	6	11	11
D	0	6	11	11	17	17
Total			21			38

Rata-rata WT

4.2

$\Sigma TA = 38$

Rerata TA = 7.6



→ Round Robin (kuantum = 2)

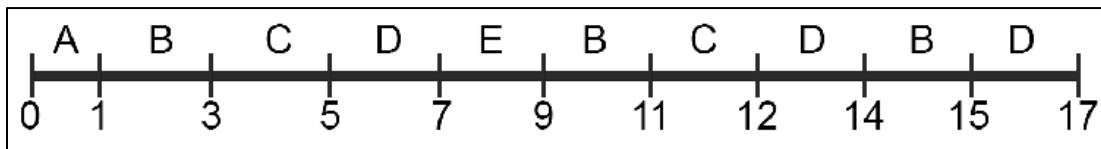
Urutan Proses	Waktu tiba	Lama eksekusi	Waktu tunggu	Mulai Eksekusi	Selesai Eksekusi	TA
A	0	1	0	0	1	1
B	0	5	10	1	15	15
C	0	3	9	3	12	12
D	0	6	11	12	17	17
E	0	2	7	7	9	9
Total			37			54

Rata-rata WT

7.4

$\Sigma TA = 54$

Rerata TA = 10.8



**PERINGATAN !!! PEMBAHASAN DI BAWAH INI BELUM TENTU BENAR.**

- 6) a) X bernilai 305 jika yang dieksekusi pertama adalah Process A dan dilanjutkan Process B.  
Dan X bernilai 315 jika eksekusi pertama Process B dilanjutkan dengan Process A.
- b) -
- c) -
- d) -