

# Sistem Operasi - 5

Rudi Rosadi, S.Si., M.Kom /  
Rahmatullah Arrizal P, S.Kom, M.T

# Penjadwalan Proses

- Penjadwalan merupakan kumpulan kebijaksanaan dan **mekanisme** di sistem operasi yang berkaitan dengan **urutan kerja** yang dilakukan sistem komputer.
- Proses Apa? Kapan dan Berapa lama berlangsung?
- Jika banyak proses yang akan dijalankan maka SO harus memutuskan proses mana yang dijalankan terlebih dahulu
- Bagian SO yang menjadi pembuat keputusan disebut dgn : Penjadwal (Scheduler)
- Algoritmanya : Scheduling Algorithm
- Sasaran utama penjadwalan proses adalah optimasi kinerja menurut kriteria tertentu

# Konsep Dasar Penjadwalan

- Sistem multiprogramming/multitasking, terjadi beberapa proses berjalan dalam suatu waktu.
- Konsep dasar multiprogramming : suatu proses akan menggunakan CPU sampai proses tersebut dalam status wait (misalnya meminta I/O) atau selesai
- Pada saat wait, maka CPU akan menganggur, u/ mengatasinya CPU dialihkan ke proses lain, demikian seterusnya.
- Proses dijalankan :
  - Terjadi siklus eksekusi CPU dan wait I/O yang disebut CPU-I/O burst.
- Eksekusi proses dimulai dengan CPU burst dan dilanjutkan dengan I/O burst, diikuti CPU burst lain, kemudian I/O burst lain.

# Kriteria Penjadwalan

## 1. **Fairness** (adil) :

- proses-proses diperlakukan sama, yaitu mendapat jatah waktu pemroses yang sama dan tak ada proses yang tak kebagian layanan pemroses sehingga mengalami kekurangan waktu.
- Meyakinkan bahwa tiap-tiap proses akan mendapatkan pembagian waktu penggunaan CPU secara terbuka

## 2. **Response time** (waktu tanggap) :

- Waktu yang dibutuhkan CPU untuk menanggapi proses yang masuk
- Sistem Interaktif: Adalah waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukkan atau transaksi sampai hasil pertama muncul di layar (terminal)
- Sistem Waktu Nyata (Realtime) : Simulasi yang menyamai proses yang sebenarnya di dunia nyata, terdapat batasan waktu
- Sasaran meminimumkan waktu tanggap

## 3. **Waiting Time** :

- Waktu yang diperlukan oleh suatu proses untuk menunggu di ready queue

# Kriteria Penjadwalan (Cont'd)

## 4. **Turnaround time :**

- Banyaknya waktu yang diperlukan untuk mengeksekusi proses, dari mulai menunggu untuk meminta tempat di memori utama, menunggu di ready queue, eksekusi oleh CPU, dan mengerjakan I/O.
- Turn around time = waktu eksekusi + waktu menunggu
- Sasaran penjadwalan adalah meminimumkan turn around time.

## 5. **Throughput :**

- Adalah jumlah kerja atau jumlah job yang dapat diselesaikan dalam satu unit waktu.
- Sasaran memaksimumkan jumlah pekerjaan

## 6. **Efisiensi :**

- Menjaga agar CPU senantiasa sibuk sepanjang waktu
- Utilitas CPU dinyatakan dalam bentuk % yaitu 0-100%. Namun dalam kenyataannya hanya berkisar antara 40-90%

# Penjadwalan CPU

- Pada saat CPU menganggur, SO menyeleksi proses-proses yang ada di memori utama (ready queue) untuk dieksekusi dan mengalokasikan CPU untuk salah satu proses tersebut, seleksi ini disebut short-term scheduler (CPU scheduler)
- Keputusan untuk menjadwalkan CPU mengikuti 4 keadaan :
  1. Apabila proses berpindah dari keadaan running ke waiting;
  2. Apabila proses berpindah dari keadaan running ke ready;
  3. Apabila proses berpindah dari keadaan waiting ke ready;
  4. Apabila proses berhenti.

# Strategi Penjadwalan

- Apabila model penjadwalan yang dipilih menggunakan keadaan 1 dan 4, maka penjadwalan semacam ini disebut non-preemptive.
  - Apabila yang digunakan adalah keadaan 2 dan 3, maka disebut dengan preemptive
- 
- Preemptive Scheduling :
    - Strategi yang memungkinkan sebuah proses yang dijalankan ditunda sementara
    - Contoh : Round Robin, Prioritas
  - Non Preemptive Scheduling
    - Tidak bisa ditunda
    - Contoh : Shortest Job First, FCFS

# First-Come First-Served Scheduling (FCFS)

- Proses yang pertama kali meminta jatah waktu untuk menggunakan CPU akan dilayani terlebih dahulu.
- Proses yang meminta CPU pertama kali akan dialokasikan ke CPU pertama kali



# Contoh

Process	Burst Time
P1	24
P2	3
P3	3

Gant Chart dengan penjadwalan FCFS adalah sebagai berikut :



- Waktu tunggu untuk **P1** adalah 0, **P2** adalah 24 dan **P3** adalah 27 sehingga rata-rata waktu tunggu adalah  $(0 + 24 + 27)/3 = 17$  milidetik.
- Sedangkan apabila proses datang dengan urutan **P2**, **P3**, dan **P1**, hasil penjadwalan CPU dapat dilihat pada gant chart berikut :



# Job Terpendek Pertama (Shortest Job First)

- Digunakan untuk sistem batch
- Run time diketahui lebih dahulu
- Mengetahuinya dengan teknik estimasi
- Pada penjadwalan SJF, proses yang memiliki CPU burst paling kecil dilayani terlebih dahulu
- Terdapat dua skema :
  1. Non preemptive, bila CPU diberikan pada proses, maka tidak bisa ditunda sampai CPU burst selesai.
  2. Preemptive, jika proses baru datang dengan panjang CPU burst lebih pendek dari sisa waktu proses yang saat itu sedang dieksekusi, proses ini ditunda dan diganti dengan proses baru. Skema ini disebut dengan Shortest-Remaining-Time-First (SRTF).

- Meskipun algoritma ini optimal, namun pada kenyataannya sulit untuk diimplementasikan karena sulit untuk mengetahui panjang CPU burst berikutnya.
- Namun nilai ini dapat diprediksi. CPU burst berikutnya biasanya diprediksi sebagai suatu rata-rata eksponensial yang ditentukan dari CPU burst sebelumnya atau “Exponential Average”

$$\tau_{n+1} = \alpha t_0 + (1 - \alpha) \tau_n$$

- $\tau_{n+1}$  = Panjang CPU burst yang diperkirakan
- $\tau_n$  = Panjang CPU burst sebelumnya
- $t_n$  = Panjang CPU burst yang ke-n (yang sedang berlangsung)
- $\alpha$  = Ukuran Pembanding antara  $\tau_{n+1}$  dengan  $\tau_n$  (0 sampai 1)

Sebagai contoh, jika  $\alpha = 0,5$ , dan:

CPU burst ( $t_n$ ):	6	4	6	4	13	13	13	...	
$\tau_n$ :	10	8	6	6	5	9	11	12	...

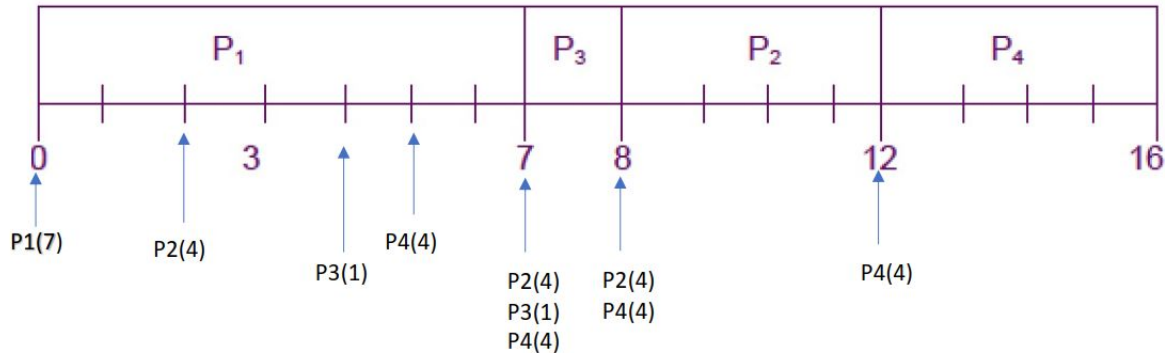
Pada awalnya  $t_1 = 6$  dan  $\tau_1 = 10$ , sehingga :

$$\tau_2 = 0,5 * 6 + (1 - 0,5) * 10 = 8$$

$$\tau_3 = 0,5 * 4 + (1 - 0,5) * 8 = 6$$

# Contoh Non-Preemptive

Process	Arrival Time	Burst Time
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4



WT : P<sub>1</sub>=0

P<sub>2</sub>=8-2=6

P<sub>3</sub>=7-4=3

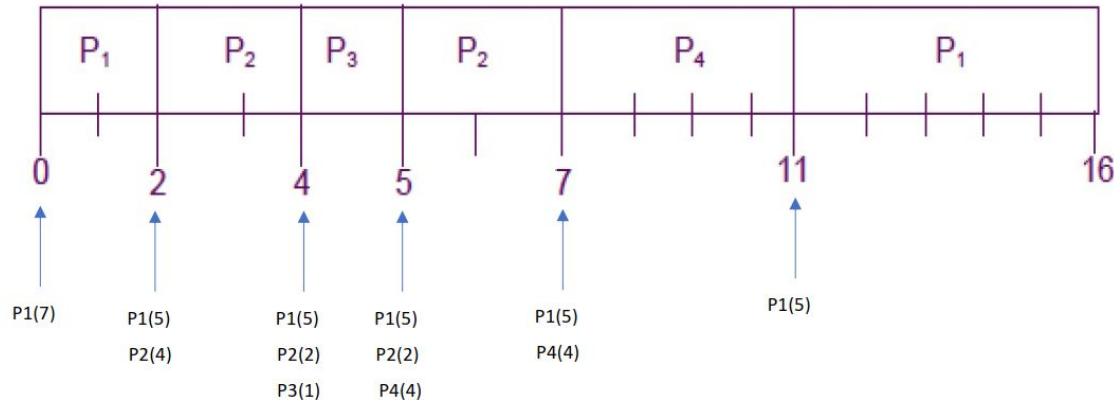
P<sub>4</sub>=12-5=7

$$AWT = (P_1 + P_2 + P_3 + P_4) / 4$$

$$= (0 + 6 + 3 + 7) / 4 = 4\text{ms}$$

# Contoh Preemptive

Process	Arrival Time	Burst Time
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4



$$\begin{aligned}
 WT : P1 &= 0 + (11 - 2) = 9 \\
 P2 &= (2 - 2) + (5 - 4) = 1 \\
 P3 &= 0 \\
 P4 &= (7 - 5) = 2
 \end{aligned}$$

$$\begin{aligned}
 AWT &= (P1 + P2 + P3 + P4) / 4 \\
 &= (9 + 1 + 0 + 2) / 4 = 3\text{ms}
 \end{aligned}$$

# Penjadwalan Prioritas

- Setiap proses diberi nomor prioritas (integer)
- CPU dialokasikan untuk proses yang memiliki prioritas paling tinggi (nilai integer terkecil biasanya merupakan prioritas terbesar)
- Jika beberapa proses memiliki prioritas yang sama, maka akan digunakan algoritma FCFS.
- Untuk mencegah proses dengan prioritas tinggi terus menerus maka prioritas proses akan diturunkan setiap interupsi sampai ada prioritas lain yang lebih tinggi.



# Contoh

- Misalnya terdapat lima proses **P1, P2, P3, P4** dan **P5** yang datang secara berurutan dengan **CPU burst** dalam milidetik

Process	Burst time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2



Waktu tunggu untuk **P1** adalah 6, **P2** adalah 0, **P3** adalah 16, **P4** adalah 18 dan **P5** adalah 1 sehingga rata-rata waktu tunggu adalah  $(6 + 0 + 16 + 18 + 1)/5 = 8.2$  milidetik

# Soal

- Terdapat 5 job (P1, P2, P3, P4, P5) yang datang hampir pada saat yang bersamaan. Estimasi waktu eksekusi (burst time) masing-masing 10, 6, 12, 9 dan 5 detik, dengan arrival time (0.0, 3.0, 5.0, 7.0, 8.0)

Process	Arrival Time	Burst Time
P1	0.0	10
P2	3.0	6
P3	5.0	12
P4	7.0	9
P5	8.0	5

Buatlah Gant Chart dan tentukan rata-rata waktu tunggu algoritma penjadwalan

- FCFS
- SJF (Non Preemptive)
- SJFP (Preemptive)