

Nama : Anggia Dwi Putri Yulistin  
 NRP : 3123521028  
 Prodi : D3 Teknik Informatika A

1. Misalkan sebuah algoritma penjadwalan lebih mengutamakan proses yang menggunakan waktu prosessor paling sedikit baru-baru ini. Mengapa algoritma ini lebih menguntungkan program I/O-bound dan tidak membuat program CPU-bound kelaparan secara permanen?

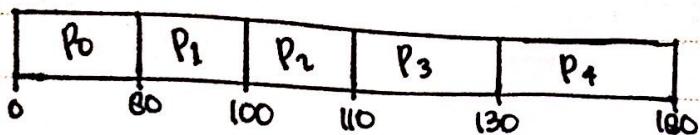
=> Algoritma yang mengutamakan proses yang menggunakan waktu prosessor paling sedikit biasanya merujuk pada algoritma penjadwalan yang mengoptimalkan CPU Utilization. Algoritma ini cenderung menguntungkan untuk proses-proses I/O bound karena cenderung memiliki waktu eksekusi yang singkat, sehingga proses tersebut tidak perlu menunggu lama dalam antrian. Ini tidak membuat program CPU-bound kelaparan secara permanen karena algoritma penjadwalan seperti SJF juga memberikan kesempatan bagi proses yang membutuhkan CPU secara intensif untuk di eksekusi. Proses CPU-bound akan dijadwalkan ketika CPU tersedia dan mungkin harus menunggu lebih lama jika terdapat proses I/O-bound yg lebih singkat waktu eksekusinya di antre di depan. Namun ini tidak menyebabkan CPU kelaparan permanen karena proses CPU-bound akan tetap di eksekusi ketika CPU tersedia.

2. Misal terdapat tugas-tugas yang harus dijalankan dengan satu proses

i	T (Pi)
0	80
1	20
2	10
3	20
4	50

Date: 10/10/2023

a.) Diagram batang untuk penjadwalan FCFS



b.) Waktu Turnaround untuk proses P3

➢ Proses P3 mulai pada waktu 110 (setelah P0, P1, dan P2 selesai)

➢ Proses P3 berjalan selama 20 unit. Jadi selesai pada waktu 130

$$\text{Waktu turnaround} = 130 - 0 = 130 \text{ unit waktu}$$

c.) Rata-rata waktu tunggu untuk proses tersebut

Rata-rata waktu tunggu :  $\frac{\text{Total waktu tunggu}}{\text{Jumlah proses}}$

$$\text{Rata-rata waktu tunggu} = 0 + 80,0 + 100,0 + 110,0 + 130,0 = \frac{420}{5} = 84$$

3. Diketahui: Rata-rata proses = 6 proses / menit

Rata-rata waktu tahanan = 2 menit

Dijawab:  $U = \lambda \cdot T_s$

$$U = 6 \text{ proses/menit} \times \frac{2}{60} \text{ menit per proses}$$

$$= 0,8$$

Fraksi waktu CPU yang sibuk dalam sistem dengan satu prosesor adalah 0,8 / 80%

4.) Sebuah algoritma penjadwalan CPU menentukan urutan eksekusi berbagai proses. Jika ada proses yang akan dijadwalkan pada satu prosesor berapa banyak jadwal yg mungkin ada? Berikan rumus dalam  $n$ .

=> Untuk menentukan berapa banyak jadwal yg mungkin ada untuk satu prosessor menggunakan rumus berikut :

$$n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$$

- $n$  adalah jumlah proses yang akan dijadwalkan pada satu prosessor
- $!$  menunjukkan faktorial , yang merupakan hasil perkalian dari semua bilangan positif hingga  $n$ .

## 5.) Hubungan antara Algoritma Penjadwalan CPU

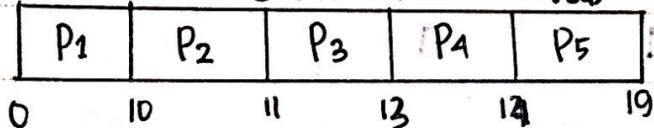
- a. Priority and SJF : Algoritma SJF adalah kasus khusus dari algoritma dimana prioritas ditentukan oleh waktu eksekusi terpendek, dari proses SJF dapat dianggap sebagai algoritma prioritas dimana prioritas diberikan berdasarkan durasi pekerjaan yg paling pendek.
- b. Multilevel Feedback Queues and FCFS : Algoritma Multilevel Feedback Queues bisa dikonfigurasikan sehingga salah satu antrian menggunakan FCFS. MLFQ adalah set algoritma yang lebih umum dimana FCFS adalah salah satu kemungkinan implementasi dalam konfigurasi antriannya. Jadi FCFS adalah subset dari MLFQ jika dalam MLFQ diatur menggunakan algoritma FCFS.
- c. Priority and FCFS : FCFS adalah kasus khusus dari algoritma prioritas dimana semua proses memiliki prioritas yang sama. Dalam FCFS, proses dijadwalkan berdasarkan urutan kedatangan tanpa mempertimbangkan prioritas tertentu. Oleh karena itu, FCFS bisa dianggap sebagai algoritma prioritas dimana semua proses memiliki prioritas yang sama.
- d. Round Robin (RR) and SJF : RR dan SJF tidak memiliki hubungan langsung dalam hal subset satu sama lain. RR bekerja dengan memberikan setiap proses waktu eksekusi yang sama dalam siklus berulang, sementara SJF bekerja dengan memilih proses dengan waktu eksekusi terpendek. Mereka adalah algoritma yang berbeda dengan tujuan yang berbeda pula. RR menekankan pada distribusi waktu yg adil, sedangkan SJF berfokus pada meminimalkan waktu rata-rata tunggu.

6.) Perbedaan antara penjadwalan jangka panjang dan jangka pendek  
 => Penjadwalan jangka panjang berfokus pada seleksi dan penempatan proses ke dalam memori untuk dieksekusi, sedangkan penjadwalan jangka pendek berfokus pada pengelolaan eksekusi proses yang telah ada dalam memori utama.

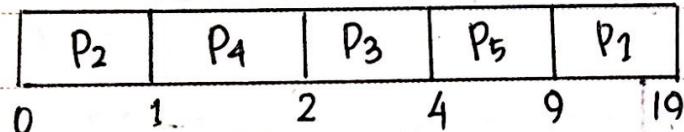
7.) a. Diagram batang untuk algoritma penjadwalan:

Process	Burst Time	Priority
P <sub>1</sub>	10	3
P <sub>2</sub>	1	1
P <sub>3</sub>	2	3
P <sub>4</sub>	1	4
P <sub>5</sub>	5	2

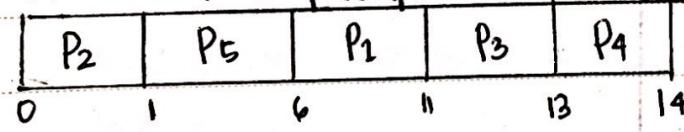
- FCFS (First-Come, First-Served)



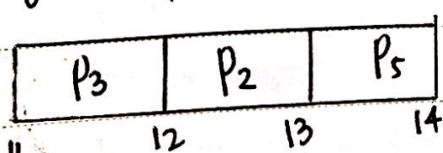
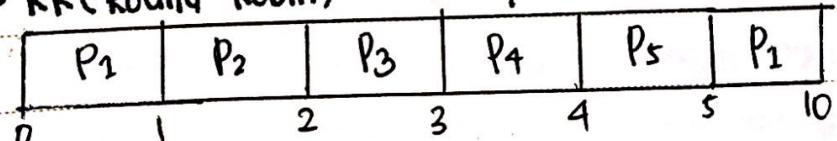
- SJF (Shortest Job First)



- Prioritas Non-preemptive



- RR (Round-Robin) dengan quantum: 1;



b. Waktu Penyelesaian untuk setiap proses

Proses	FCFS	SJF	Prioritas Non-preemptive	Round Robin
P <sub>1</sub>	10	9	16	19
P <sub>2</sub>	11	1	20	21
P <sub>3</sub>	13	4	18	7
P <sub>4</sub>	14	2	19	4
P <sub>5</sub>	19	0	6	14

c. Waktu tunggu untuk setiap proses

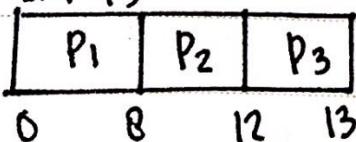
Proses	FCFS	SJF	Prioritas Non-preemptive	RR
P <sub>1</sub>	0	9	6	9
P <sub>2</sub>	10	0	0	11
P <sub>3</sub>	11	2	14	5
P <sub>4</sub>	13	21	18	3
P <sub>5</sub>	14	4	1	9

8. Perbaikkan rangkaian proses berikut dengan amanya CPU burst dan waktu kedatangannya.

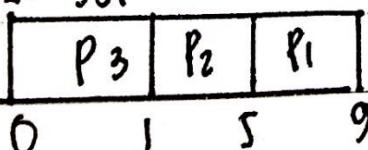
Process	Burst time	Priority
P <sub>1</sub>	8	0
P <sub>2</sub>	4	0.4
P <sub>3</sub>	1	1

a. Diagram Gantt

1. FCFS



2. SJF



## b. turnaround time

Process	FCFS	SJF
P <sub>1</sub>	13	9
P <sub>2</sub>	9	5
P <sub>3</sub>	12	12

## c. Waiting time

Process	FCFS	SJF
P <sub>1</sub>	5	1
P <sub>2</sub>	0	0
P <sub>3</sub>	0	0

## g. a. First Come First Served (FCFS)

=> Algoritma penjadwalan non-preemptive dimana proses yang datang pertama akan dijalankan pertama. Tidak ada pertimbangan terhadap panjang proses. FCFS dapat menyebabkan waktu tunggu lama untuk proses pendek jika mereka datang setelah proses panjang.

## b. Round Robin (RR)

RR adalah algoritma penjadwalan preemptive dimana setiap proses dibenjol jatah waktu yang tetap. Penjadwalan berkeliling melalui proses-proses memberikan setiap proses kesempatan untuk dijalankan selama satu quantum. Proses pendek akan mendapatkan keuntungan lebih banyak.

## c. Multilevel Feedback Queues

menggunakan beberapa antrian dengan tingkat prioritas yang berbeda. Proses secara dinamis dipindahkan antara antrian berdasarkan perilaku dan umur. MLFQ sangat mendukung proses pendek karena mereka bisa dengan cepat berpindah ke tingkat prioritas yang lebih tinggi, memastikan mereka dijalankan lebih cepat dibandingkan proses panjang.

10. a. Waiting time

⇒ Waiting time adalah waktu yang dihabiskan sebuah proses menunggu di antrian siap sebelum dieksekusi oleh CPU.

b. Response time

Response time adalah waktu yang berlalu dari saat sebuah proses diajukan hingga respons pertama dihasilkan

c. Throughput

throughput adalah jumlah proses yang diselesaikan oleh sistem dalam periode waktu tertentu.