



BAB : CPU SCHEDULING  
NAMA : DANI ADRIAN  
NIM : 225150201111009  
TANGGAL : 11/04/2023  
ASISTEN : ZHAFRAN RAMA AZMI  
GIBRAN HAKIM

## 5.4. Langkah Praktikum

1. Masukkan variabel berikut dengan jumlah proses sebanyak 3. Capture / Snapshot output-nya dan simpan sebagai laporan.

**Jawab:**

```
userlinux@Linux: ~  
userlinux@Linux:~$ ./main1  
Algoritma CPU Scheduling FCFS  
  
Jumlah Proses = 3  
  
Nama Proses = P1  
Arrival time = 1  
Burst time = 2  
  
Nama Proses = P2  
Arrival time = 1  
Burst time = 2  
  
Nama Proses = P3  
Arrival time = 2  
Burst time = 1  
  
=====   
Tabel Proses  
=====   
| no | proses | time arrival | burst |  
=====   
| 1 | P1 | 1 | 2 |  
| 2 | P2 | 1 | 2 |  
| 3 | P3 | 2 | 1 |  
=====   
Total waiting time = 5  
Turn around time = 10  
  
=====   
Tabel Waktu Proses  
=====   
| no | proses | waiting time | turn arround |  
=====   
| 1 | P1 | 0 | 2 |  
| 2 | P2 | 2 | 4 |  
| 3 | P3 | 3 | 4 |  
=====   
  
Gant-Chart  
  
P1 P2 P3  
|=====|=====|=====|  
1 3 5 6 //diperoleh dari penjumlahan Burst  
  
Average Waiting Time : 1.666667  
Average Turn Around Time : 3.333333  
userlinux@Linux:~$
```

Program ./main1 akan memberikan beberapa output, diantaranya sebagai berikut :

- Tabel proses (Nomor proses, Nama proses, Arrival time, dan Burst time)
- Total waiting time dan total turn around time untuk seluruh proses



**LABORATORIUM PEMBELAJARAN ILMU KOMPUTER**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS BRAWIJAYA**

- Tabel waktu proses (Nomor proses, Nama proses, Waiting time, dan Turnaround time)
- Gantt chart, menunjukkan urutan eksekusi proses berdasarkan waktu.
- Nilai rata-rata waiting time dan rata rata turnaround time untuk seluruh proses.

2. Jalankan program tersebut beberapa kali dengan nilai variable masukan yang berbeda-beda. Amati output hasil percobaannya.

**Jawab:**

```
userlinux@Linux: ~$ ./program1.c
Arrival time = 0
Burst time = 1
Nama Proses = B
Arrival time = 2
Burst time = 3
Nama Proses = C
Arrival time = 4
Burst time = 5
Nama Proses = D
Arrival time = 6
Burst time = 7

Tabel Proses
=====
| no | proses | time arrival | burst |
=====
| 1 | A | 0 | 1 |
| 2 | B | 2 | 3 |
| 3 | C | 4 | 5 |
| 4 | D | 6 | 7 |
=====
Total waiting time = 2
Turn around time = 18

Tabel Waktu Proses
=====
| no | proses | waiting time | turn around |
=====
| 1 | A | 0 | 1 |
| 2 | B | -1 | 2 |
| 3 | C | 0 | 5 |
| 4 | D | 3 | 10 |
=====

Gantt Chart
A B C D
=====
0 1 4 9 16 //di peroleh dari penjumlahan Burst

Average waiting time : 0.500000
Average turn around time : 4.500000
userlinux@Linux: ~$
```



**LABORATORIUM PEMBELAJARAN ILMU KOMPUTER**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS BRAWIJAYA**

```
userlinux@Linux: ~  
userlinux@Linux:~$ ./main1  
Algoritma CPU Scheduling FCFS  
  
Jumlah Proses = 3  
Nama Proses = A1  
Arrival time = 0  
Burst time = 6  
  
Nama Proses = A2  
Arrival time = 8  
Burst time = 1  
  
Nama Proses = A3  
Arrival time = 5  
Burst time = 9  
  
Tabel Proses  
=====|  
| no | proses | time arrival | burst |  
=====|  
| 1 | A1 | 0 | 6 |  
| 2 | A3 | 5 | 9 |  
| 3 | A2 | 8 | 1 |  
=====|  
Total waiting time = 8  
Turn around time = 24  
  
Tabel Waktu Proses  
=====|  
| no | proses | waiting time | turn around |  
=====|  
| 1 | A1 | 0 | 6 |  
| 2 | A3 | 1 | 10 |  
| 3 | A2 | 7 | 8 |  
=====|  
  
Gant-Chart  
A1 A3 A2  
|=====|=====|=====|  
0 6 15 16 //diperoleh dari penjumlahan Burst  
  
Average Waiting Time : 2.666667  
Average Turn Around Time : 8.000000  
userlinux@Linux:~$
```

Dani Adrian  
225150201111009

Ln 1, Col 1 100% Windows (CRLF) UTF-8

Inputan nilai variable yang berbeda-beda akan memberikan output nilai yang berbeda-beda pula.

3. Dari hasil percobaan tersebut, simpulkanlah jenis algoritma penjadwalan apa yang digunakan dalam program tersebut?

**Jawab:**

First Come First Serve, sebagai salah satu algoritma penjadwalan berbasis antrian yang paling sederhana. Proses pertama yang tiba akan dieksekusi terlebih dahulu tanpa memperhatikan prioritas atau kepentingan suatu proses. Proses dengan arrival time yang lebih awal akan dieksekusi terlebih dahulu.



## 5.5. Pembahasan

1. Masukkan variable berikut dengan: Jumlah proses sebanyak 3, time quantum 2, Amati hasil outputnya!

Nama Proses	Arrival Time	Burst Time
P1	0	3
P2	1	2
P3	2	1

Jawab:

```
userlinux@Linux: ~  
userlinux@Linux: $ ./main2  
Masukkan jumlah proses : 3  
Masukkan arrival time untuk Proses P1 : 0  
Masukkan burst time untuk Proses P1 : 3  
Masukkan arrival time untuk Proses P2 : 1  
Masukkan burst time untuk Proses P2 : 2  
Masukkan arrival time untuk Proses P3 : 2  
Masukkan burst time untuk Proses P3 : 1  
Masukkan time quantum 2  
  
Process |Turnaround time|waiting time  
P[2] | 3 | 1  
P[3] | 3 | 2  
P[1] | 6 | 3  
  
Average Waiting Time = 2.000000  
Average Turnaround Time = 4.000000  
userlinux@Linux: $
```

Output berisi informasi diantaranya sebagai berikut :

- Tabel proses (Nomor proses, Nama proses, Waiting time, dan Turnaround time)
- Rata-rata waiting time untuk seluruh proses
- Rata-rata turnaround time untuk seluruh proses

Output akan menganalisis kinerja algoritma CPU scheduling FCFS dalam mengatur eksekusi proses berdasarkan arrival dan burst time.

2. Jika variable pada nomor 4 di atas diubah menjadi: jumlah proses sebanyak 3, time quantum 2. Amati hasil outputnya!

Nama Proses	Arrival Time	Burst Time
P1	0	5
P2	20	4
P3	25	5

Jawab:



```
userlinux@linux:~$ ./main2
Masukkan jumlah proses : 3
Masukkan arrival time untuk Proses P1 :0
Masukkan burst time untuk Proses P1 :5
Masukkan arrival time untuk Proses P2 :20
Masukkan burst time untuk Proses P2 :4
Masukkan arrival time untuk Proses P3 :25
Masukkan burst time untuk Proses P3 :5
Masukkan time quantum 2

Process |Turnaround time|waiting time
P[1]    |      5      |      0
```

Ketika jumlah proses diubah menjadi 3 dan time quantum 2, output yang ditampilkan akan berbeda dibandingkan sebelumnya. Output akan berhenti di P[1] dan tidak akan melanjutkan output Turnaround time dan Waiting time seluruh proses.

3. Berdasarkan hasil pengamatan anda, maka algoritma apa yang akan digunakan dalam program tersebut?

**Jawab:**

Round Robin, algoritma penjadwalan proses yang menggunakan konsep time quantum untuk membagi waktu CPU antara proses-proses yang berada dalam antrian. Setiap proses diberikan waktu eksekusi sebanyak time quantum, dan jika proses tersebut belum selesai, maka akan dikembalikan ke antrian dan proses berikutnya akan dieksekusi selama time quantum yang sama.

## 5.6. Kesimpulan

First Come First Served adalah algoritma penjadwalan proses yang mengeksekusi proses berdasarkan urutan kedatangan (arrival time) ke CPU. Proses yang datang pertama kali akan dieksekusi terlebih dahulu dan proses proses berikutnya akan dieksekusi sesuai dengan urutan kedatangan. First Come First Serve menghasilkan waktu tunggu (waiting time) yang lama jika ada suatu proses yang memiliki burst time yang lama dan tiba sebelum proses lain yang sudah ada di antrian.

Round Robin adalah algoritma penjadwalan proses yang menggunakan konsep time quantum untuk membagi waktu CPU secara adil antara proses-proses yang



**LABORATORIUM PEMBELAJARAN ILMU KOMPUTER**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS BRAWIJAYA**

---

dalam antrian. Setiap proses diberikan waktu eksekusi sebanyak time quantum, dan jika proses belum selesai, maka akan dikembalikan ke antrian dan proses berikutnya akan dieksekusi selama time quantum yang sama.