

LAPORAN PRAKTIKUM SISTEM OPERASI

**MODUL 11 : PENJADWALAN PROSES DAN MANAJEMEN MEMORI
(OS SIM)**



DISUSUN OLEH :

NAMA : BIMA TRIADMAJA

NIM : L200210137

KELAS : C

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2022/2023**

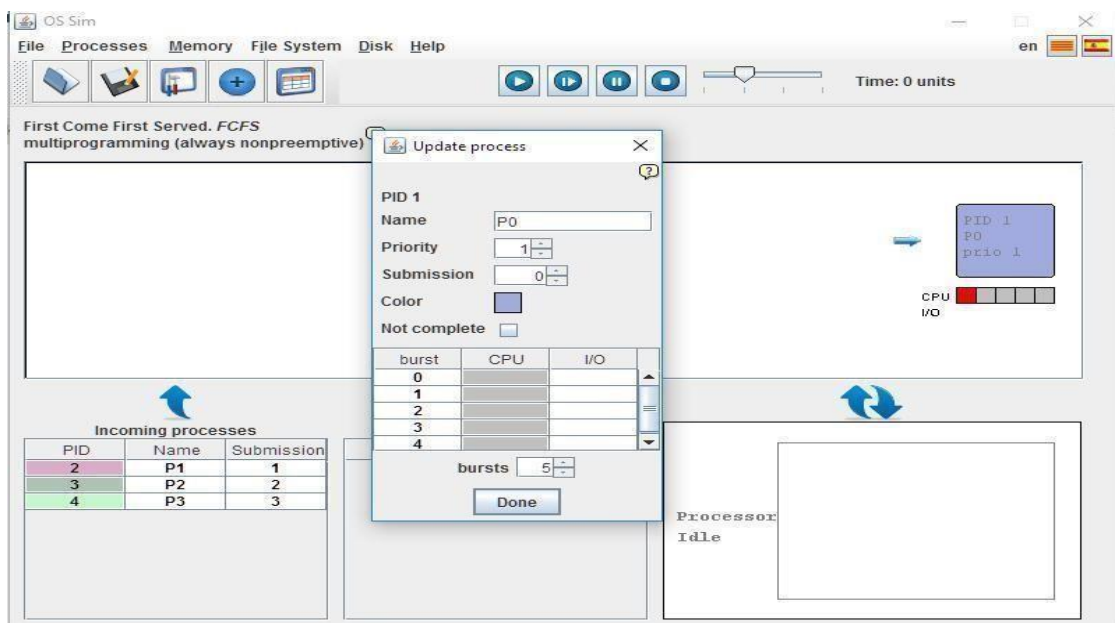
Langkah Kerja

Kegiatan 1. Penjadwalan Proses

1.1 First-Come, First-Served (FCFS)

- Bukalah program OSSim, selanjutnya pilih menu processes -> processsscheduling
- Selanjutnya pilihlah setting dan pilih algoritma **First-Come, First-Served (FCFS)**
- Lakukan input proses sesuai dengan tabel berikut dengan memulai dengan P0 sebagai input proses yang pertama

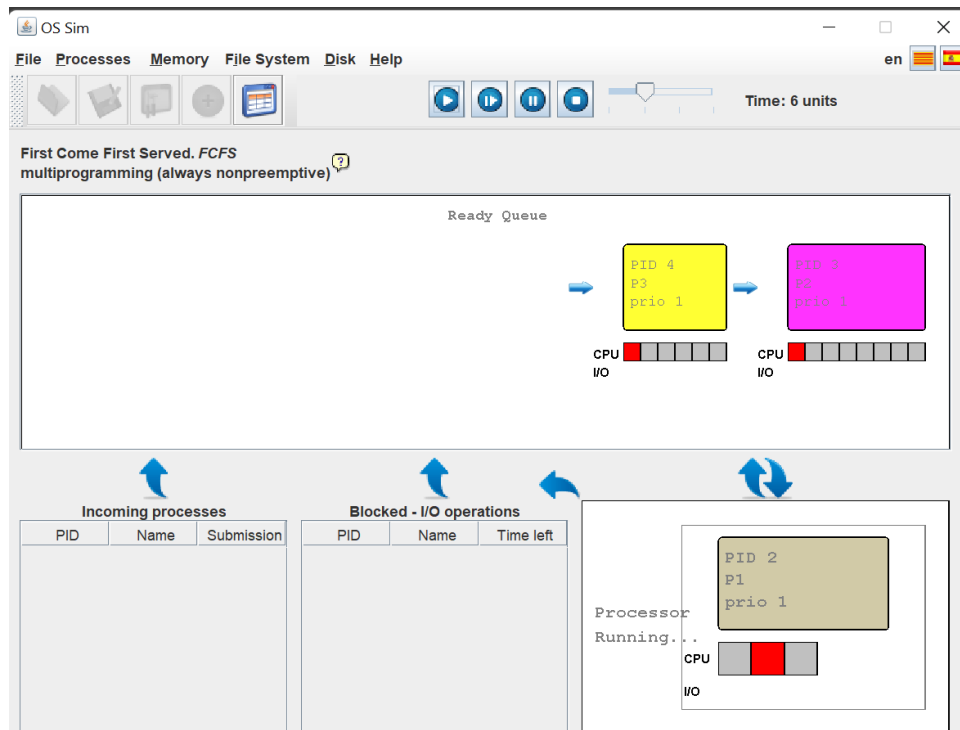
Process	Arrival Time	Burst Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	8
P3	3	6	16



- Jika input sudah selesai dilakukan. Pilih tombol start pada bagian atas. Amati dan analisa proses yang terjadi.

➤ Hasil Kegiatan :

1. Proses :



2. Hasil :

Process Scheduling Information											
Efficiency (%)		1.00									
Throughput (processes/time unit)		0.18									
Avg. Turnaround Time (time)		12.75									
Avg. Waiting Time (time)		7.25									
Avg. Response Time (time)		7.25									
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO	
1	P0	1	0	-	5	0	0	5	1.0	0.0	
2	P1	1	0	-	3	5	5	8	0.375	0.0	
3	P2	1	0	-	8	8	8	16	0.5	0.0	
4	P3	1	0	-	6	16	16	22	0.272727...	0.0	

3. Tabel :

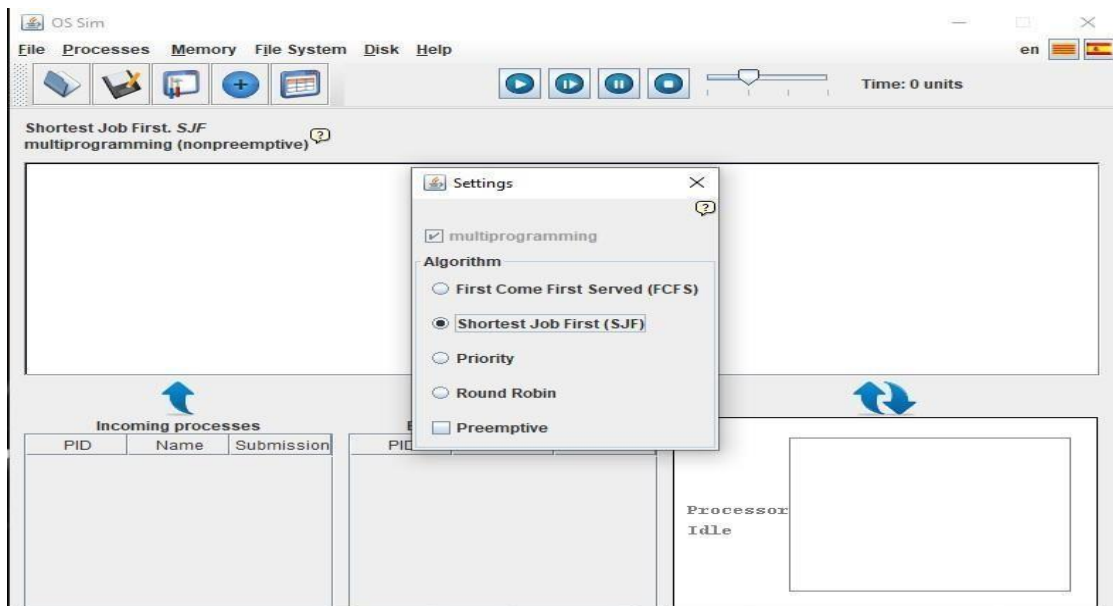
Process	Wait Time : Service Time – Arrival Time
P0	0
P1	5
P2	8
P3	16
Av Wait Time	7.25

- Pada algoritma ini, proses di eksekusi dimulai dari antrian pertama terlebih dahulu dengan menggunakan konsep first in first out. Ini merupakan algoritma penjadwalan yang paling sederhana.

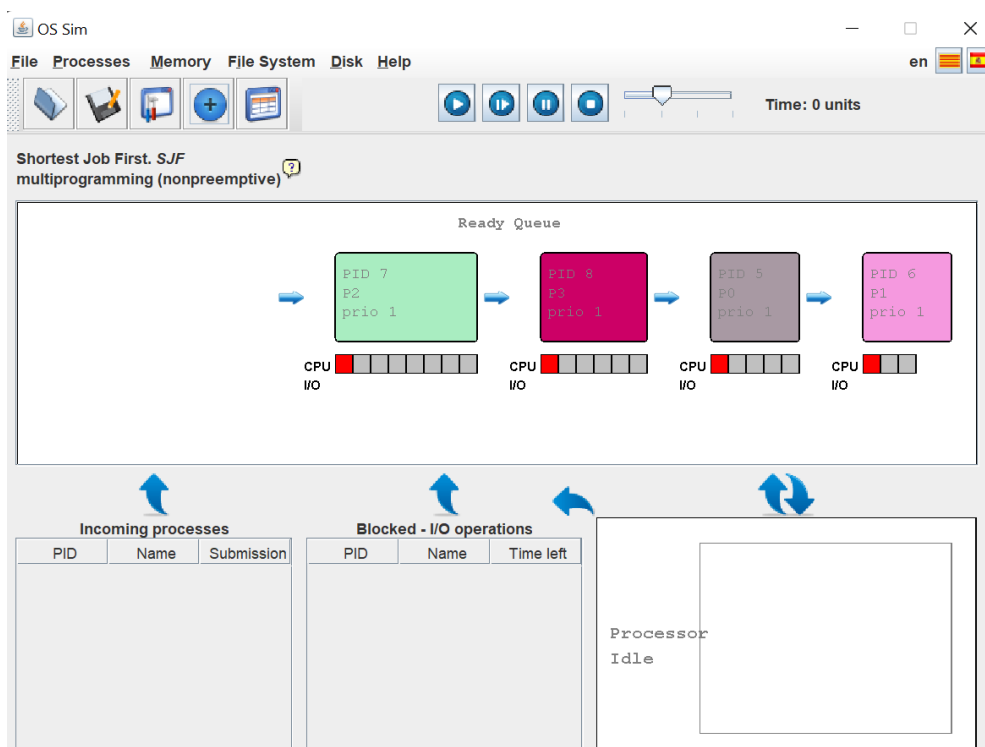
1.2 Shortest Job First (SJF)

- Bukalah program OS Sim, selanjutnya pilih menu processes > process scheduling
- Selanjutnya pilihlah setting dan pilih algoritma **Shortest Job First (SJF)**. algoritma ini terdiri dari 2 jenis yaitu non- preemptive dan preemptive. Untuk mengaktifkan preemptive dengan mencentang menu tersebut. Sebaliknya jika menonaktifkan maka hanya cukup menghilangkan centang saja.
- Selanjutnya klik tombol start. Amati dan analisa proses yang terjadi. Lakukan perbandingan dari hasil keduanya.

➤ Hasil Kegiatan : SJF Non-Preemptive



- Proses :



b. Hasil :

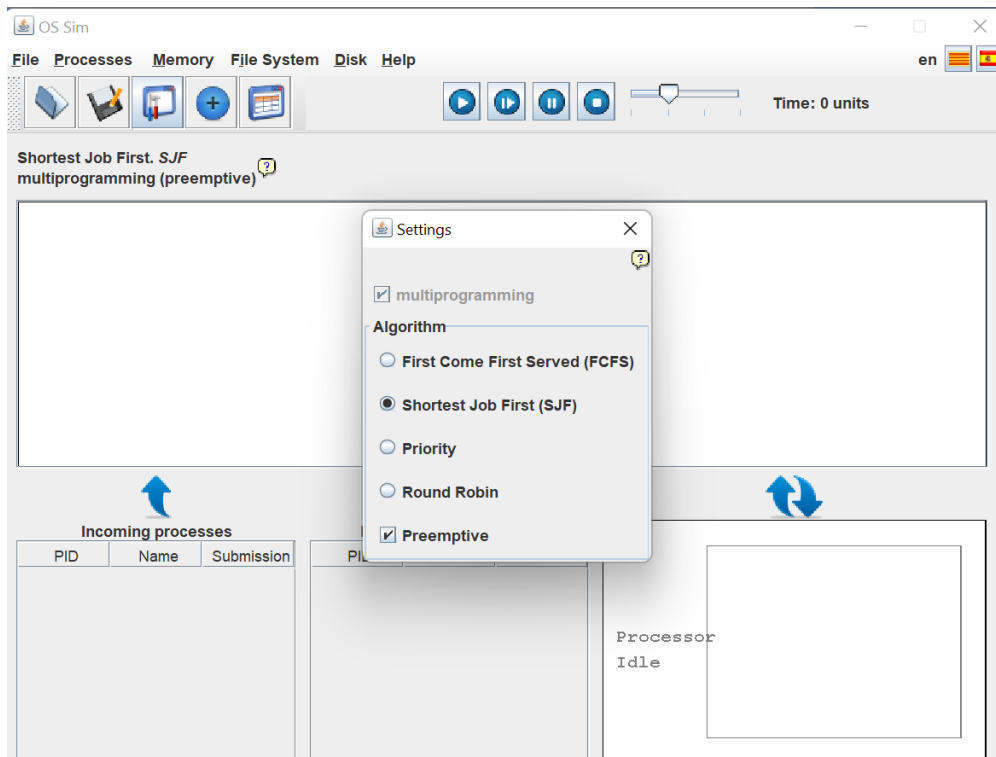
Process Scheduling Information										
Efficiency (%)		1.00								
Throughput (processes/time unit)		0.18								
Avg. Turnaround Time (time)		11.75								
Avg. Waiting Time (time)		6.25								
Avg. Response Time (time)		6.25								
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
6	P1	1	0	-	3	0	0	3	1.0	0.0
5	P0	1	0	-	5	3	3	8	0.625	0.0
8	P3	1	0	-	6	8	8	14	0.4285714...	0.0
7	P2	1	0	-	8	14	14	22	0.3636363...	0.0

c. Tabel :

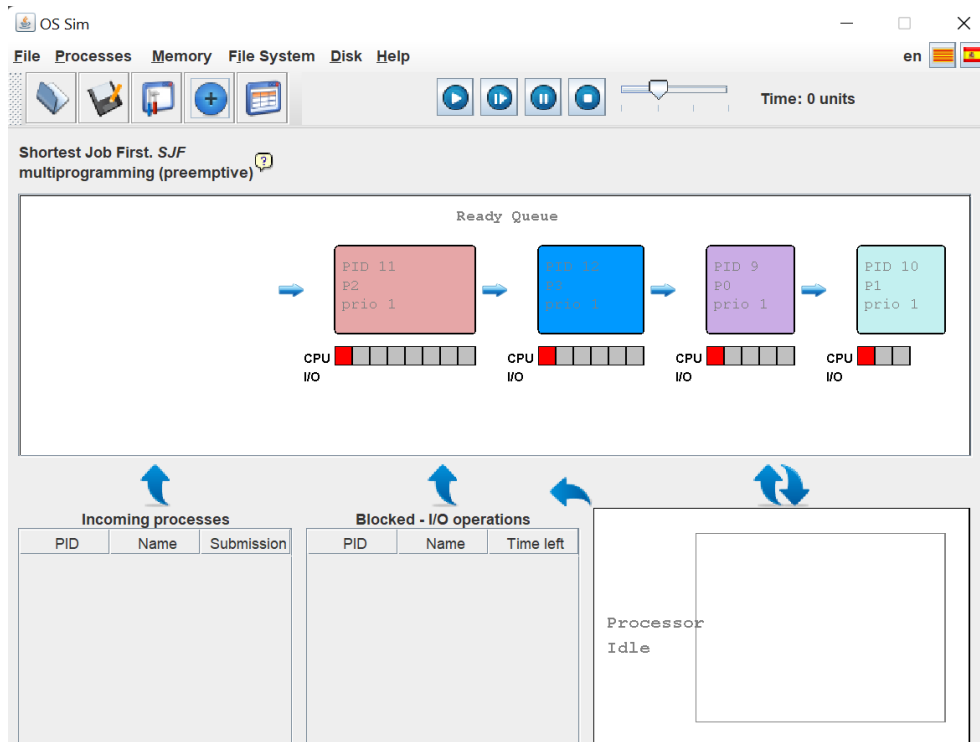
Process	Wait Time : Service Time – Arrival Time
P0	0
P1	3
P2	8
P3	14
Av Wait Time	6.25

- Pada algoritma ini, proses dengan burst time terkecil akan dieksekusi terlebih dahulu, sehingga akan dieksekusi berurutan sesuai dengan besar burst time yang dibutuhkan.

SJF Preemptive



a. Proses :



b. Hasil :

Process Scheduling Information										
Efficiency (%)		1.00								
Throughput (processes/time unit)		0.18								
Avg. Turnaround Time (time)		11.75								
Avg. Waiting Time (time)		6.25								
Avg. Response Time (time)		6.25								
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
10	P1	1	0	-	3	0	0	3	1.0	0.0
9	P0	1	0	-	5	3	3	8	0.625	0.0
12	P3	1	0	-	6	8	8	14	0.4285714...	0.0
11	P2	1	0	-	8	14	14	22	0.3636363...	0.0

c. Tabel :

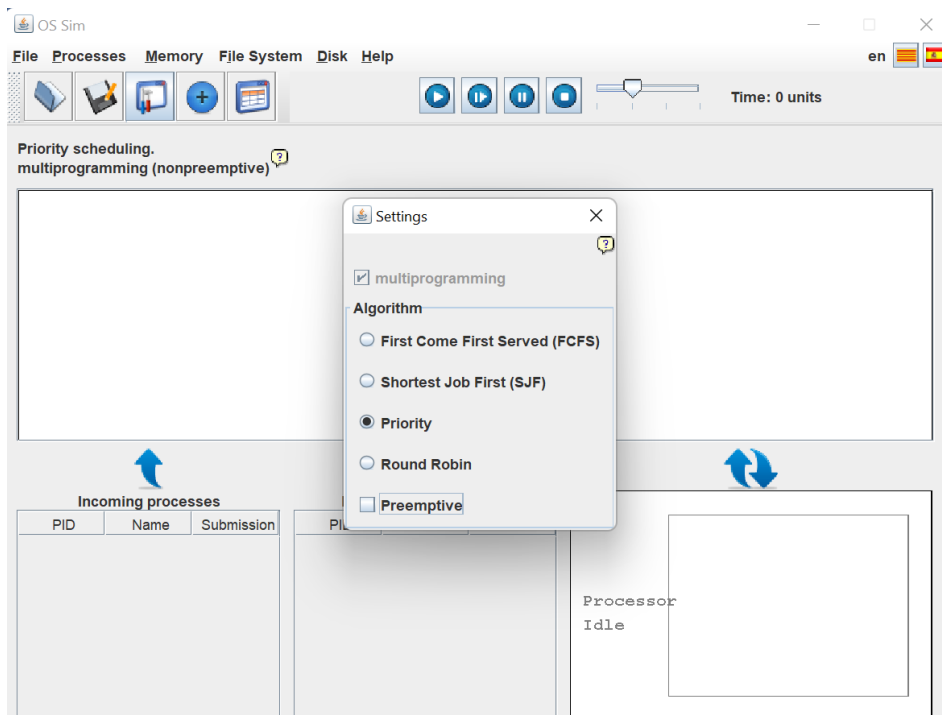
Process	Wait Time : Service Time – Arrival Time
P0	0
P1	3
P2	8
P3	14
Av Wait Time	6.25

- Pada algoritma ini, proses yang memiliki burst time paling sedikit akan dieksekusi terlebih dahulu, dan berlaku pada proses yang pertama kali datang (preemptive) sehingga rata-rata waktu tunggu akan kecil.

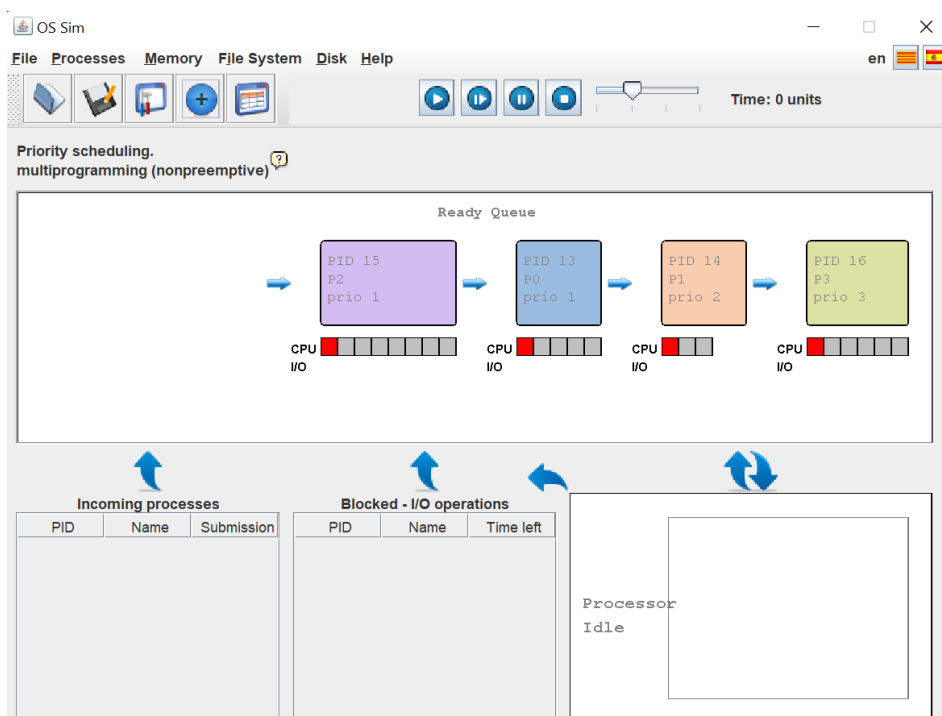
1.3 Priority

- Pilihlah menu setting dan pilih algoritma **Priority**. Selanjutnya tambahkan priority pada setiap proses.
- Selanjutnya klik tombol start. Lakukan pengamatan dan Analisa proses yang terjadi.

➤ Hasil Kegiatan : Priority Non-Preemptive



- Proses :



b. Hasil :

Process Scheduling Information											
Efficiency (%)		1.00									
Throughput (processes/time unit)		0.18									
Avg. Turnaround Time (time)		12.75									
Avg. Waiting Time (time)		7.25									
Avg. Response Time (time)		7.25									
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO	
16	P3	3	0	-	6	0	0	6	1.0	0.0	
14	P1	2	0	-	3	6	6	9	0.333333...	0.0	
13	P0	1	0	-	5	9	9	14	0.3571428...	0.0	
15	P2	1	0	-	8	14	14	22	0.3636363...	0.0	

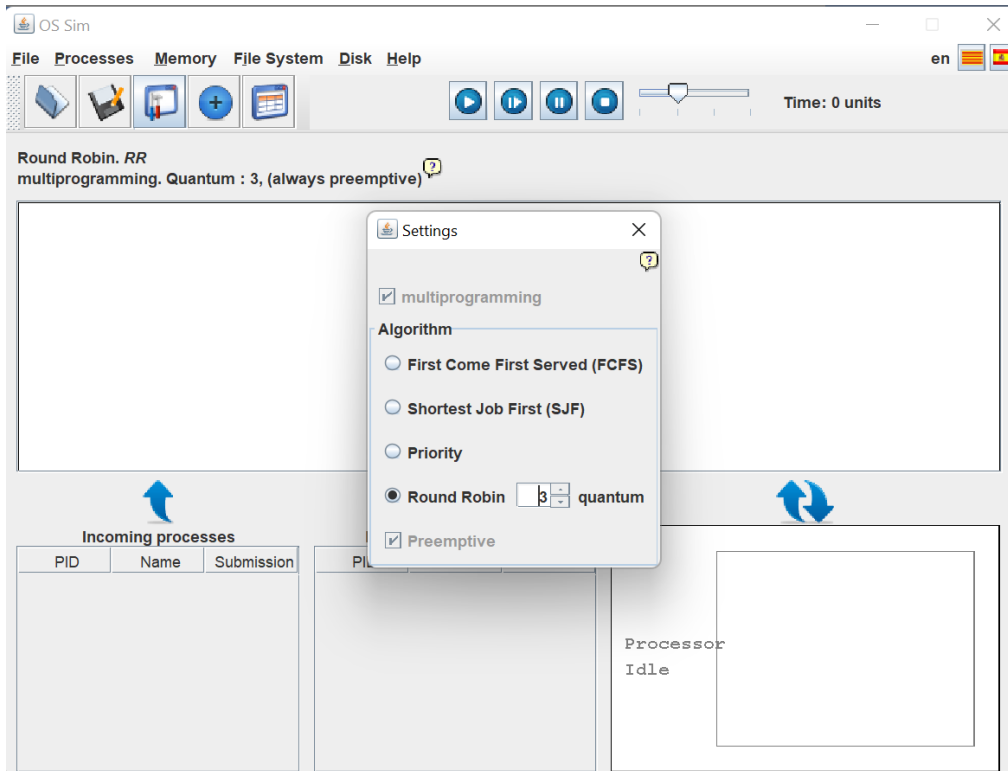
c. Tabel :

Process	Wait Time : Service Time – Arrival Time
P0	0
P1	6
P2	9
P3	14
Av Wait Time	7.25

- Pada algoritma ini, proses akan dieksekusi sesuai dengan prioritasnya tanpa memandang burst time yang dibutuhkan.

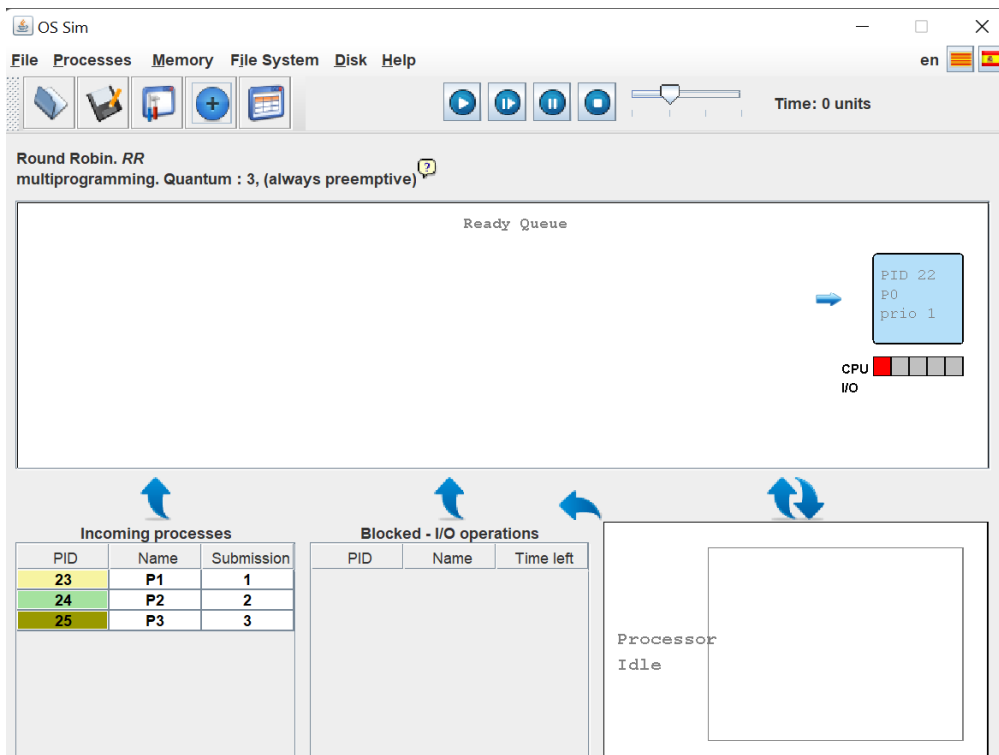
1.4 Round Robin

- Pilihlah menu setting dan pilih algoritma **Round Robin**. Selanjutnya tambahkan quantum time sebesar 3.
- Selanjutnya klik tombol start. Lakukan pengamatan dan Analisa proses yang terjadi.



➤ Hasil Kegiatan :

- Proses :



b. Hasil :

Process Scheduling Information										
Efficiency (%)		1.00								
Throughput (processes/time unit)		0.18								
Avg. Turnaround Time (time)		14.00								
Avg. Waiting Time (time)		8.50								
Avg. Response Time (time)		3.00								
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
23	P1	2	1	-	3	2	2	5	0.6	0.0
22	P0	1	0	-	5	0	9	14	0.3571428...	0.0
25	P3	3	3	-	6	6	11	17	0.3529411...	0.0
24	P2	1	2	-	8	4	12	20	0.4	0.0

c. Tabel :

Process	Wait Time : Service Time – Arrival Time
P0	9
P1	2
P2	12
P3	11
Av Wait Time	8.50

- Pada algoritma ini, semua proses memiliki jatah waktu eksekusinya sehingga akan adil tetapi dikarenakan memiliki jatah eksekusi dan terjadi banyak switch saat waktu jatahnya habis, rata-rata waktu tunggu menjadi semakin besar.

Kegiatan 2: Manajemen Memori

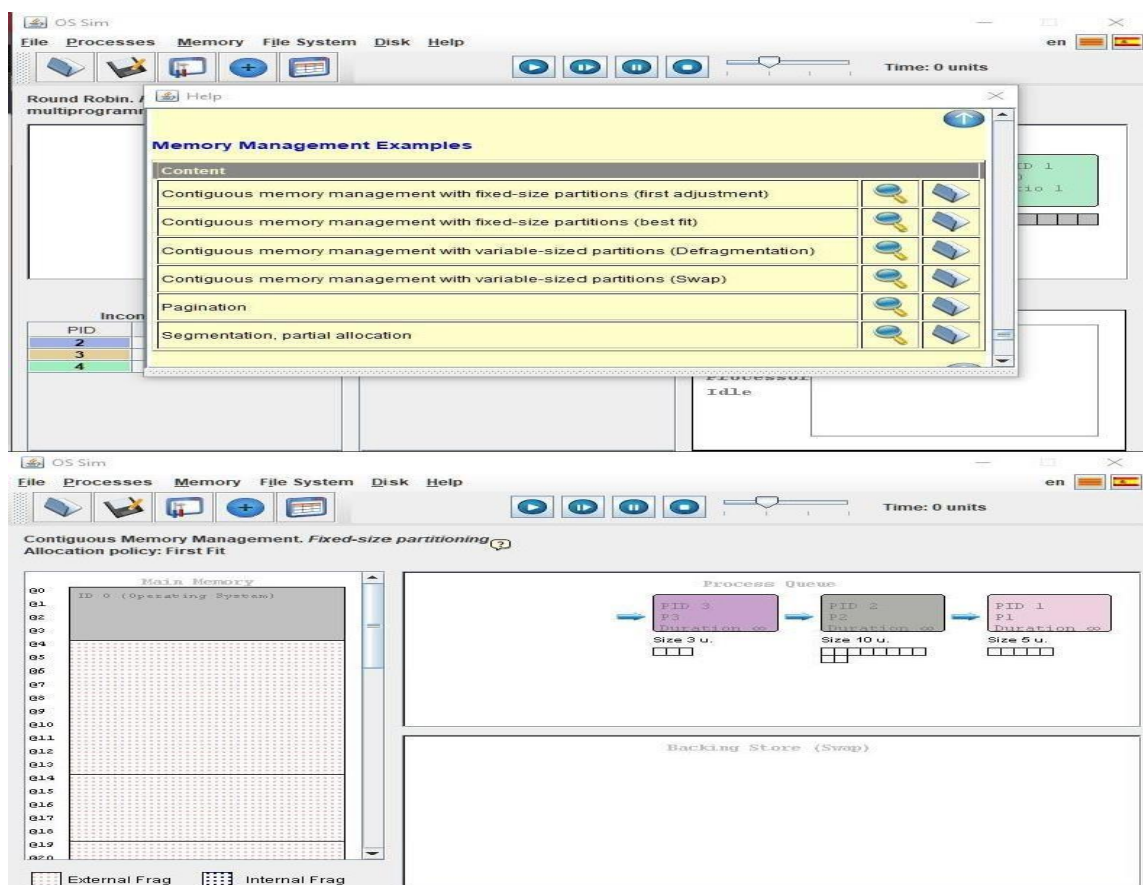
2.1 Contiguous memory management dengan menggunakan partisi berukuran tetap (fixed-size partition) dan aturan first fit

Memori dibagi menjadi partisi-partisi berukuran tetap, lalu seluruh proses akan dialokasikan pada satu dari partisi yang tersedia. Pada aturan first fit, partisi pertama yang dapat memuat proses akan dipilih.

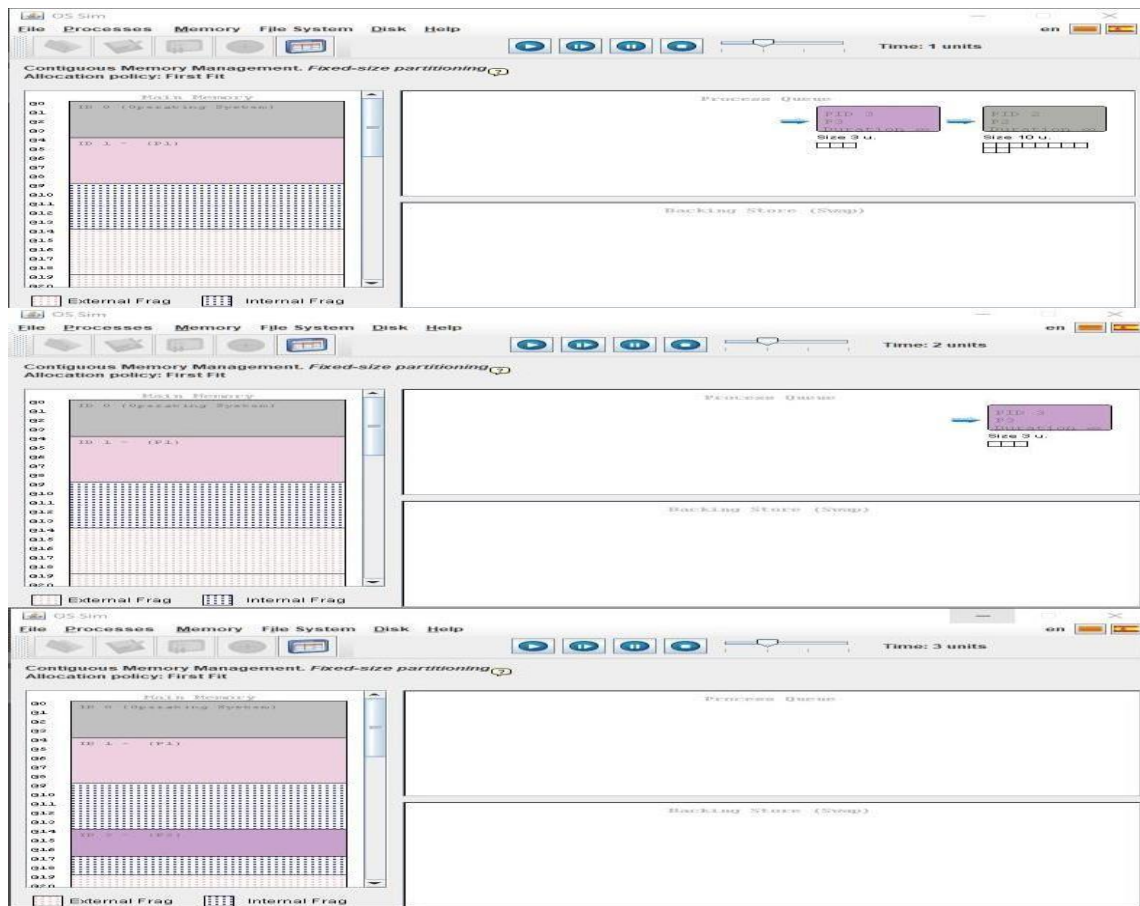
Partisi		Proses	
Alamat awal	Unit ukuran	Proses	Unit ukuran
@4	10	P1	5
@14	5	P2	10
@19	4	P3	3
@23	15		
@38	10		
@48	14		

Langkah :

1. Pilih menu help >> examples...>> memory management >> Contiguous memory management with fixed-size partitions (first adjustment). Pilihlah folder hingga menampilkan tampilan berikut.



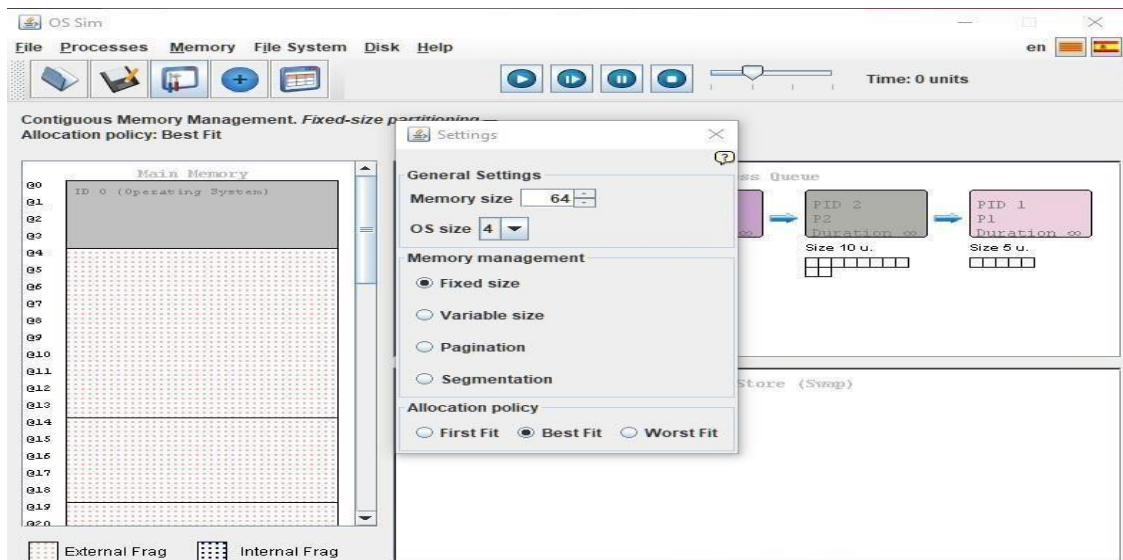
2. Klik Step By Step :



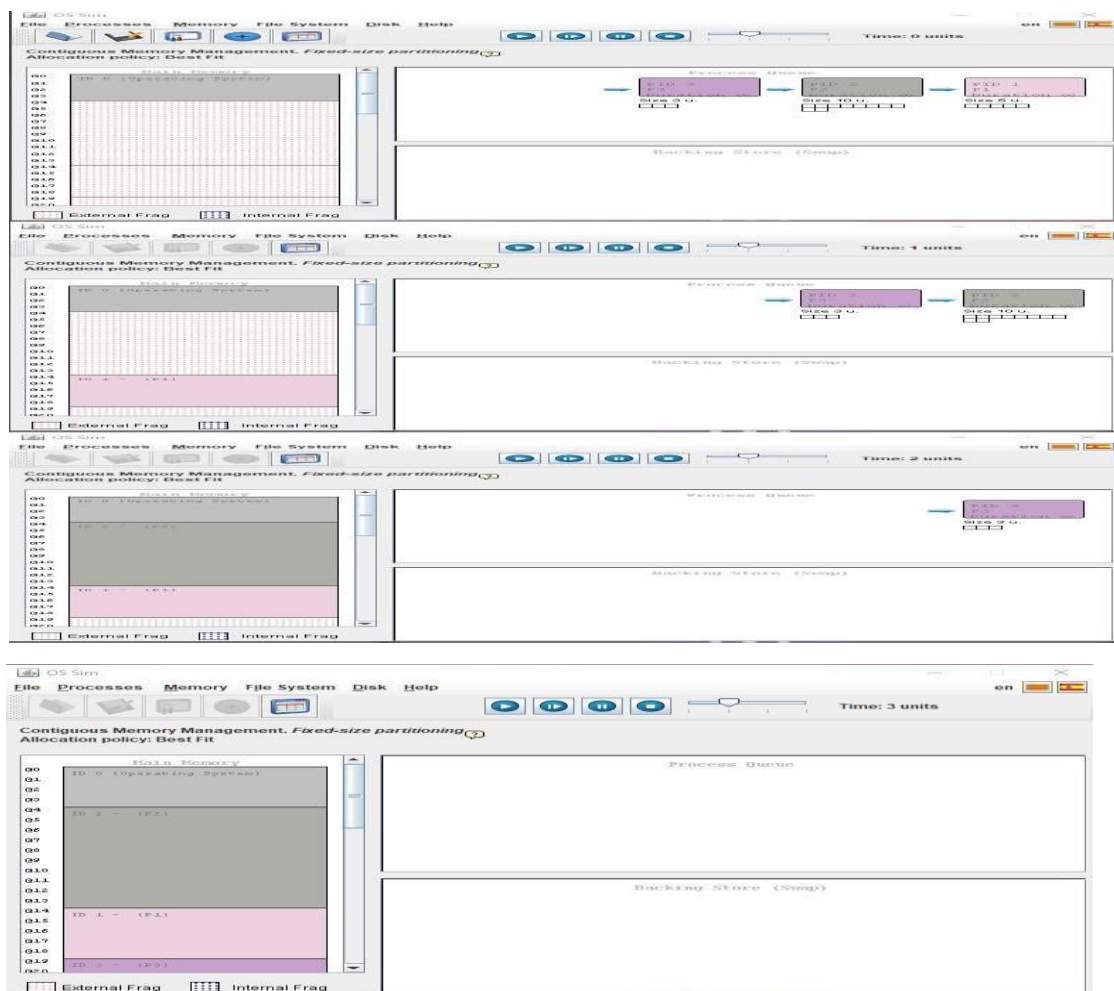
- Dengan aturan first fit, banyak terjadi internal fragmentasi. Ini terjadi karena ketika suatu proses menemukan partisi yang cukup untuk dimuat, maka ia akan dimuat di partisi tersebut tanpa memandang apakah sisa partisi banyak atau tidak.

2.2 Contiguous Memory Management dengan menggunakan partisi berukuran tetap (fixed-size partition) dan aturan best fit

1. Mengganti ke Contiguous Memory Management fixed-size partition with best fit



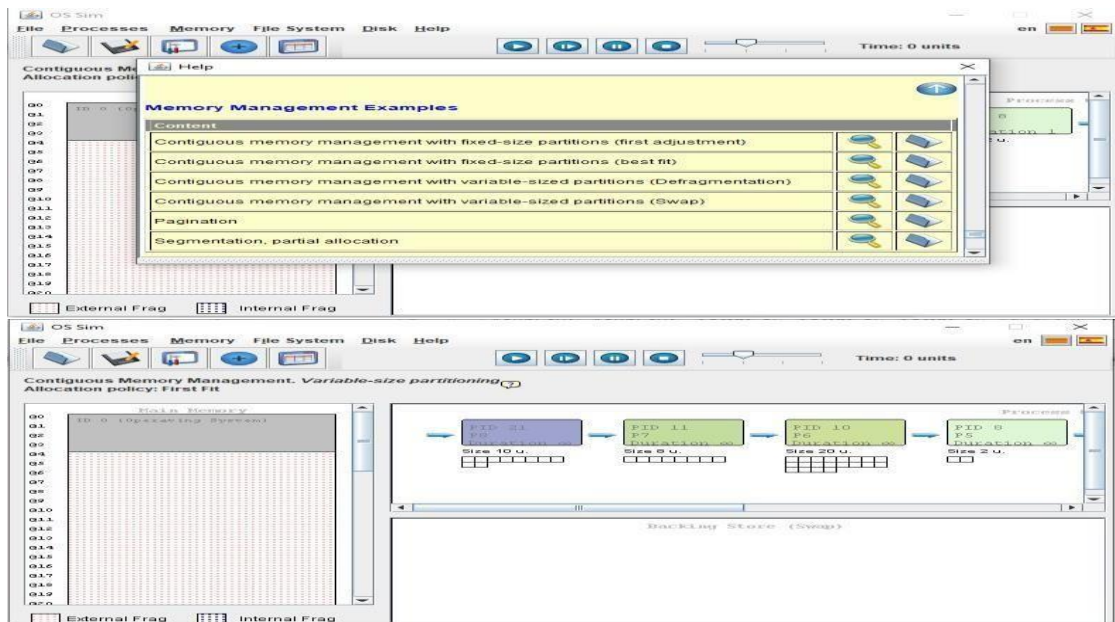
2. Klik Step By Step



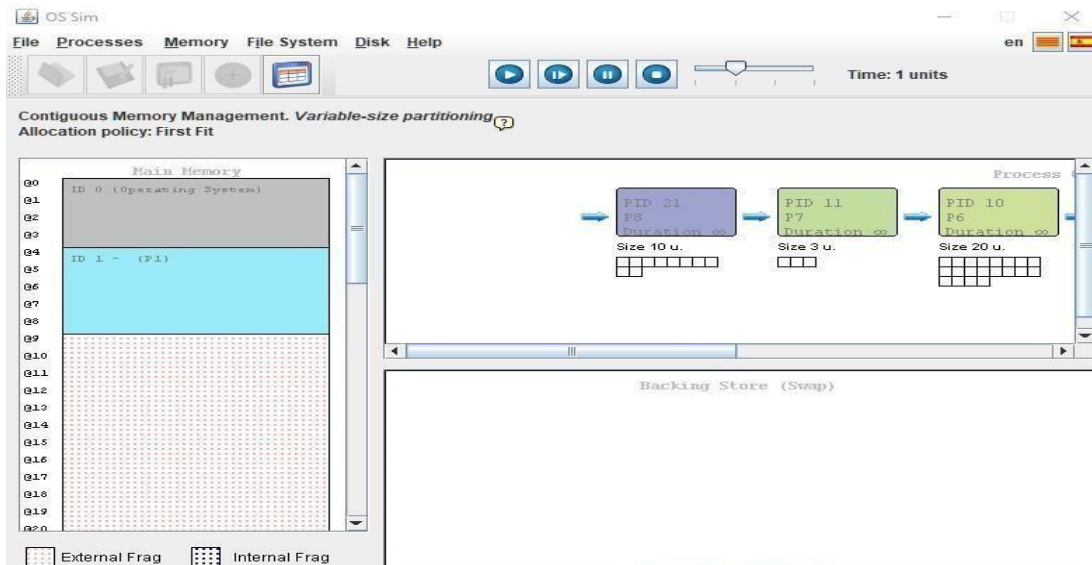
- Dengan menggunakan aturan best fit, internal fragmentasi berkurang. Ini terjadi karena proses dimuat pada ukuran partisi terkecil akan tetapi masih dapat memuat proses tersebut sehingga internal fragmentasi dapat diminimalisasi.

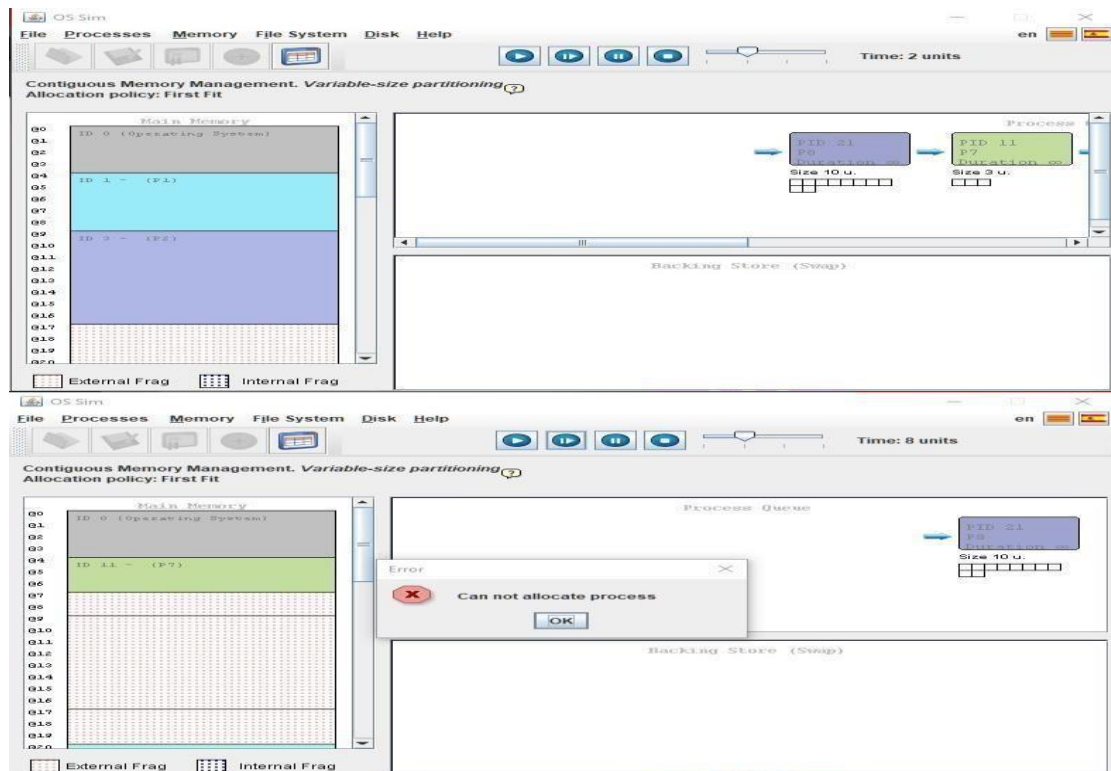
2.3 Contiguous Memory Management dengan menggunakan partisi berukuran tidak tetap (variable-size partition) dengan defragmentasi

1. Mengubah manajemen memori di bagian help lalu management memory lalu yang defragmentasi



2. Klik Step By Step

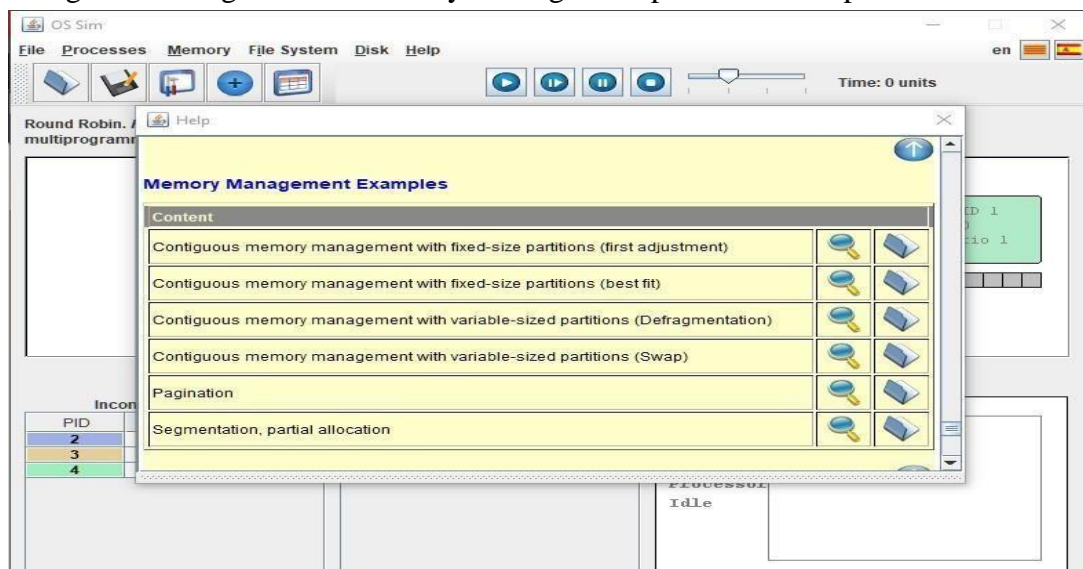




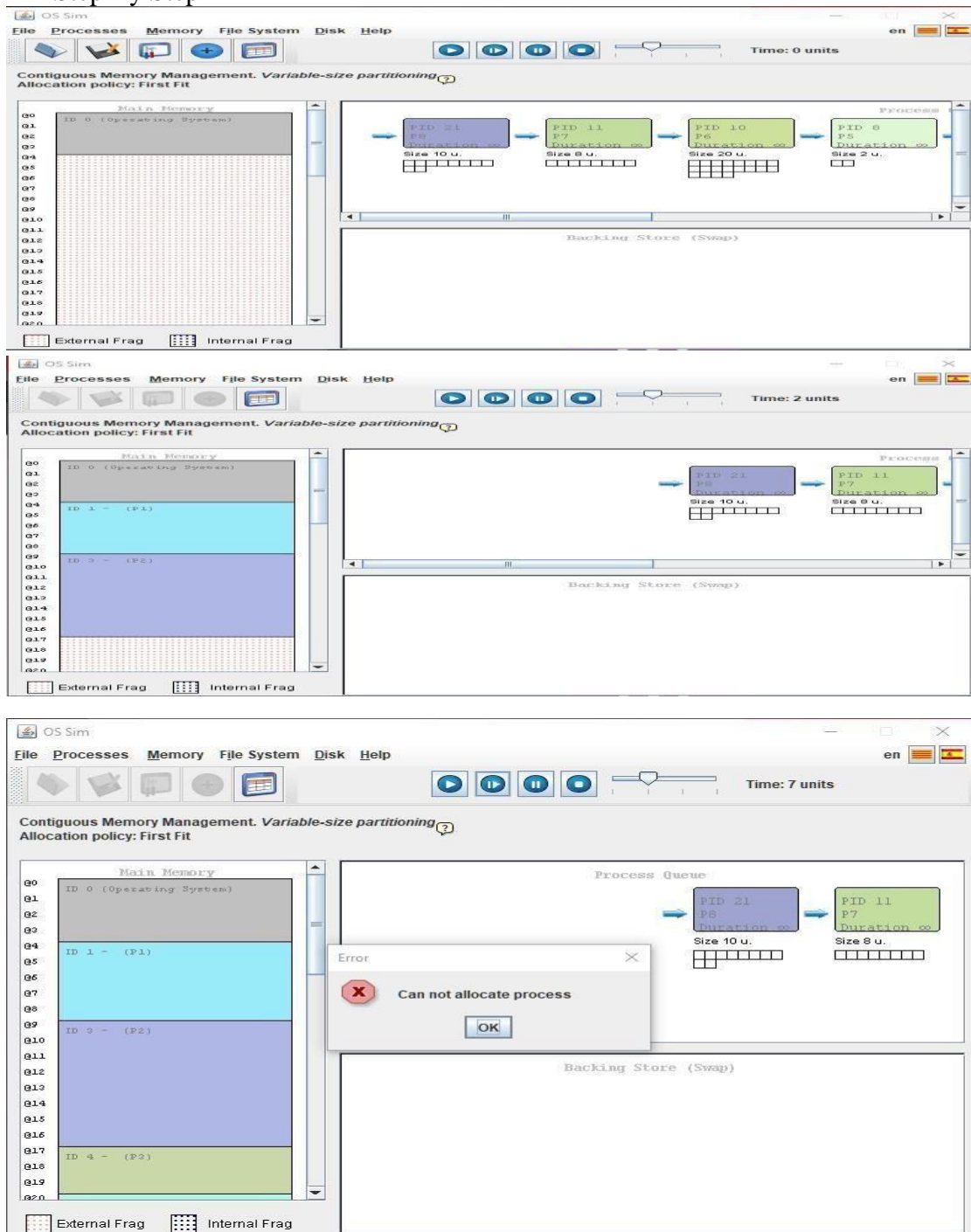
- Pada langkah terakhir terdapat sebuah proses yang tidak dapat dimuat. Ini terjadi karena pembuatan partisi berdasarkan proses yang dimuat pertama kali tetapi tidak ada partisi yang dapat memuat proses terakhir tersebut.

2.4 Contiguous Memory Management dengan menggunakan partisi berukuran tidak tetap (variable-size partition) dengan swap

1. Mengubah management memori nya di bagian help lalu klik Swap



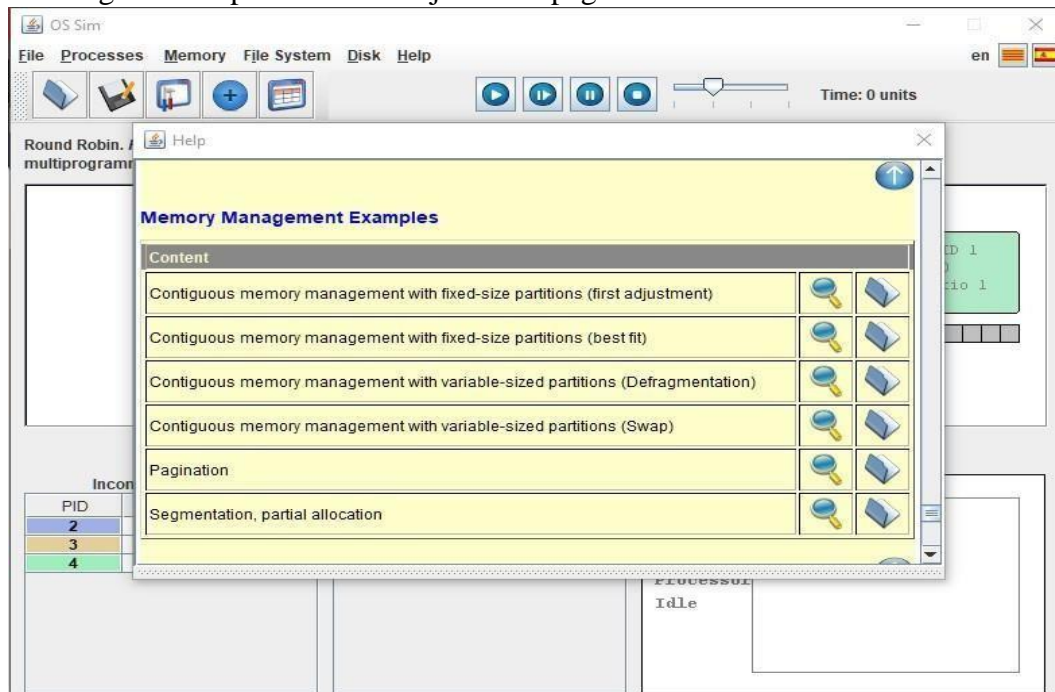
2. Klik Step By Step



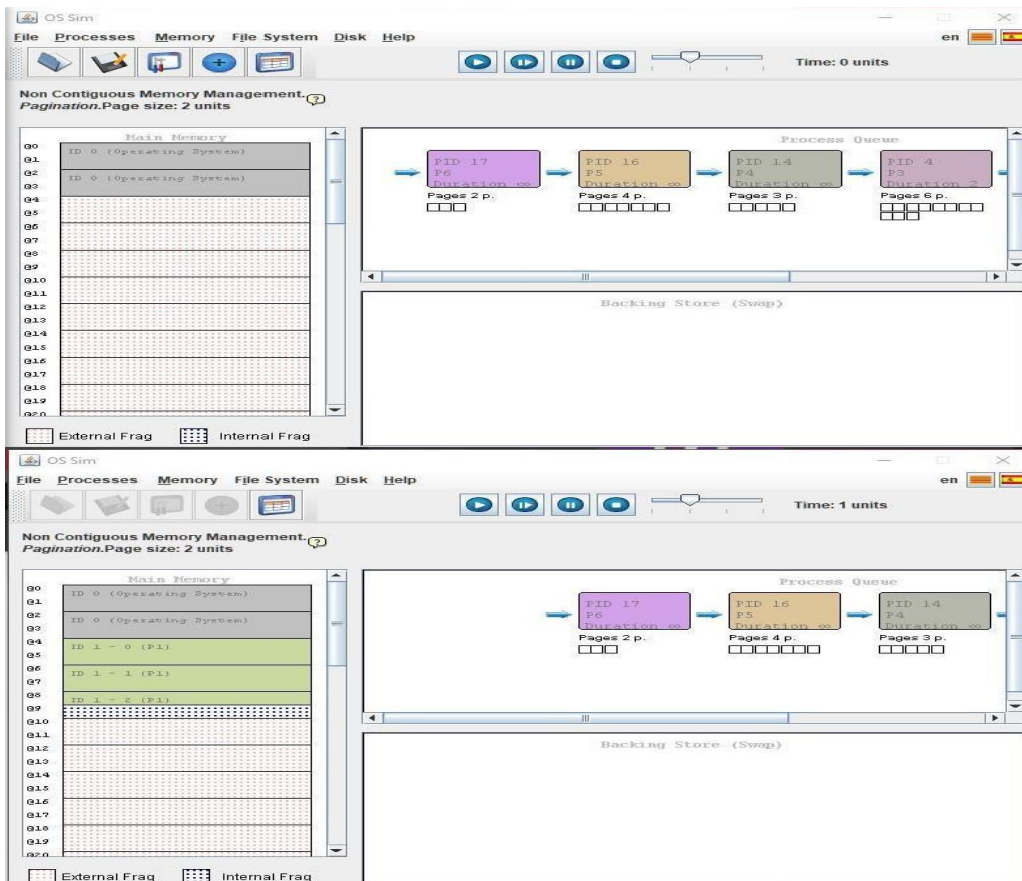
- Sampai langkah ke 7 akan ada notifikasi Can not allocate process. Hal ini mungkin terjadi karena tidak adanya proses yang tidak aktif sehingga tidak terjadi swap dan partisi tidak ada yang mencukupi untuk memuat proses yang tersisa.

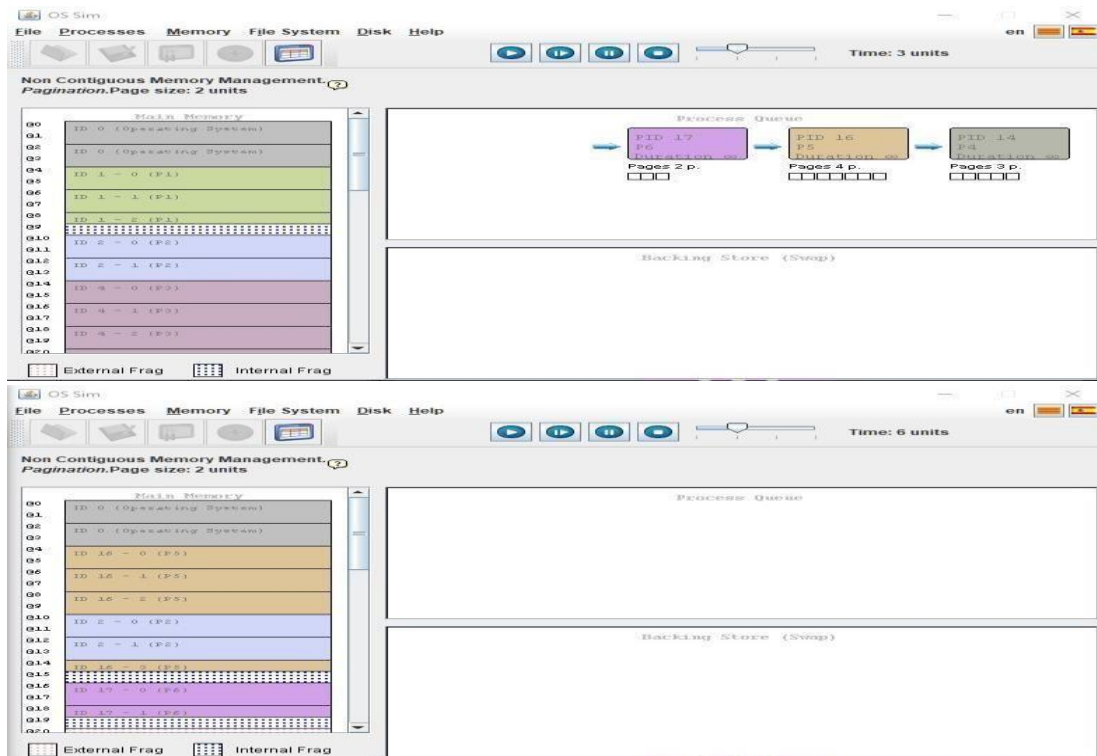
2.5 Pagination (ukuran page 2 unit)

1. Mengubah Help memori manajemen di pagination



2. Klik Step By Step

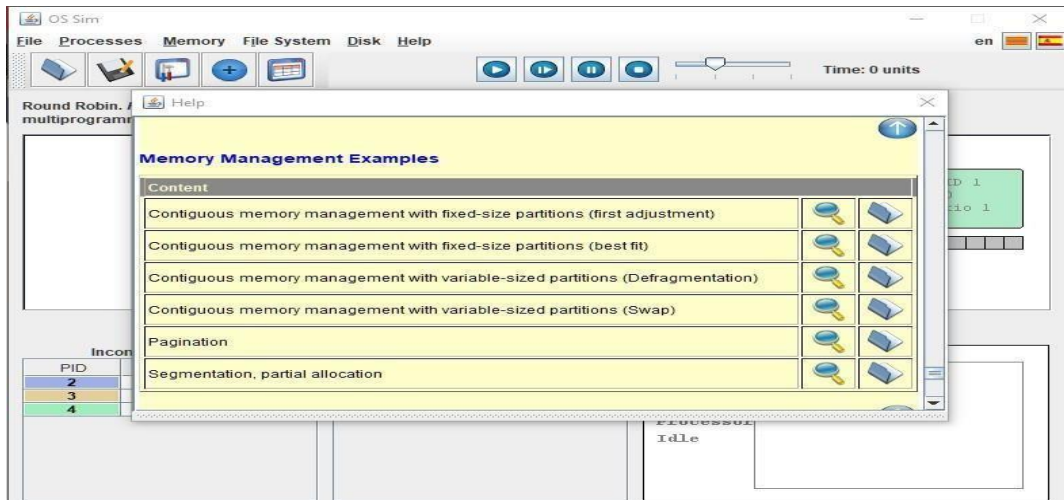




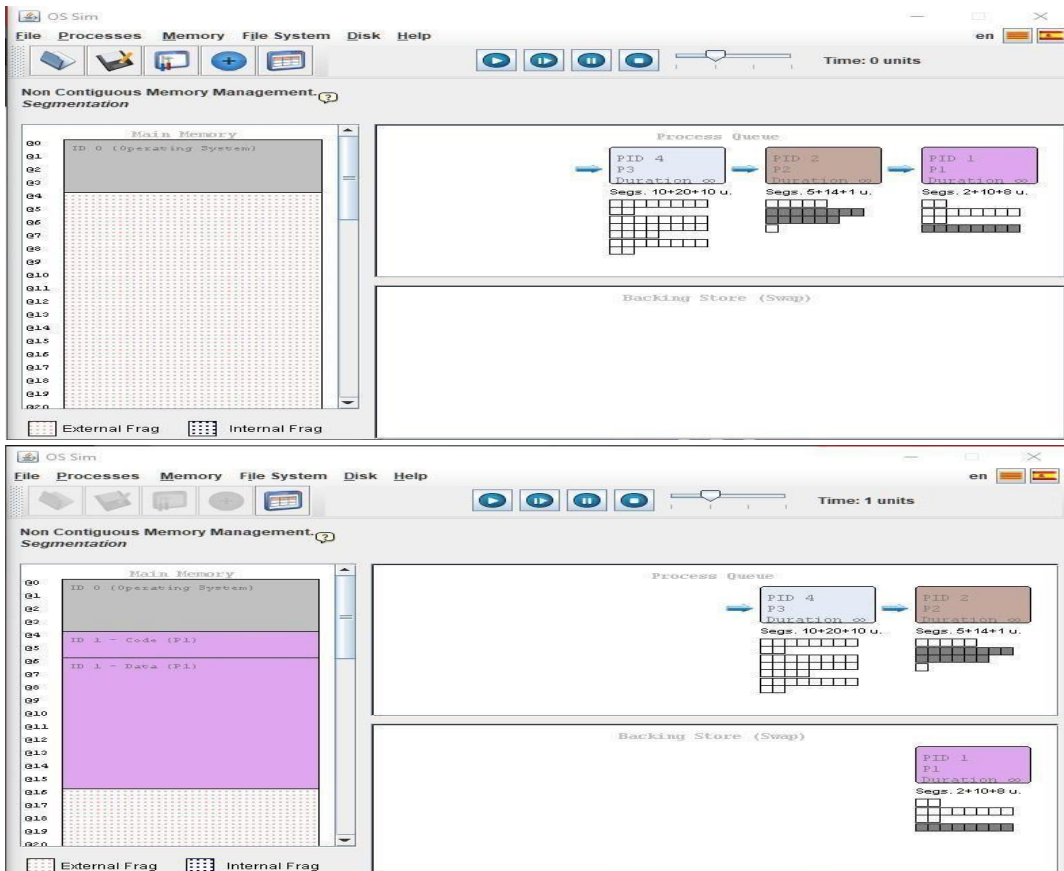
- Terlihat pada percobaan ini bahwa setiap proses dibagi menjadi beberapa page dengan ukuran tetap setiap pagennya, yaitu 2 unit

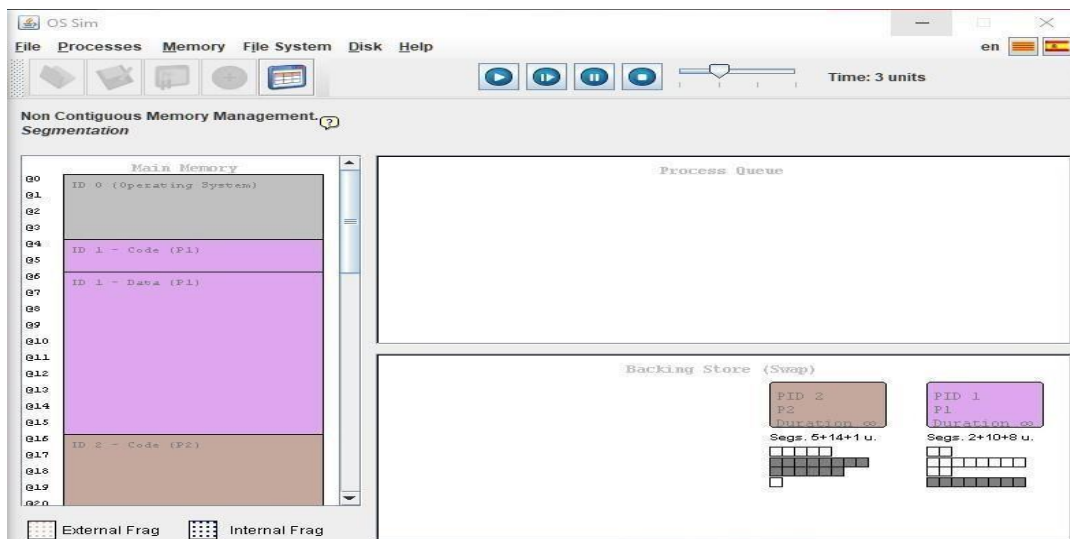
2.6 Segmentation (alokasi parsial)

1. Mengubah manajemen memori di help lalu example, dan setelah itu segmentation



2. Klik Step By Step





- Semua proses dapat dimuat, karena setiap proses dibagi menjadi beberapa segmen berdasarkan logical structure-nya. Dan karena tidak semua segmen proses tersebut dimuat di memori, semua proses dapat dimuat, dan yang segmen yang tidak dimuat tadi dapat di swap-out.