PRAKTIKUM SISTEM OPERASI

MODUL 11

Nama : NICKY JULYATRIKA SARI

NIM : L200200101

PROGRAM STUDI

INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

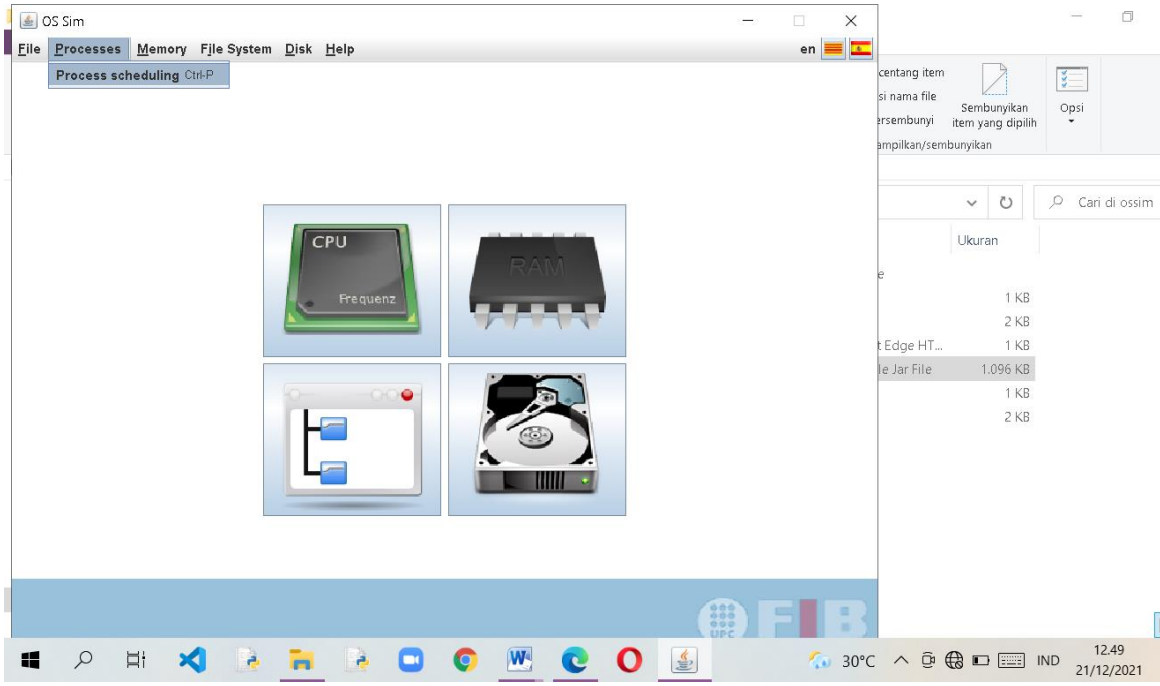
UNIVERSITAS MUHAMMADIYAH SURAKARTA

TAHUN 2021/2022

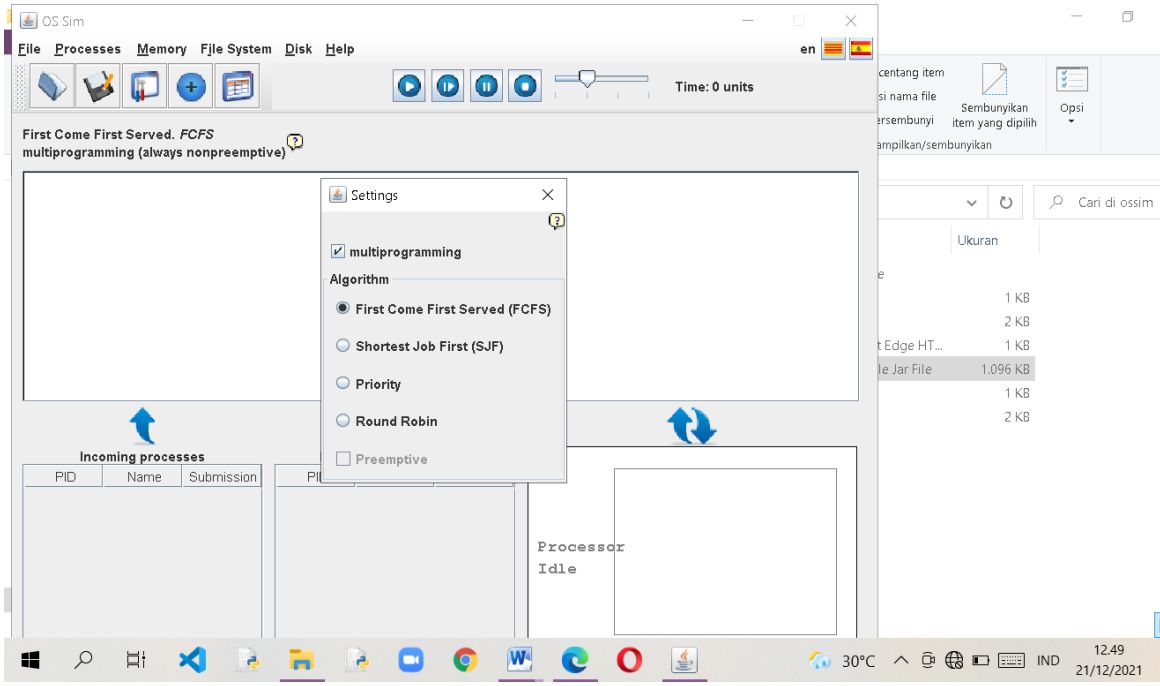
1. Kegiatan 1. Penjadwalan Proses

1.1First-Come, First-Served (FCFS)

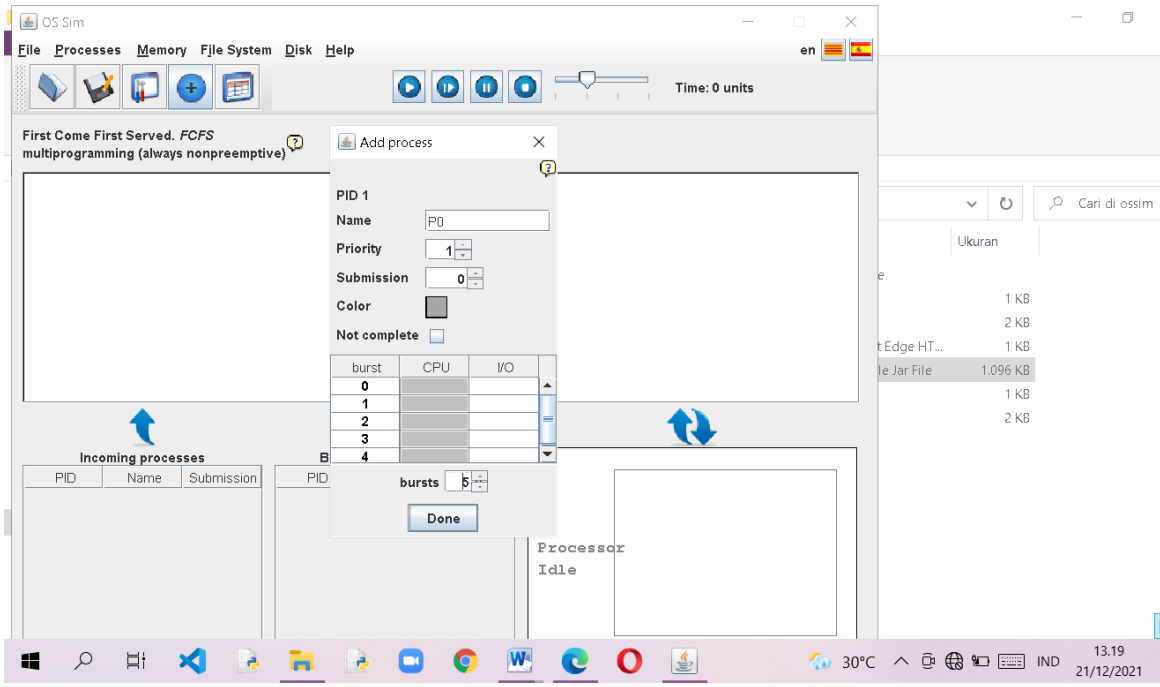
a. Bukalah program OSSim, selanjutnya pilih menu processes -> process scheduling.

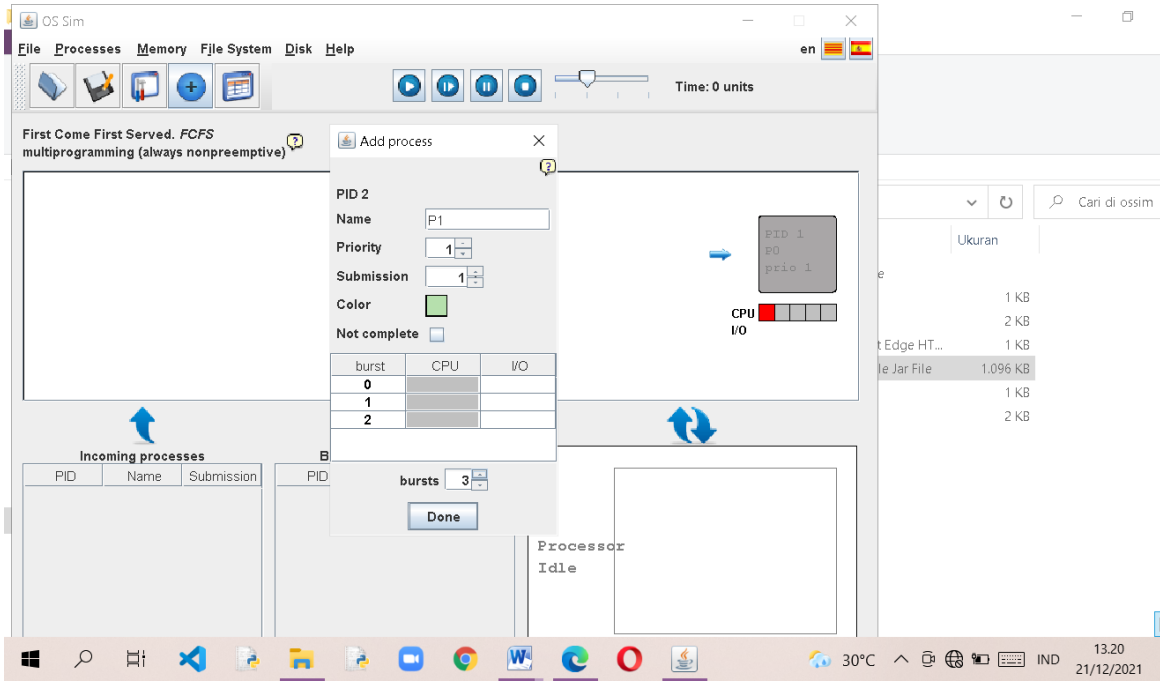


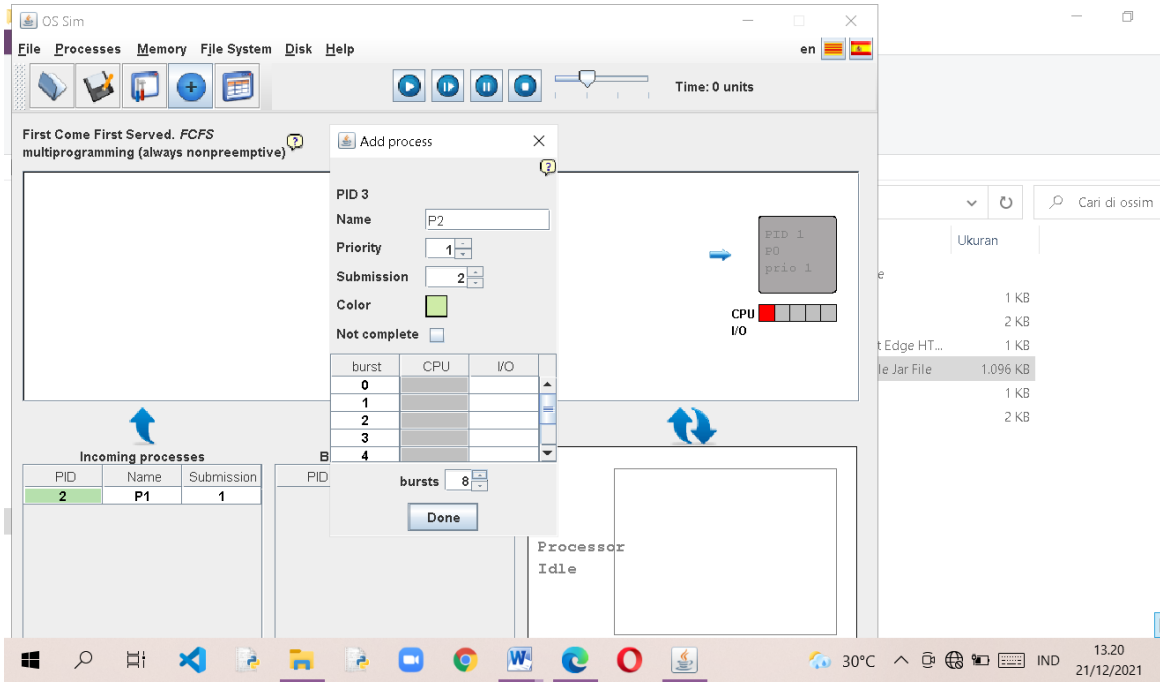
b. Selanjutnya pilihlah setting dan pilih algoritma First-Come, First-Served (FCFS).

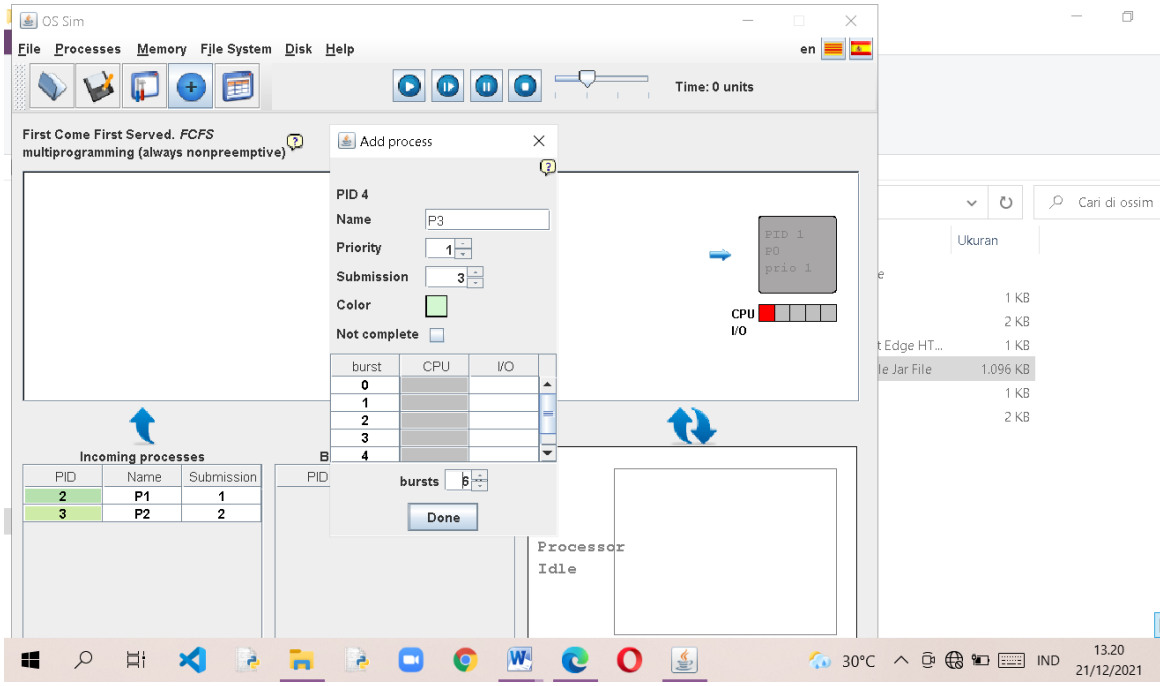


c. Lakukan input proses sesuai dengan tabel dengan memulai dengan P0 sebagai input proses yang pertama.

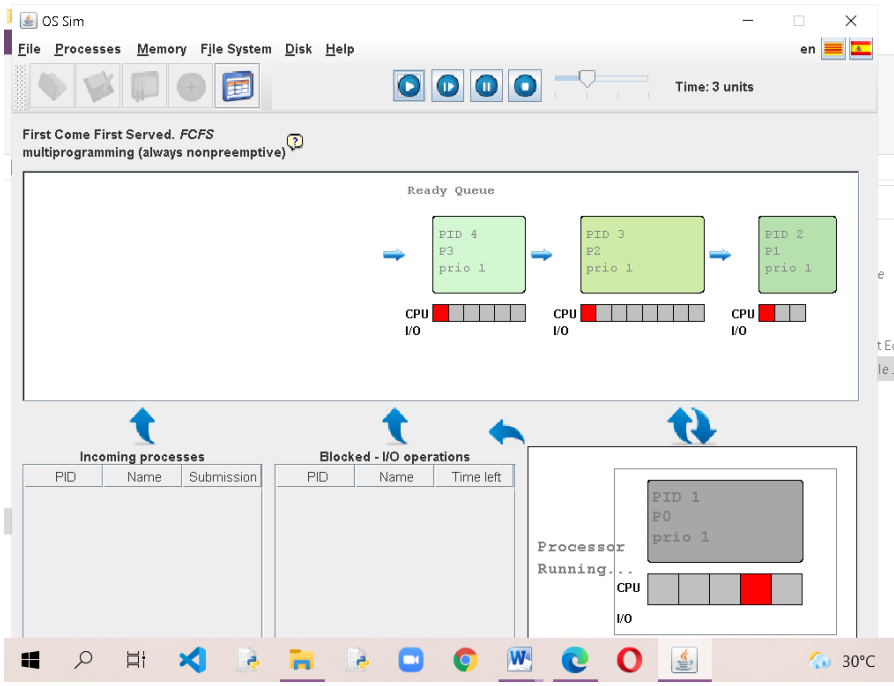


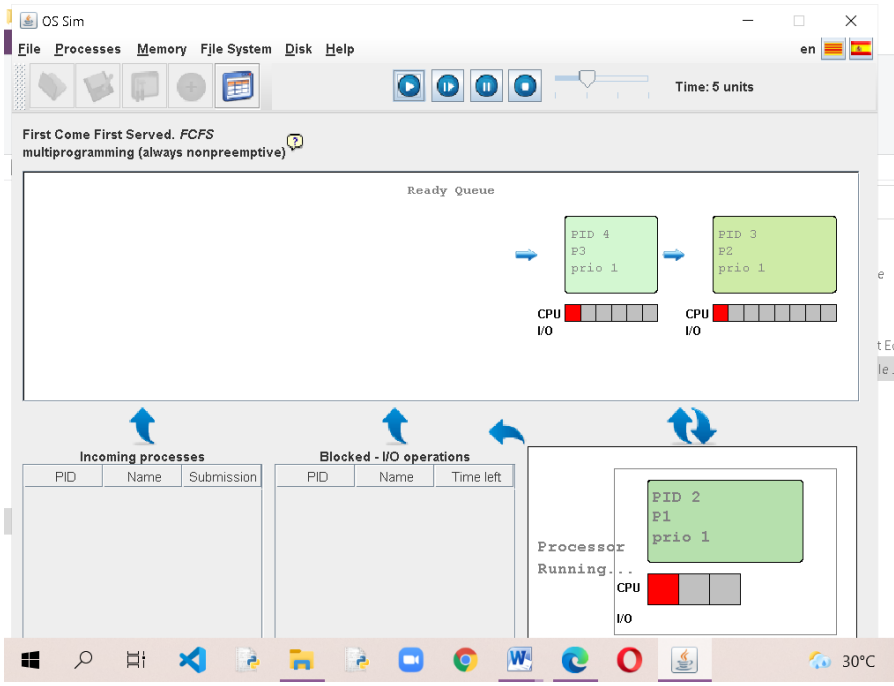


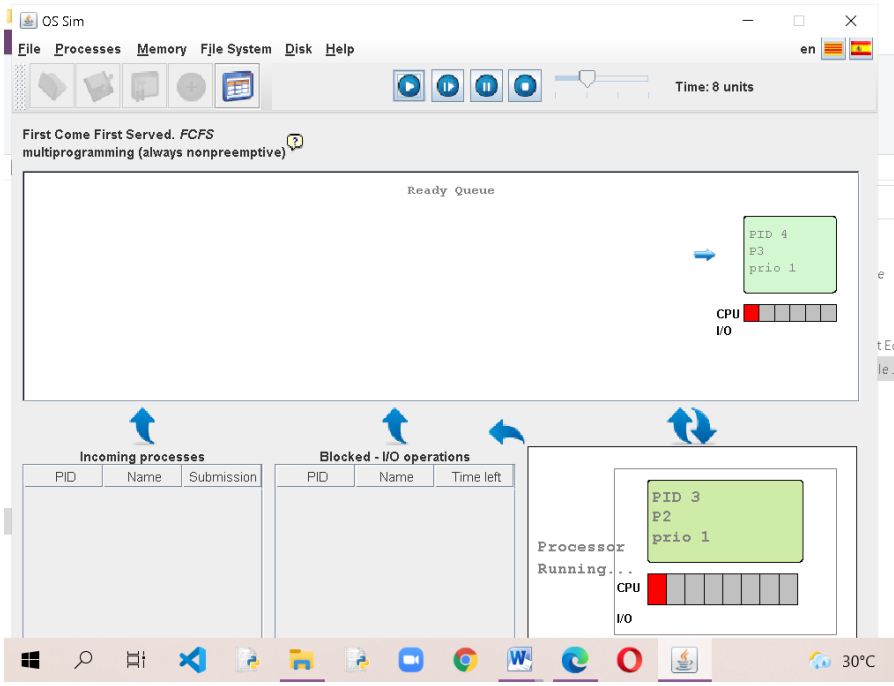


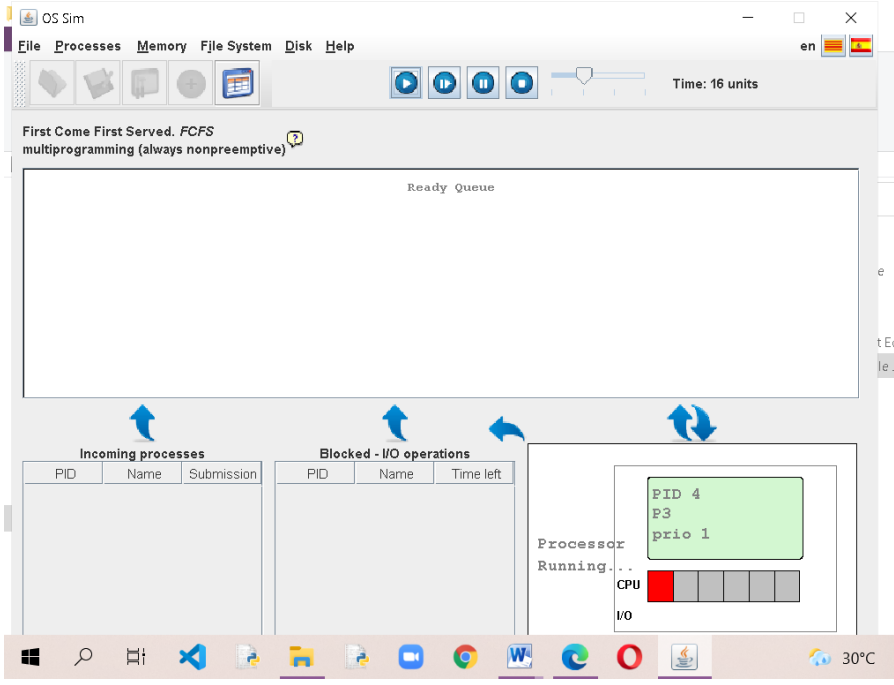


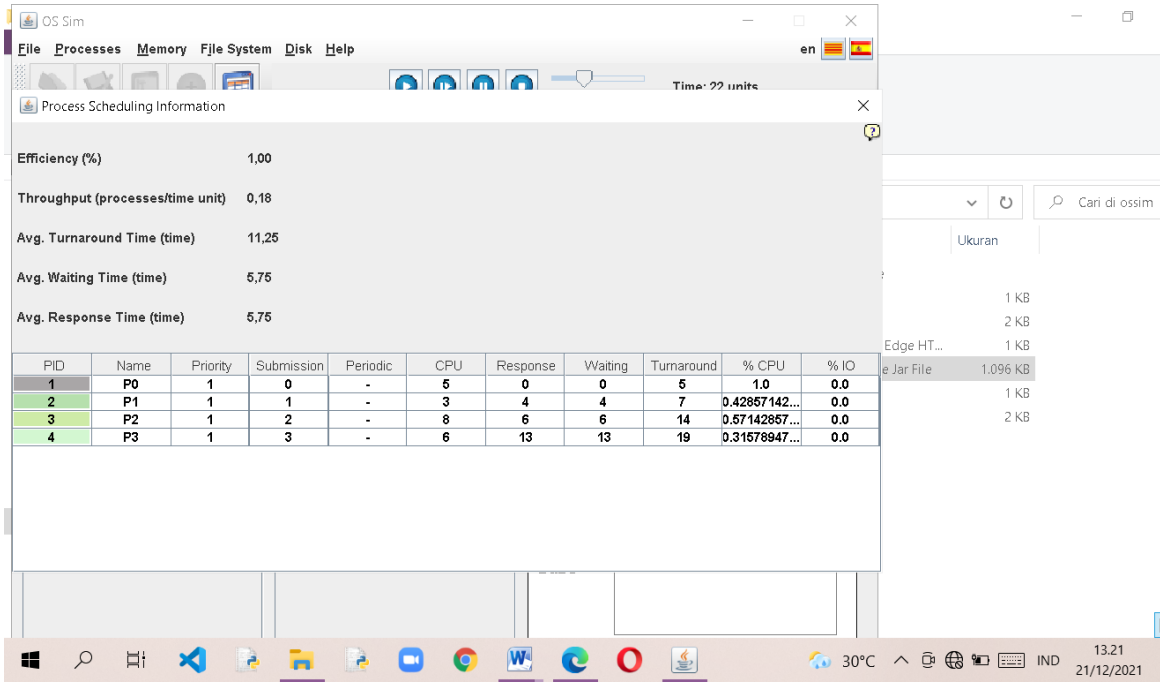
d. Memulai proses.


OS Sim
File Processes Memory File System Disk Help
en
Time: 3 units
First Come First Served. FCFS
multiprogramming (always nonpreemptive)
Ready Queue
PID 4
P3
prio 1
PID 3
P2
prio 1
PID 2
P1
prio 1
CPU
I/O
Incoming processes
PID
Name
Submission
Blocked - I/O operations
PID
Name
Time left
Processor
Running...
PID 1
P0
prio 1
30°C
IND
13.20
21/12/2021


OS Sim
File Processes Memory File System Disk Help
en
Time: 5 units
First Come First Served. FCFS
multiprogramming (always nonpreemptive)
Ready Queue
PID 4
P3
prio 1
PID 3
P2
prio 1
CPU
I/O
Incoming processes
PID
Name
Submission
Blocked - I/O operations
PID
Name
Time left
Processor
Running...
PID 2
P1
prio 1
30°C
IND
13.20
21/12/2021


OS Sim
File Processes Memory File System Disk Help
en
Time: 8 units
First Come First Served. FCFS
multiprogramming (always nonpreemptive)
Ready Queue
PID 4
P3
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
PID 3
P2
prio 1
Processor
Running...
CPU
I/O
30°C
IND
13.20
21/12/2021


OS Sim
File Processes Memory File System Disk Help
en
Time: 16 units
First Come First Served. FCFS
multiprogramming (always nonpreemptive)
Ready Queue
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
PID 4
P3
prio 1
Processor
Running...
CPU
I/O
30°C
IND
13.21
21/12/2021



e. Mengisi Tabel.

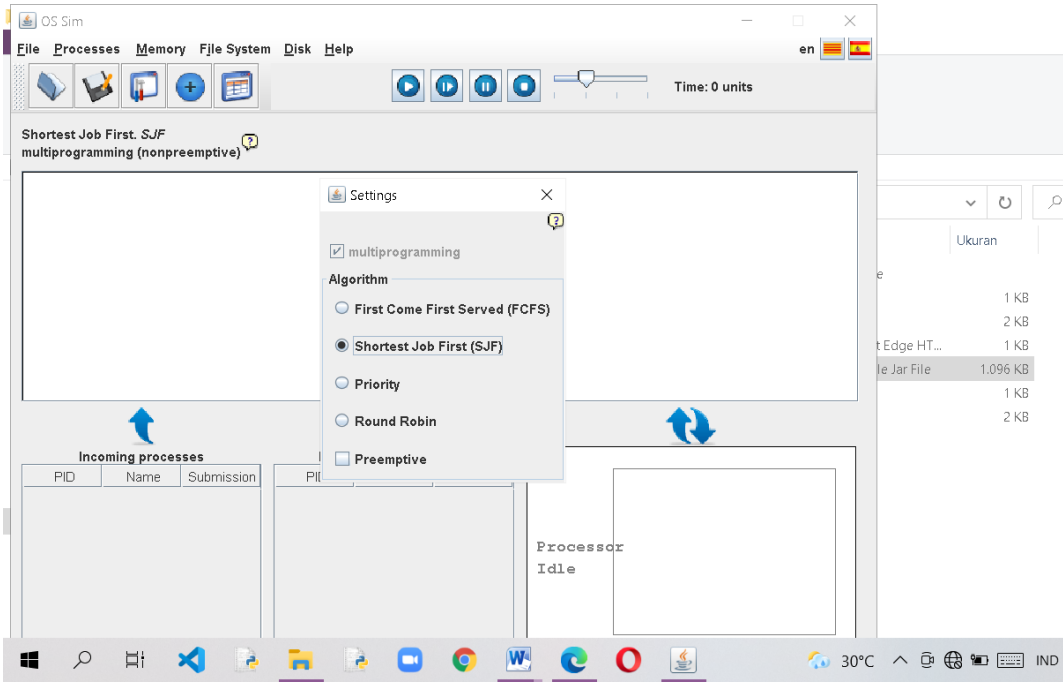
Process	Wait Time : Service Time – Arrival Time
P0	0
P1	4
P2	16
P3	13
Av Wait Time	5.75

f. Pada algoritma ini, proses di eksekusi sesuai dengan urutan datangnya tanpa memandang burst time yang dibutuhkan oleh proses tersebut.

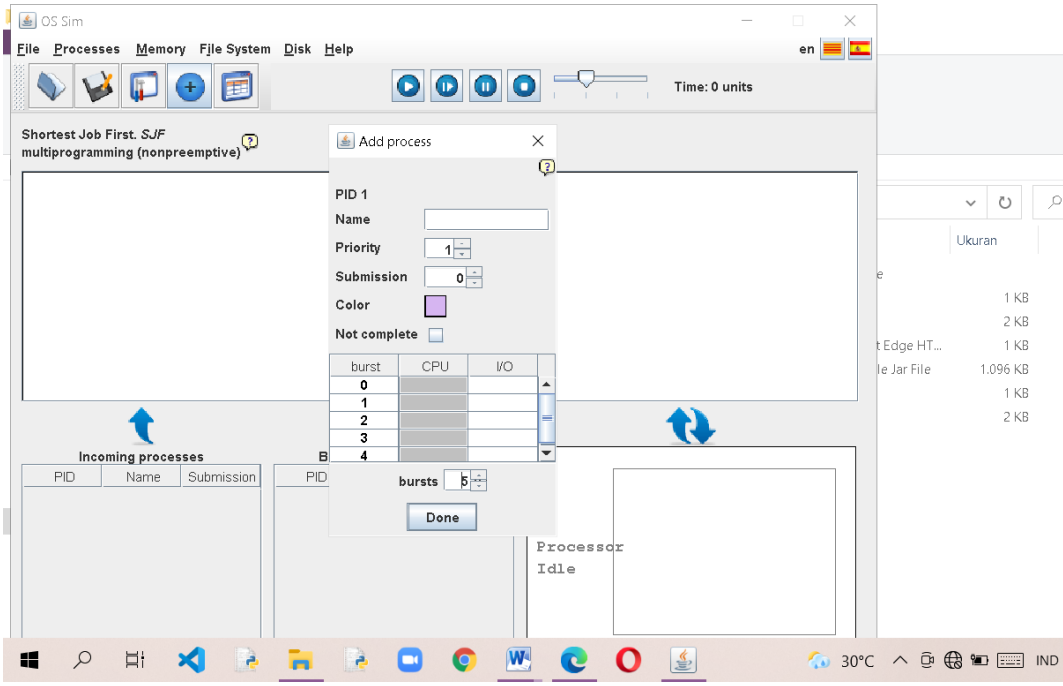
1.2Shortest Job First (SJF)

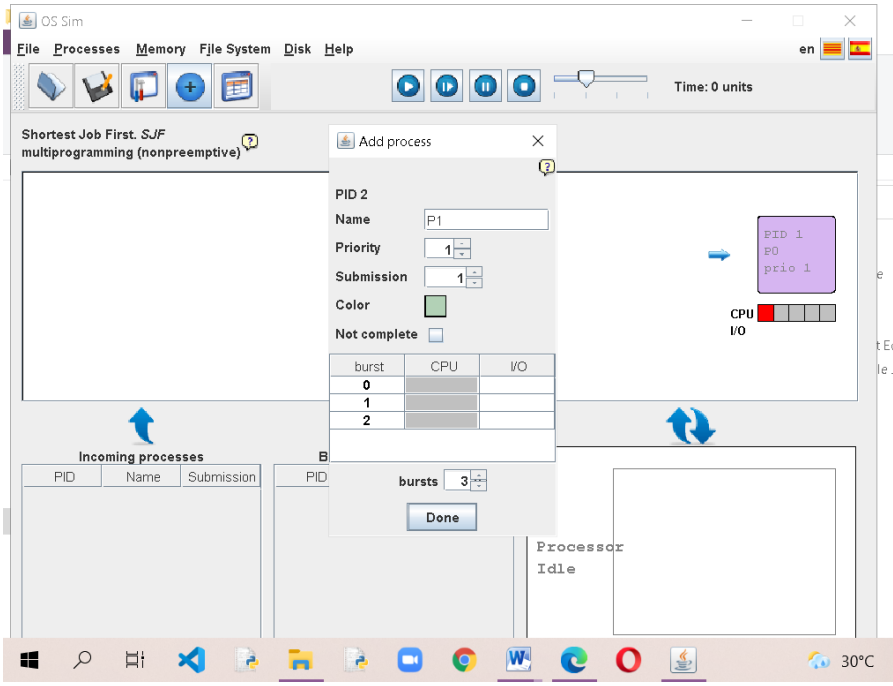
a. Non-preemtive

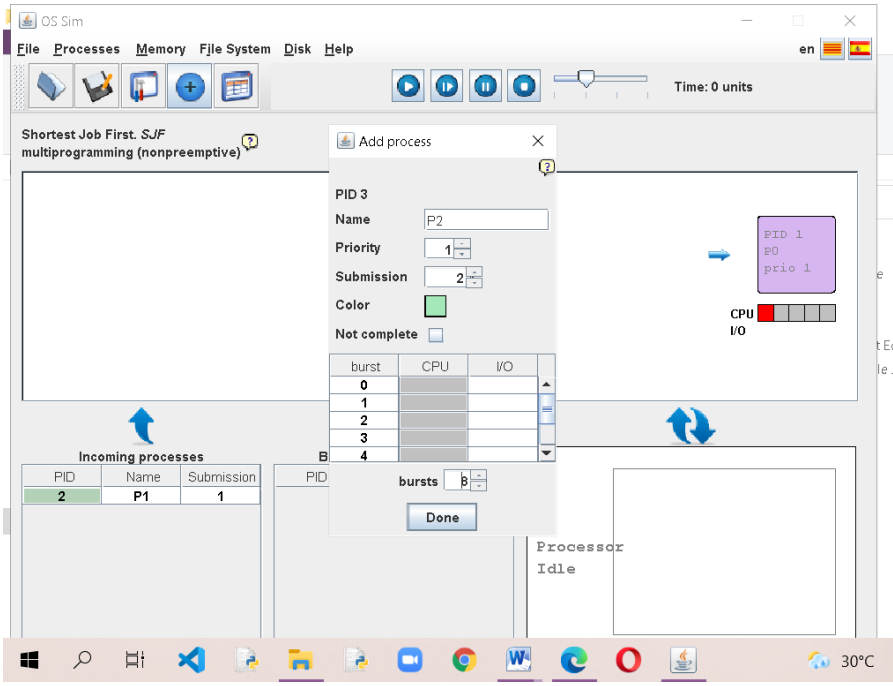
i. Mengubah Algoritma proses menjadi SJF non-premtive.

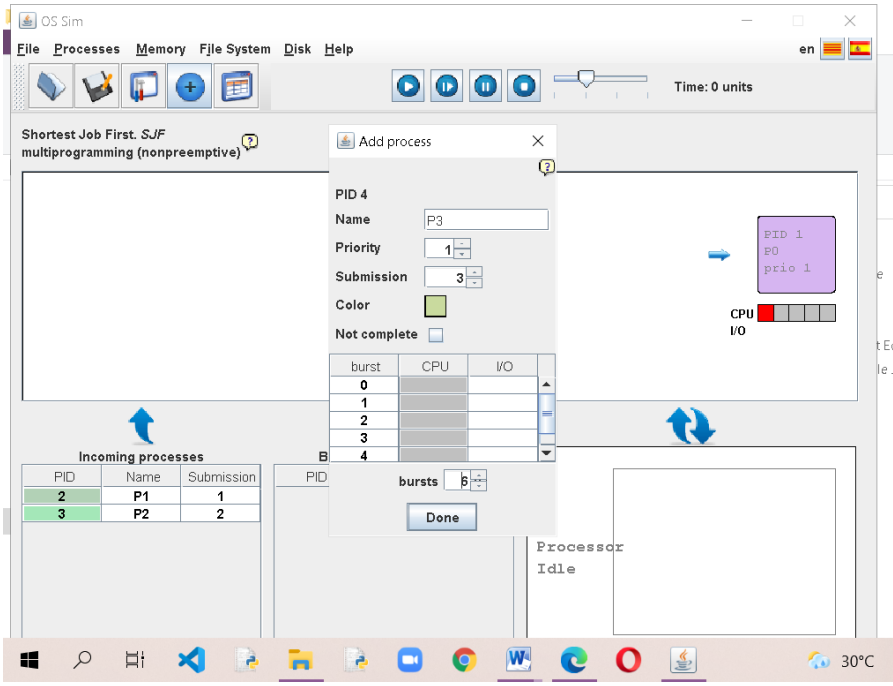


ii. Lakukan input proses sesuai dengan tabel dengan memulai dengan P0 sebagai input proses yang pertama.

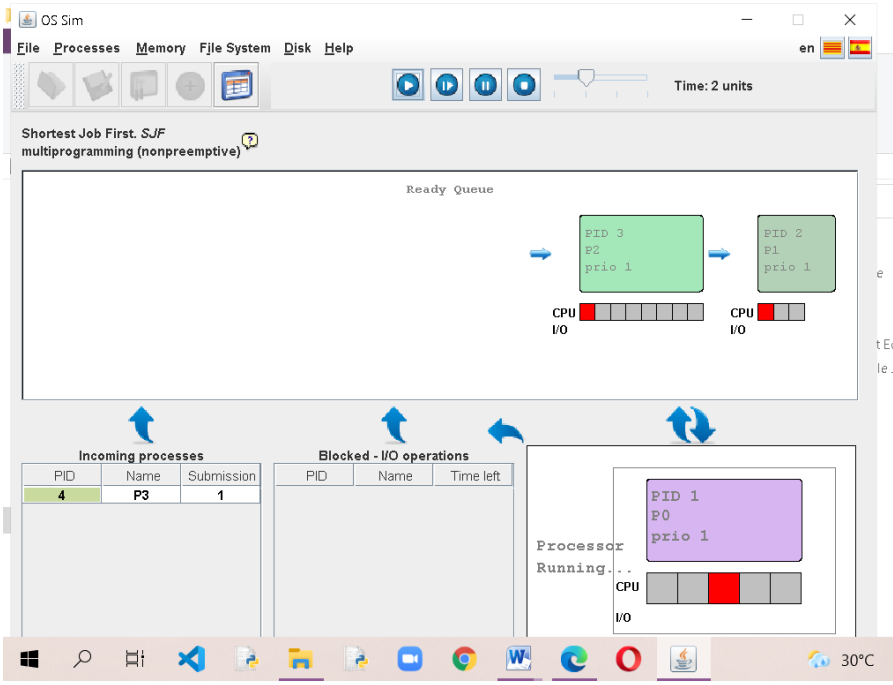


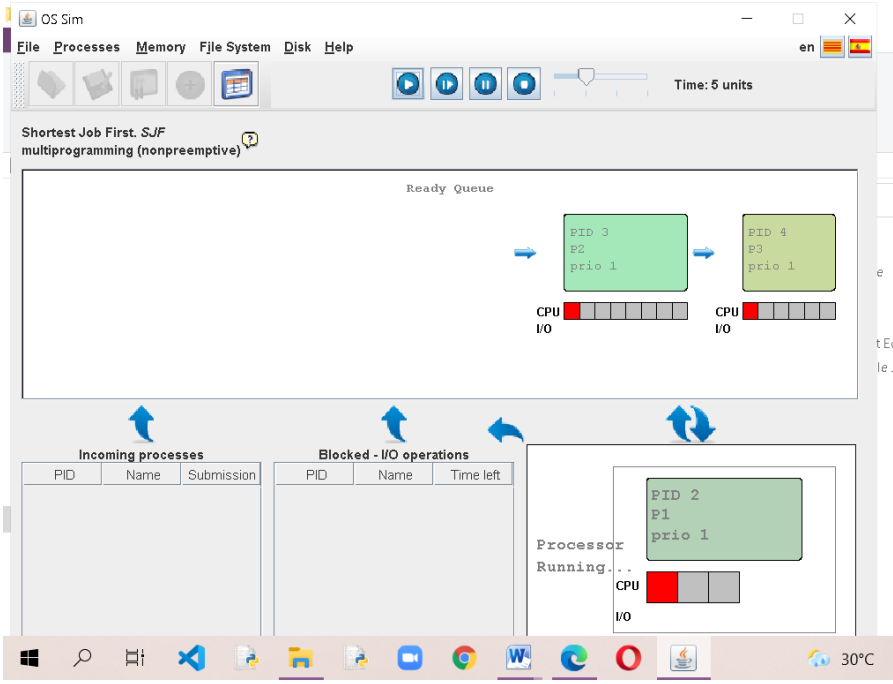

OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
Shortest Job First. SJF multiprogramming (nonpreemptive)
Add process
PID 2
Name P1
Priority 1
Submission 1
Color
Not complete
burst CPU I/O
0
1
2
bursts 3
Done
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Processor
Idle
30°C
IND

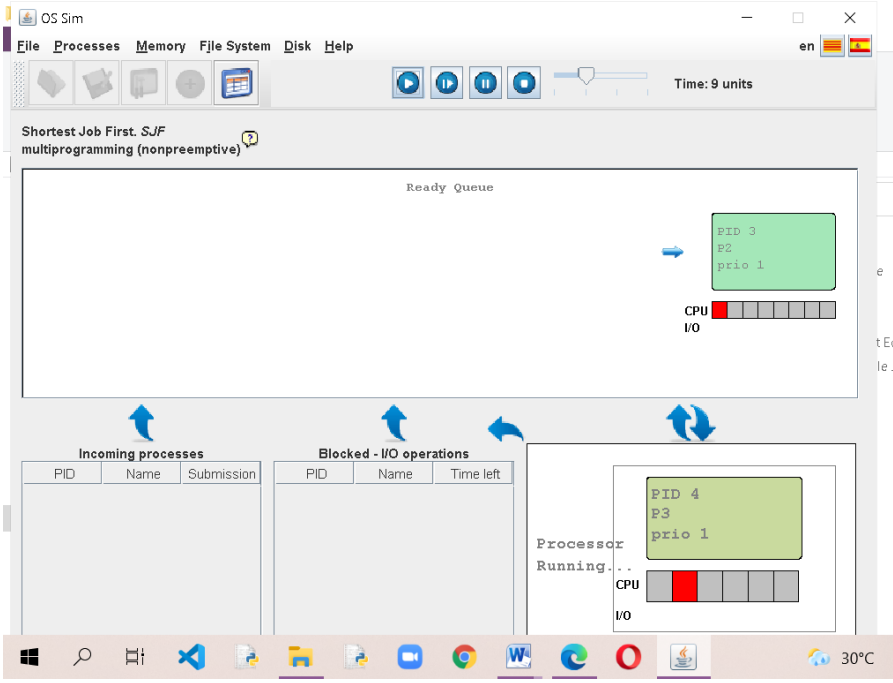

OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
Shortest Job First. SJF multiprogramming (nonpreemptive)
Add process
PID 3
Name P2
Priority 1
Submission 2
Color
Not complete
burst CPU I/O
0
1
2
3
4
bursts 5
Done
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
2 P1 1
Processor
Idle
30°C
IND

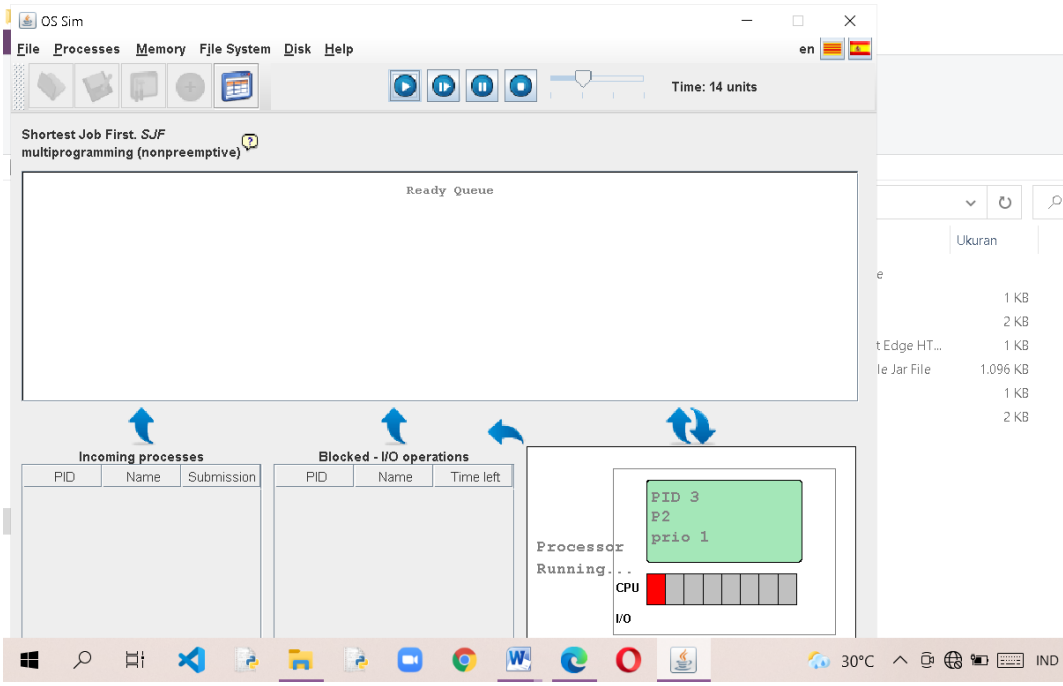


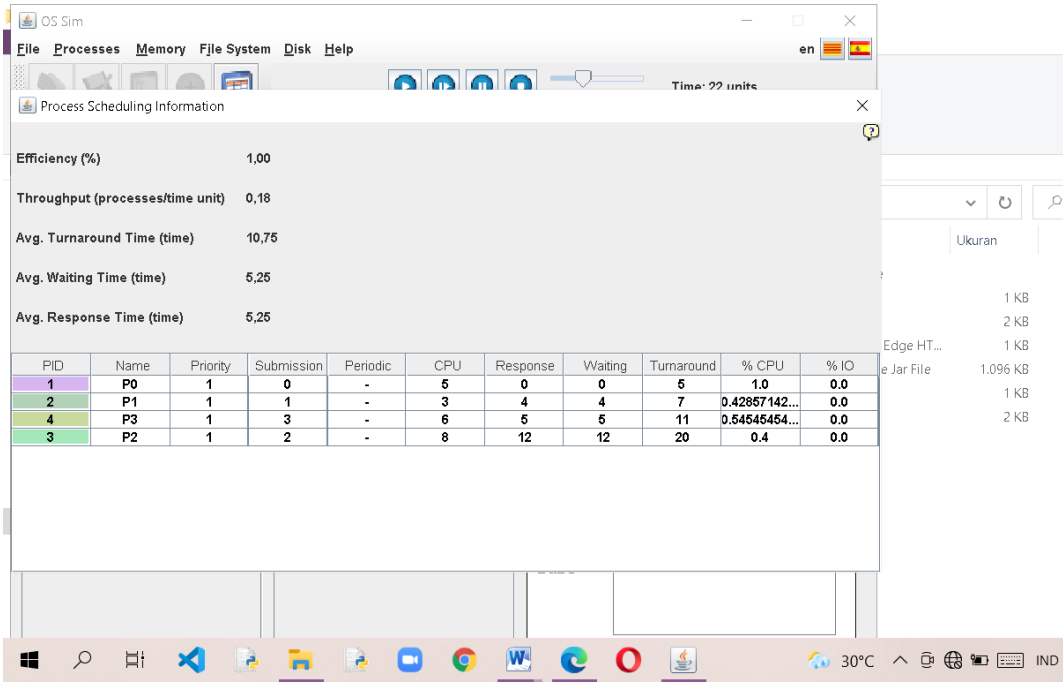
iii. Memulai proses.




OS Sim
File Processes Memory File System Disk Help
en
Time: 5 units
Shortest Job First. SJF
multiprogramming (nonpreemptive)
Ready Queue
PID 3
P2
prio 1
PID 4
P3
prio 1
CPU
I/O
CPU
I/O
Incoming processes
PID
Name
Submission
Blocked - I/O operations
PID
Name
Time left
PID 2
P1
prio 1
Processor
Running...
CPU
I/O
30°C
IND


OS Sim
File Processes Memory File System Disk Help
en
Time: 9 units
Shortest Job First. SJF
multiprogramming (nonpreemptive)
Ready Queue
PID 3
P2
prio 1
CPU
I/O
Incoming processes
PID
Name
Submission
Blocked - I/O operations
PID
Name
Time left
PID 4
P3
prio 1
Processor
Running...
CPU
I/O
30°C
IND





iv. Mengisi table

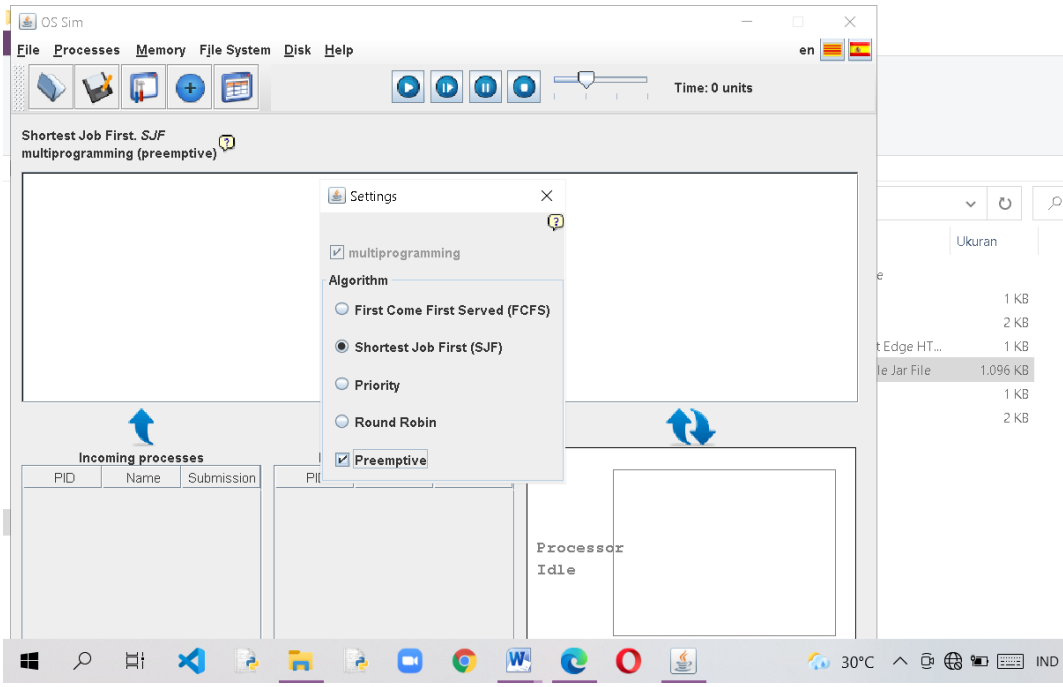
Process	Wait Time : Service Time – Arrival Time
P0	0
P1	4
P2	5
P3	12

Av Wait Time	5.25

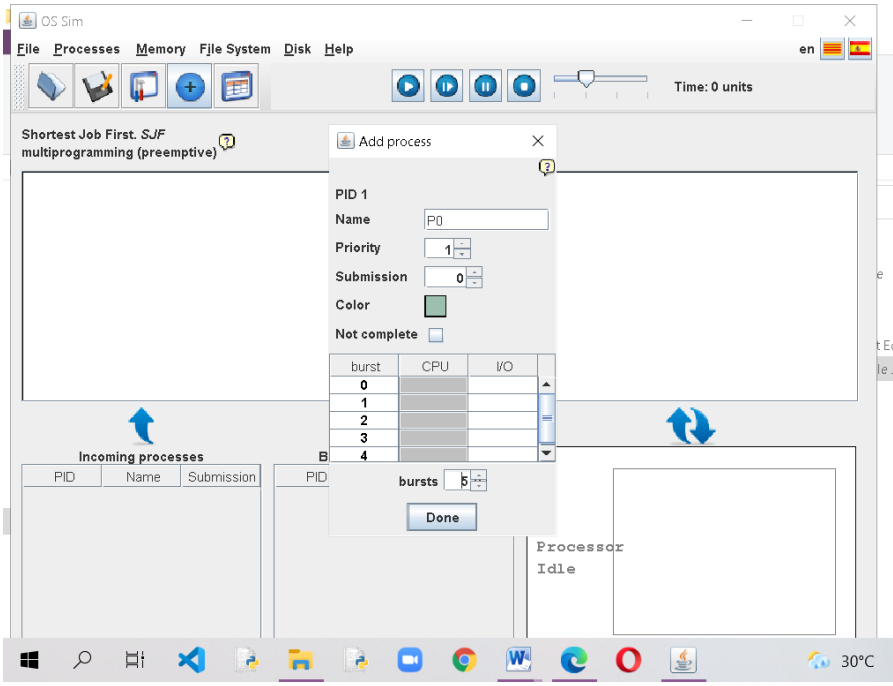
v. Pada algoritma ini, proses dengan burst time terkecil akan dieksekusi terlebih dahulu, sehingga akan dieksekusi berurutan sesuai dengan besar burst time yang dibutuhkan. Akan tetapi hal ini tidak berlaku untuk proses yang datang pertama kali.

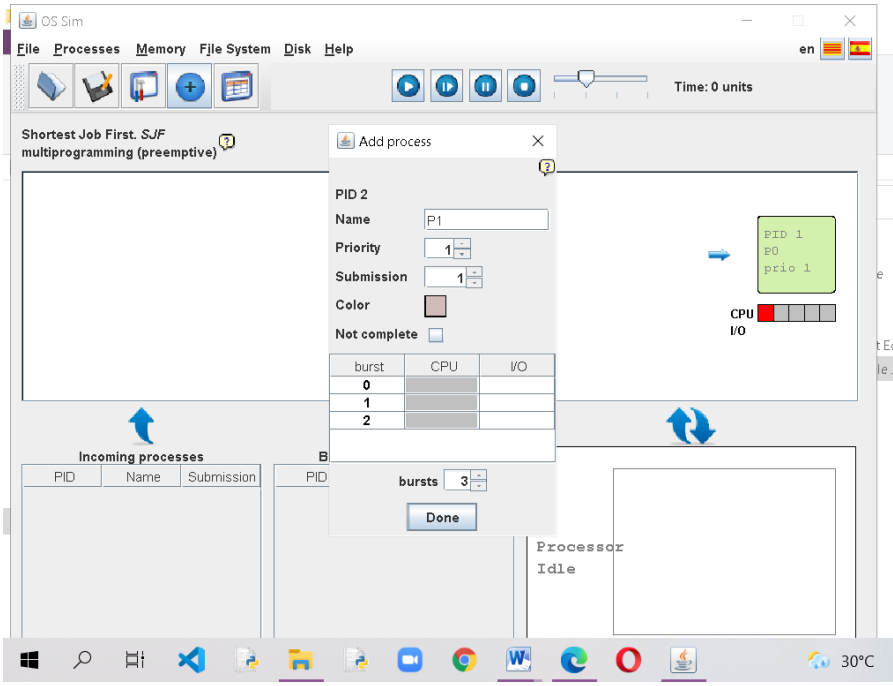
b. Preemptive

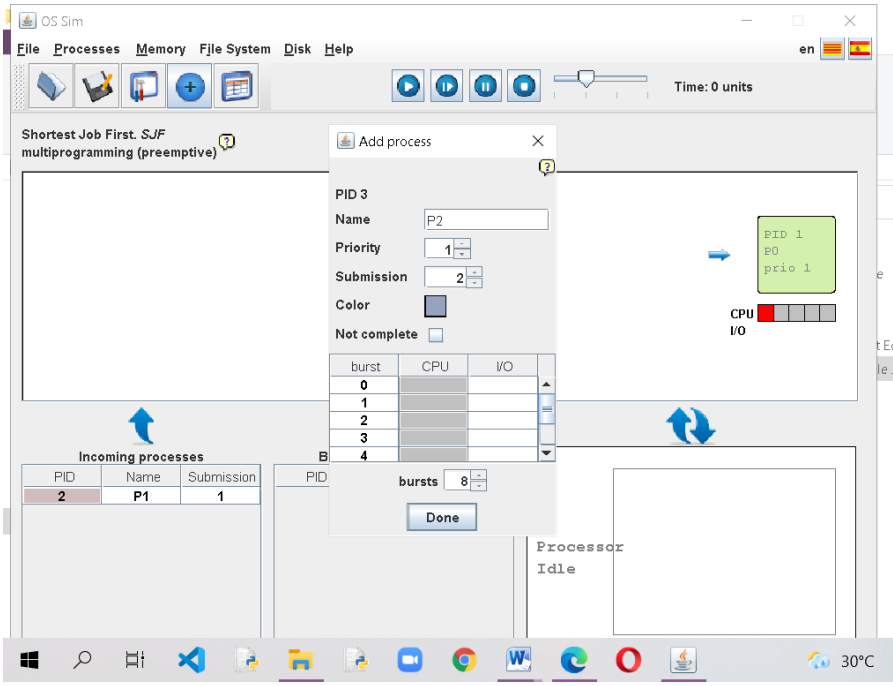
i. Mengubah algoritma proses menjadi SJF preemptive

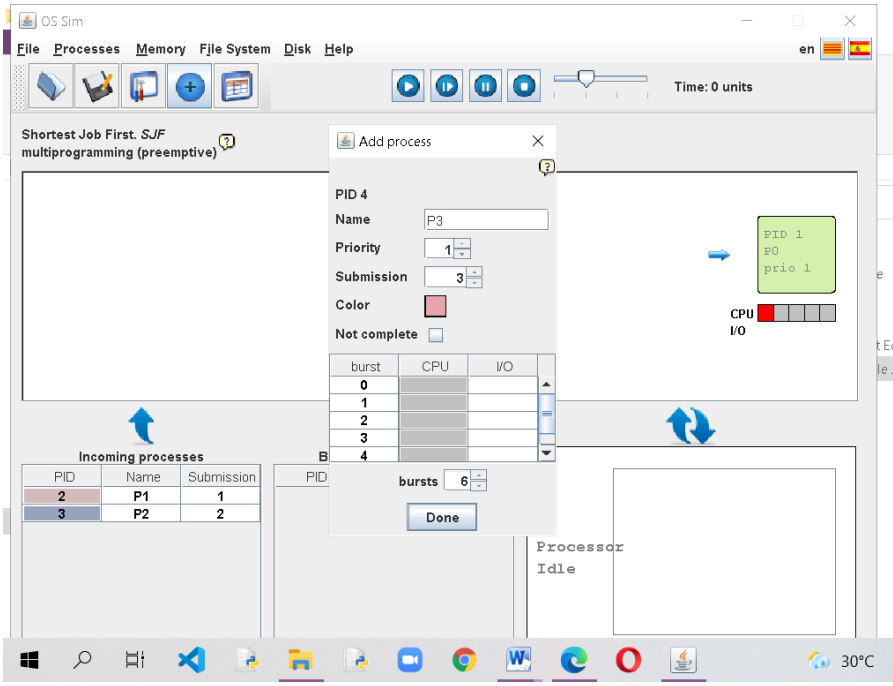


ii. Lakukan input proses sesuai dengan tabel dengan memulai dengan P0 sebagai input proses yang pertama.

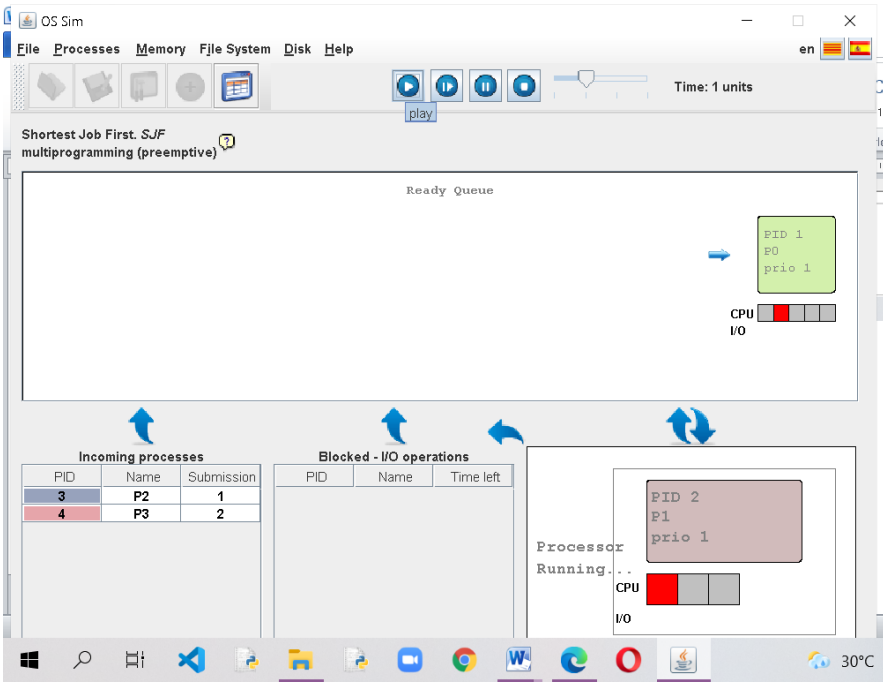

OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
Shortest Job First. SJF multiprogramming (preemptive)
Add process
PID 1
Name P0
Priority 1
Submission 0
Color
Not complete
burst CPU I/O
0
1
2
3
4
bursts 5
Done
Incoming processes
PID Name Submission
Processor
Idle
30°C
IND

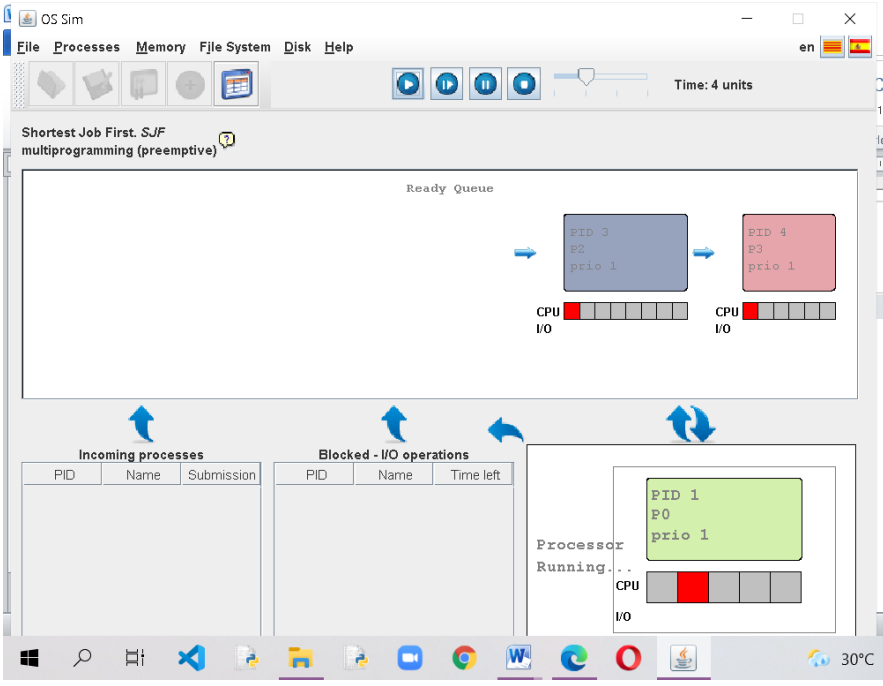

OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
Shortest Job First. SJF multiprogramming (preemptive)
Add process
PID 2
Name P1
Priority 1
Submission 1
Color
Not complete
burst CPU I/O
0
1
2
bursts 3
Done
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Processor
Idle
30°C
IND

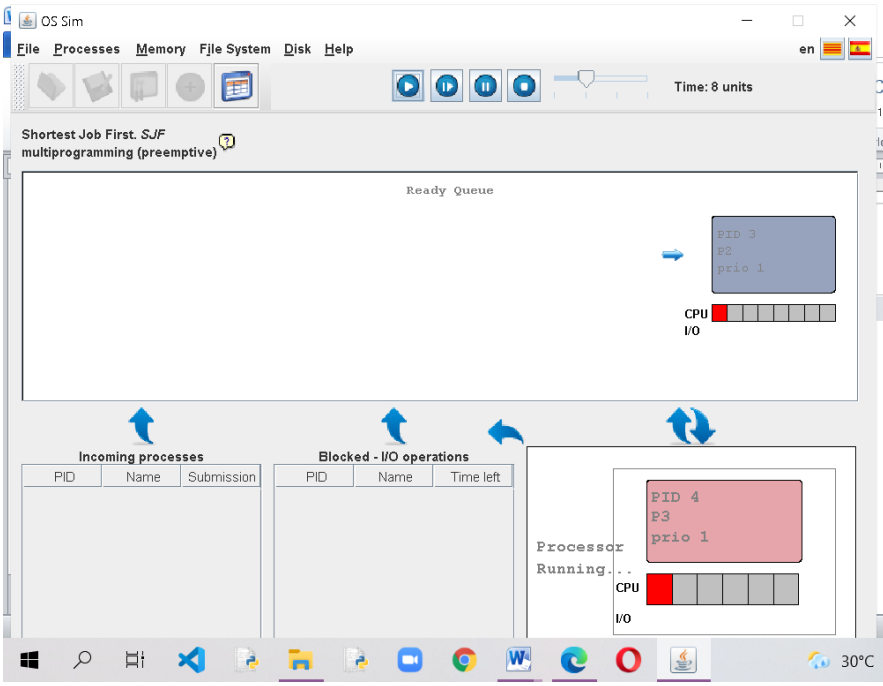


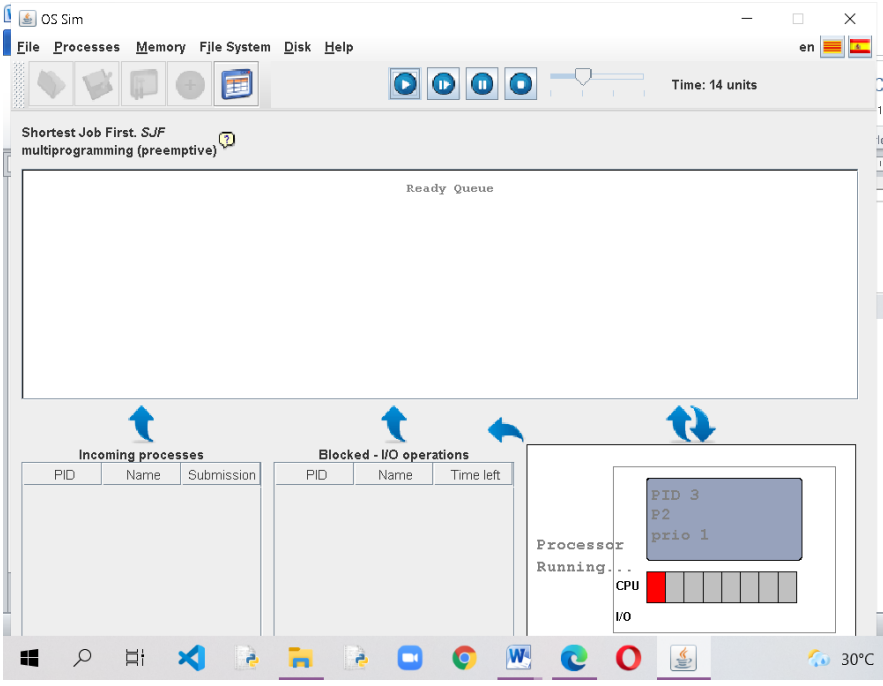


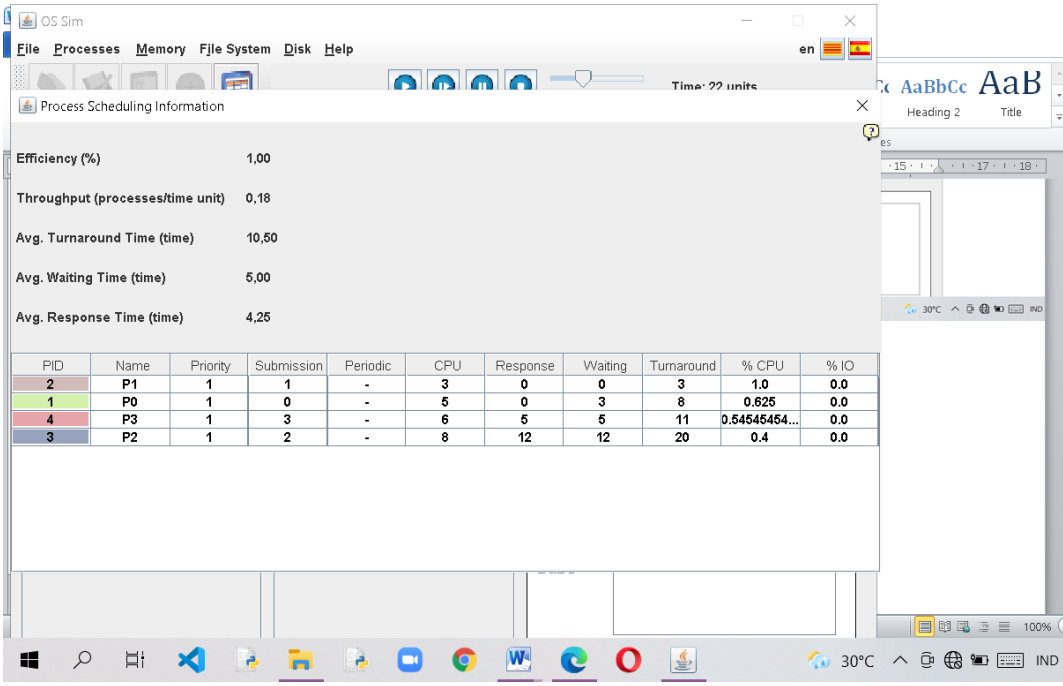
iii. Memulai proses.


OS Sim
File Processes Memory File System Disk Help
en
play
Time: 1 units
Shortest Job First. SJF
multiprogramming (preemptive)
Ready Queue
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
3 P2 1
4 P3 2
Blocked - I/O operations
PID Name Time left
PID 2
P1
prio 1
Processor
Running...
CPU
I/O
30°C
IND


OS Sim
File Processes Memory File System Disk Help
en
Time: 4 units
Shortest Job First. SJF
multiprogramming (preemptive)
Ready Queue
PID 3
P2
prio 1
PID 4
P3
prio 1
CPU
I/O
CPU
I/O
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
PID 1
P0
prio 1
Processor
Running...
CPU
I/O
30°C
IND


OS Sim
File Processes Memory File System Disk Help
en
Time: 8 units
Shortest Job First. SJF
multiprogramming (preemptive)
Ready Queue
PID 3
P2
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
PID 4
P3
prio 1
Processor
Running...
CPU
I/O


OS Sim
File Processes Memory File System Disk Help
en
Time: 14 units
Shortest Job First. SJF
multiprogramming (preemptive)
Ready Queue
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
PID 3
P2
prio 1
Processor
Running...
CPU
I/O



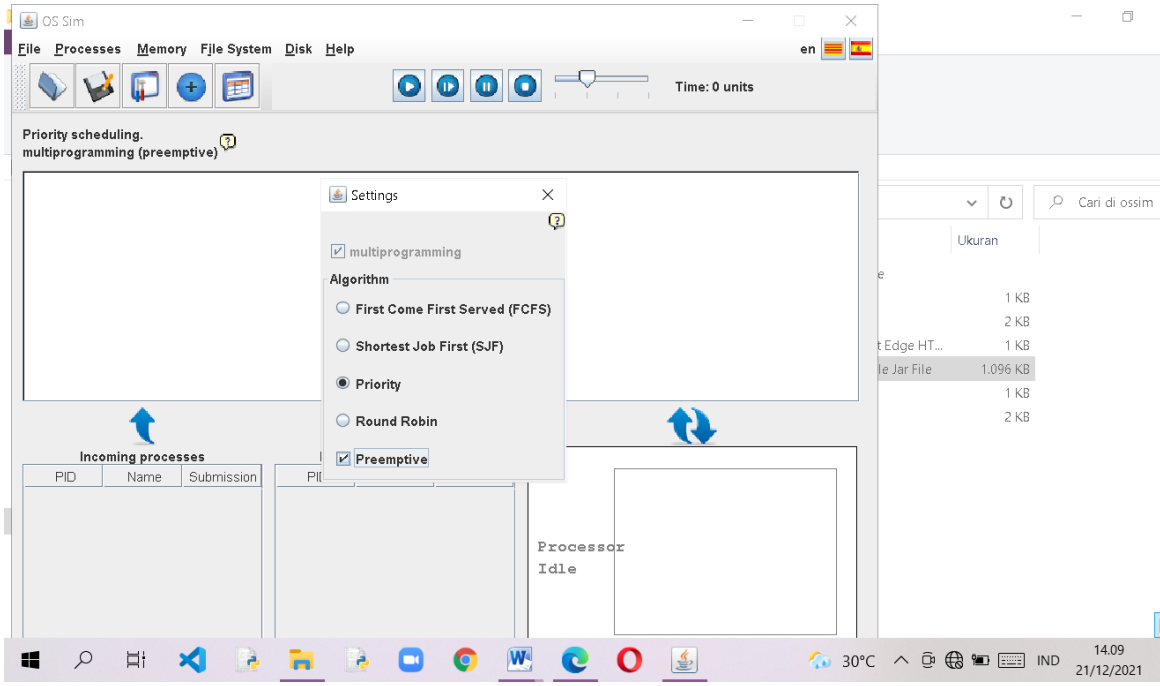
iv. Mengisi table.

Process	Wait Time : Service Time – Arrival Time
P0	0
P1	3
P2	5
P3	12
Av Wait Time	5.00

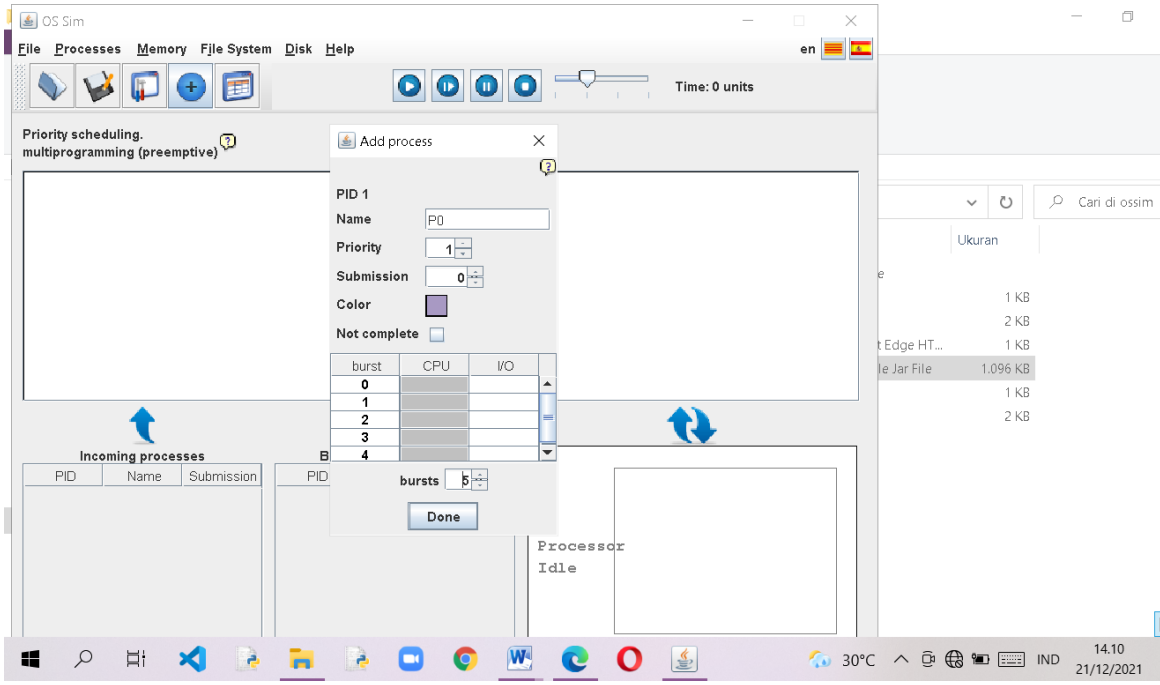
v. Pada algoritma ini, proses yang memiliki burst time paling sedikit akan dieksekusi terlebih dahulu, dan berlaku pada proses yang pertama kali datang (preemtive) sehingga rata-rata waktu tunggu akan kecil.

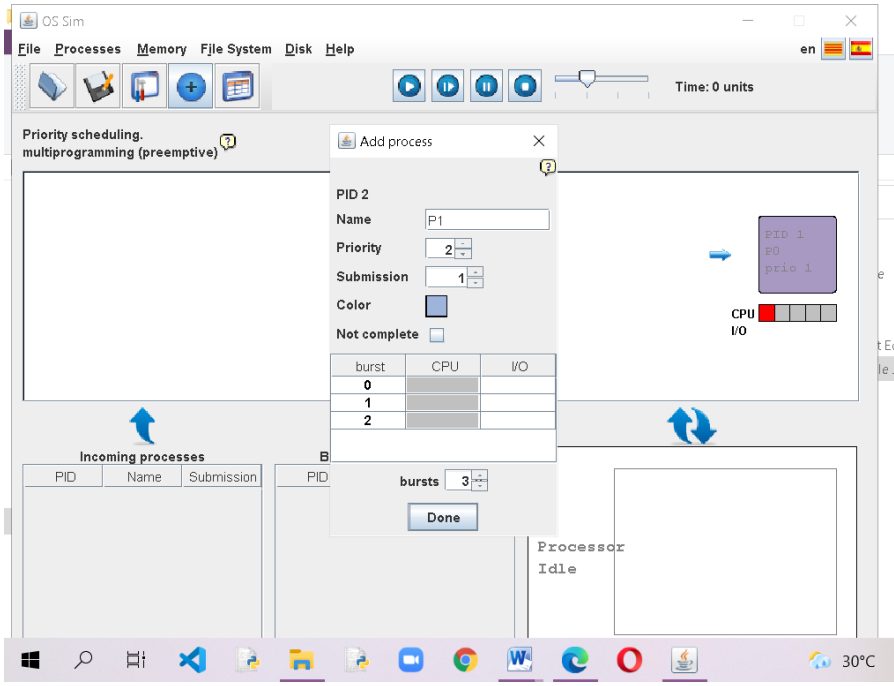
1.3Priority

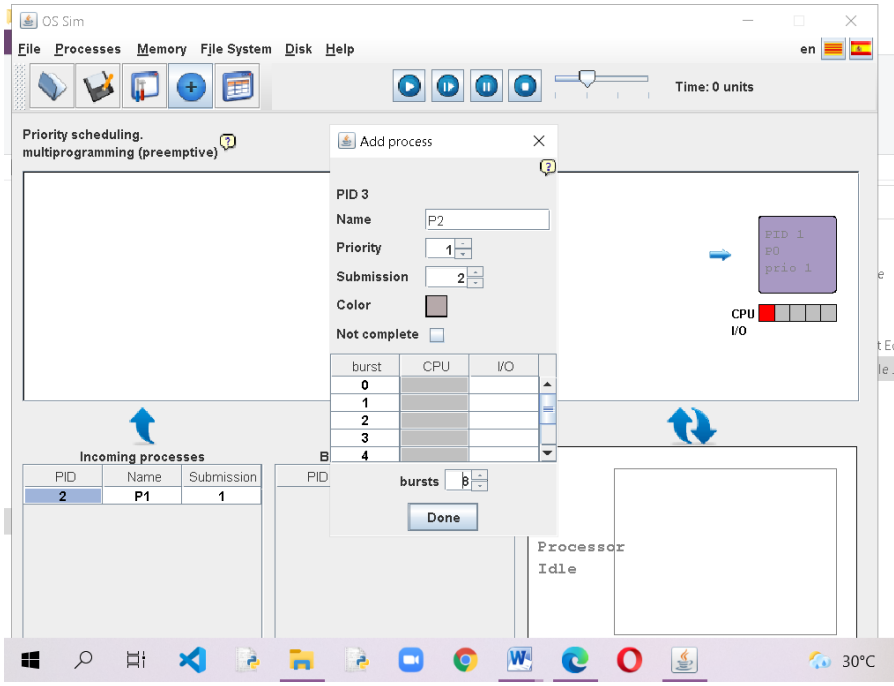
a. Mengubah algoritma proses menjadi Priority.

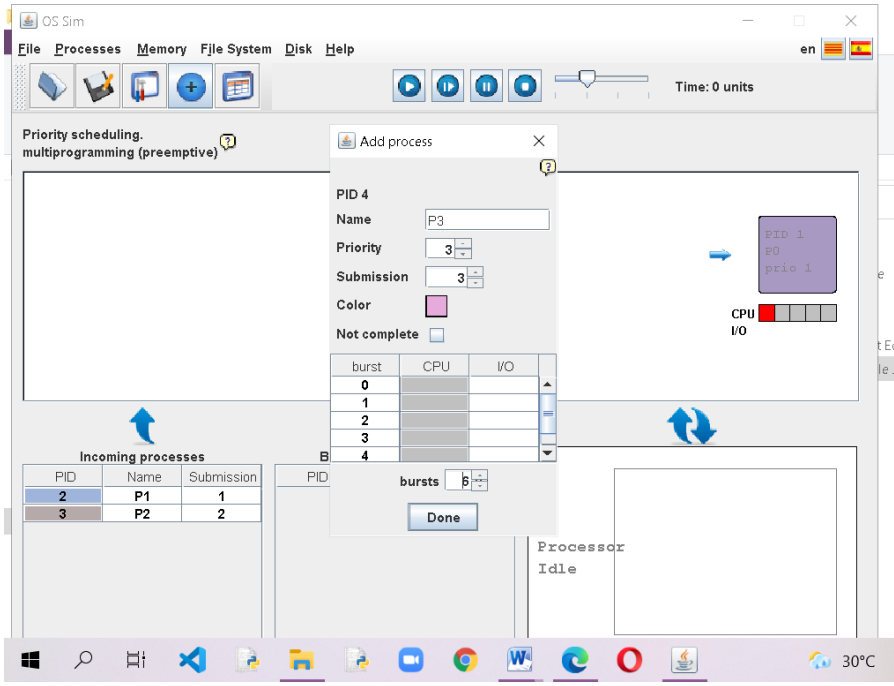


b. Lakukan input proses sesuai dengan tabel dengan memulai dengan P0 sebagai input proses yang pertama.

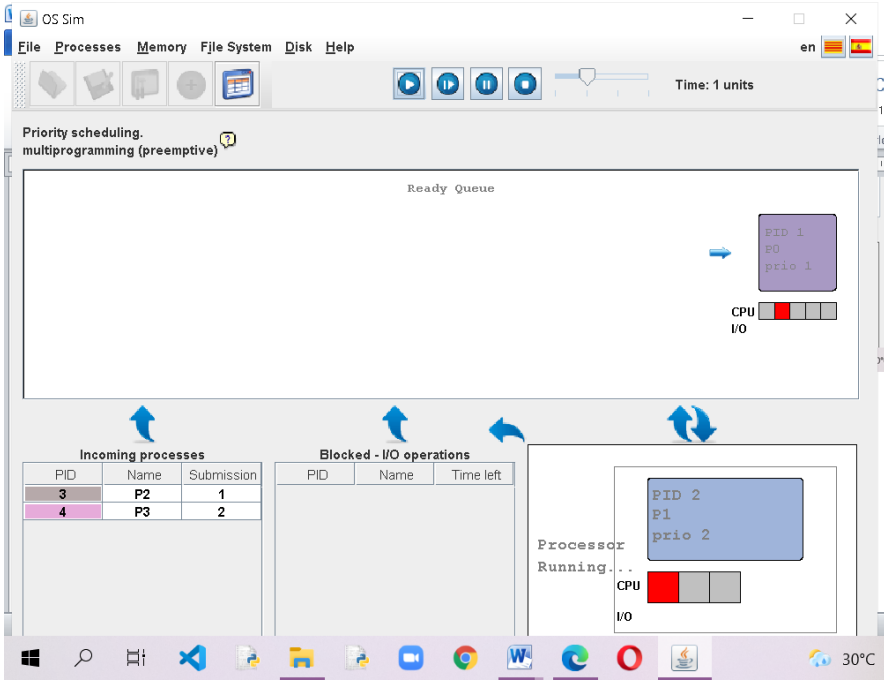


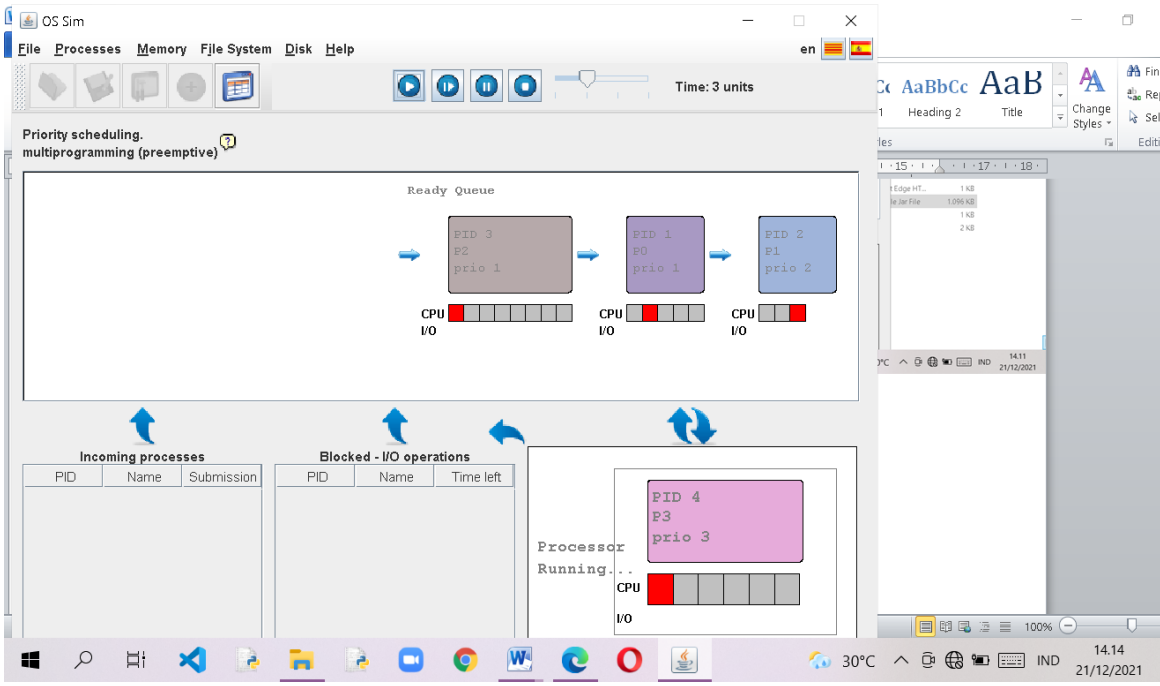

OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
Priority scheduling.
multiprogramming (preemptive)
Add process
PID 2
Name P1
Priority 2
Submission 1
Color
Not complete
burst CPU I/O
0
1
2
bursts 3
Done
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Processor
Idle
30°C
IND
14.11
21/12/2021

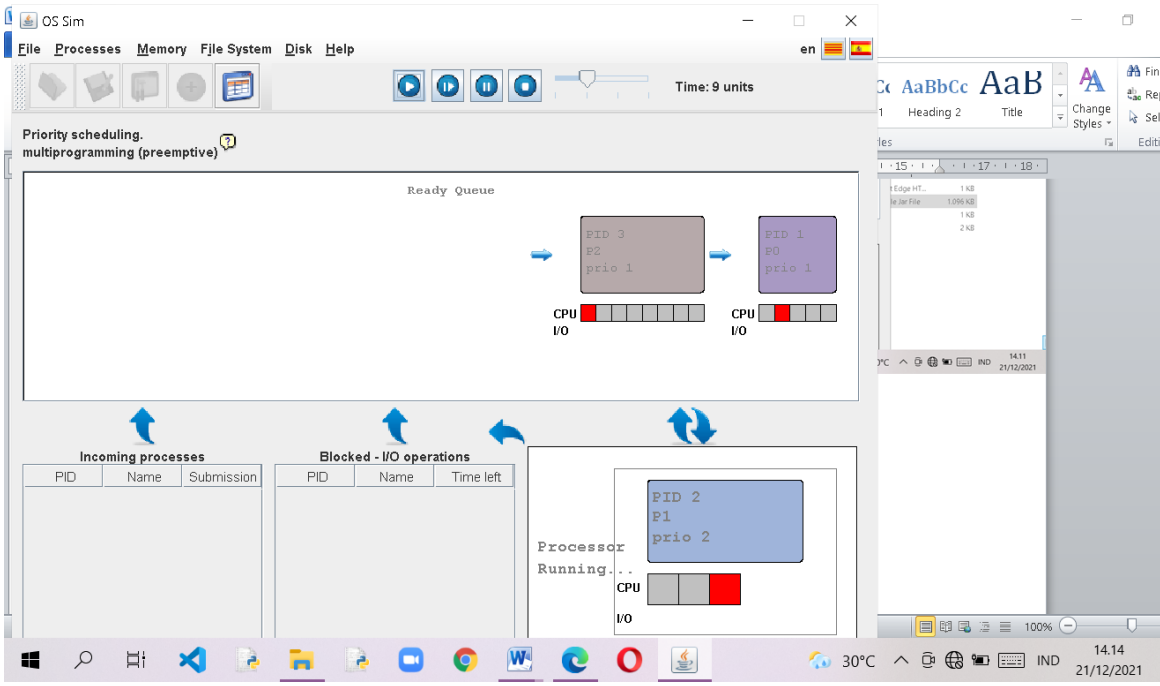

OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
Priority scheduling.
multiprogramming (preemptive)
Add process
PID 3
Name P2
Priority 1
Submission 2
Color
Not complete
burst CPU I/O
0
1
2
3
4
bursts 5
Done
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
2 P1 1
Processor
Idle
30°C
IND
14.11
21/12/2021

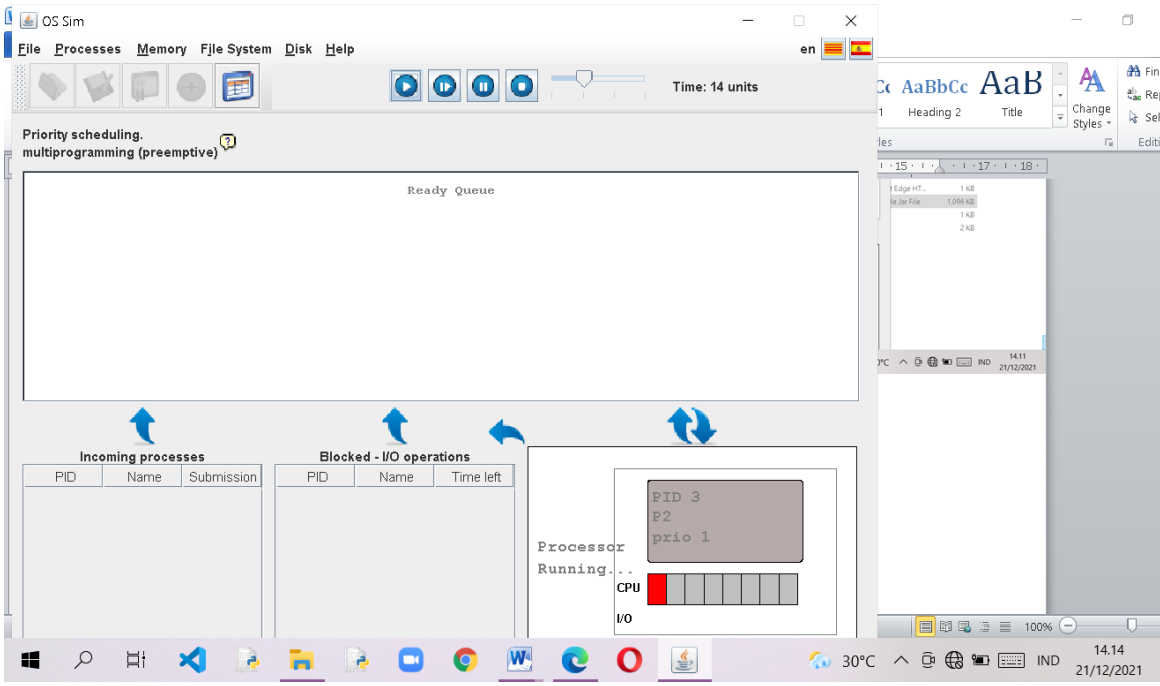


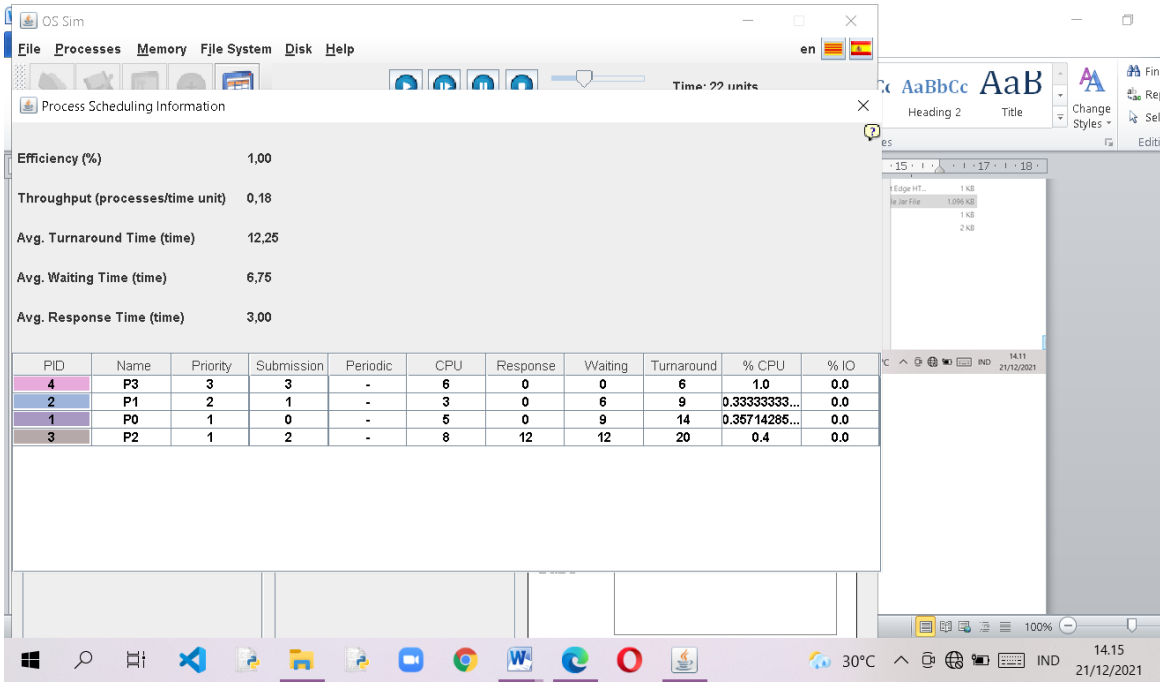
c. Memulai proses.




OS Sim
File Processes Memory File System Disk Help
en
Time: 3 units
Priority scheduling.
multiprogramming (preemptive)
Ready Queue
PID 3
P2
prio 1
PID 1
P0
prio 1
PID 2
P1
prio 2
CPU
I/O
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
Processor
Running...
PID 4
P3
prio 3
30°C
IND
14.14
21/12/2021


OS Sim
File Processes Memory File System Disk Help
en
Time: 9 units
Priority scheduling.
multiprogramming (preemptive)
Ready Queue
PID 3
P2
prio 1
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
Processor
Running...
PID 2
P1
prio 2
30°C
IND
14.14
21/12/2021





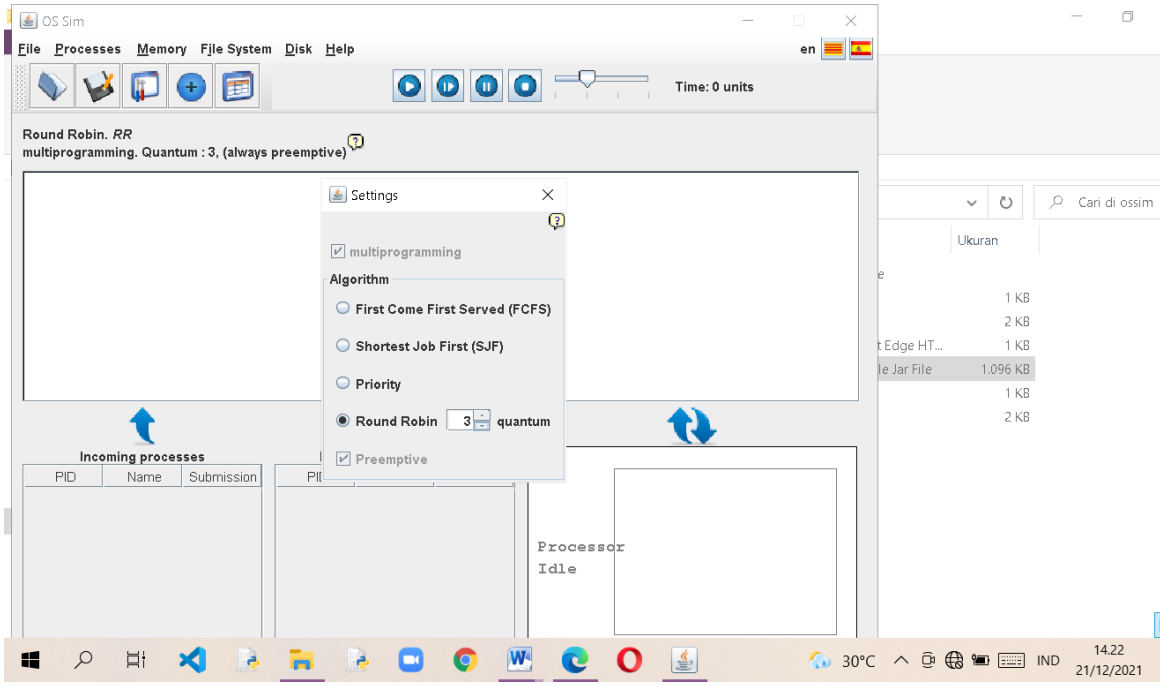
d. Mengisi Table.

Process	Wait Time : Service Time – Arrival Time
P0	0
P1	6
P2	9
P3	12
Av Wait Time	6.75

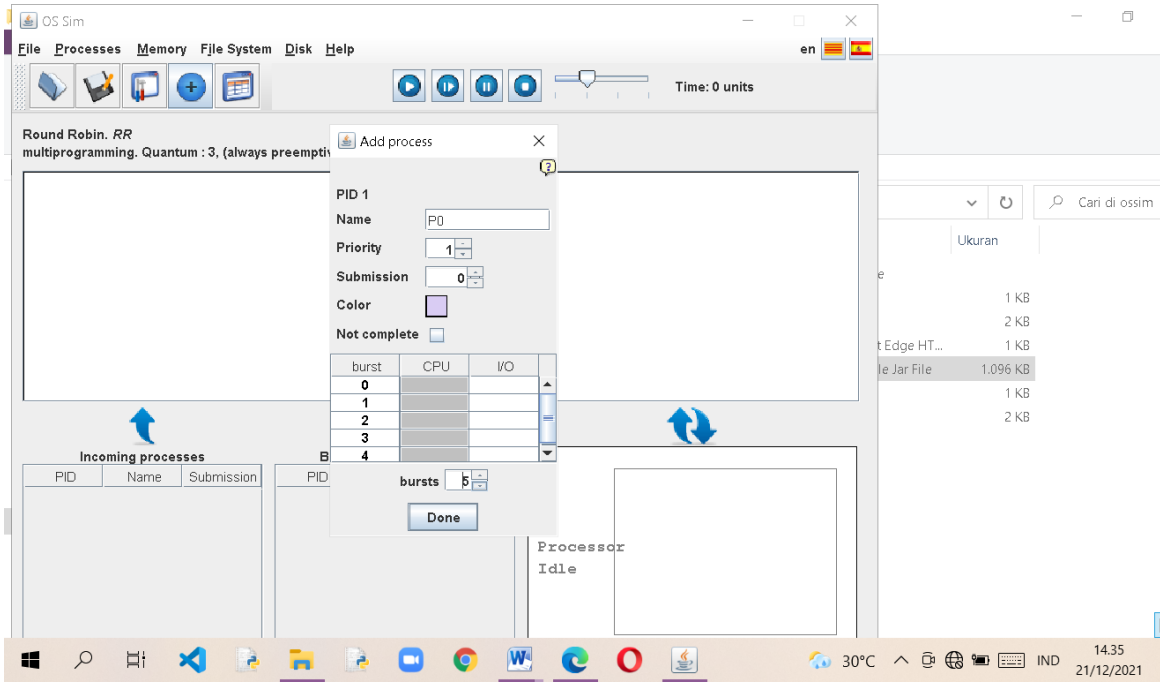
e. Pada algoritma ini, proses akan dieksekusi sesuai dengan prioritasnya tanpa memandang burst time yang dibutuhkan.

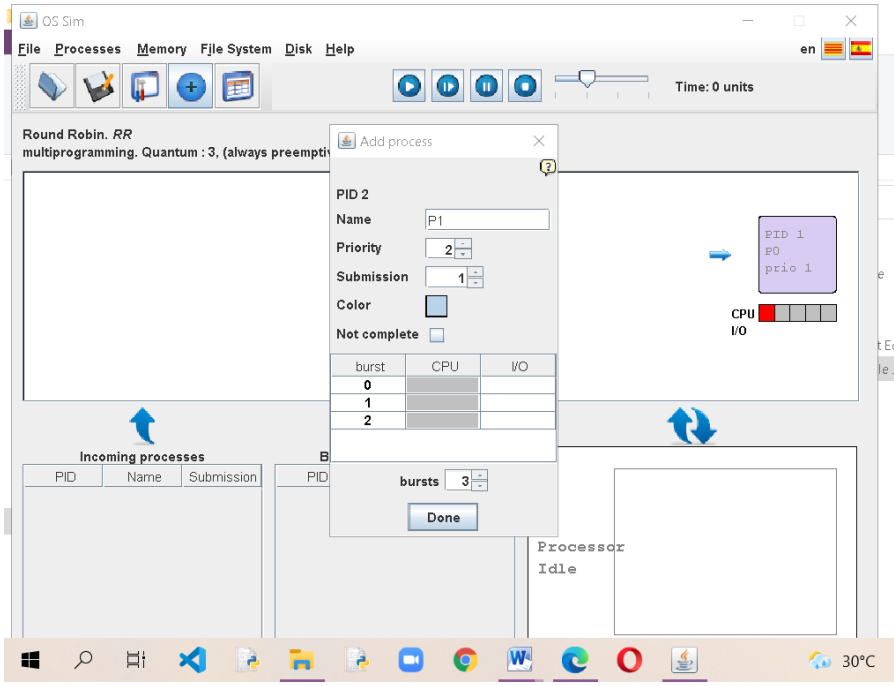
1.4Round Robin

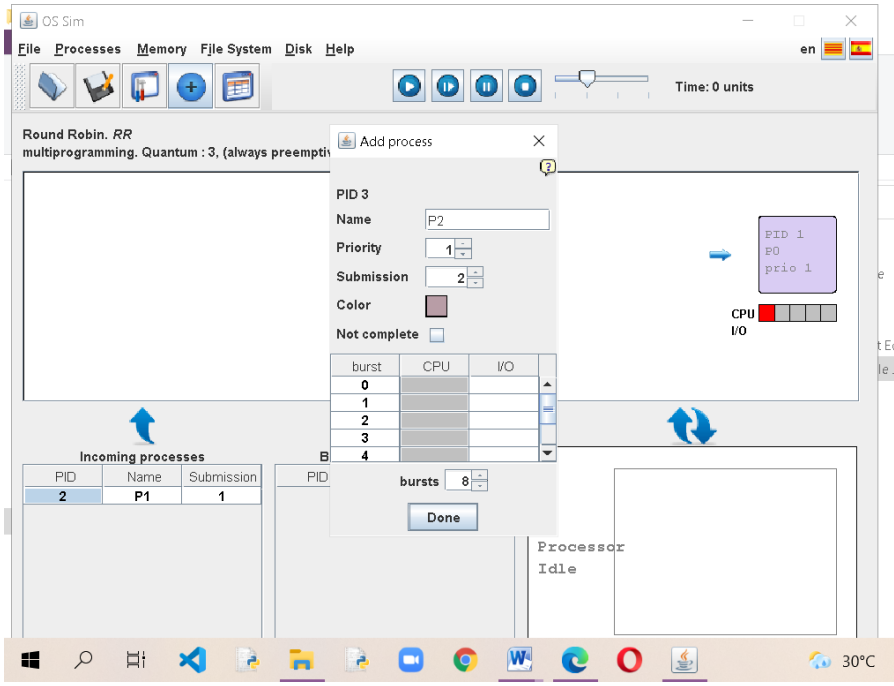
a. Mengubah algoritma proses menjadi Round Robin dengan quantum time-nya adalah 3.

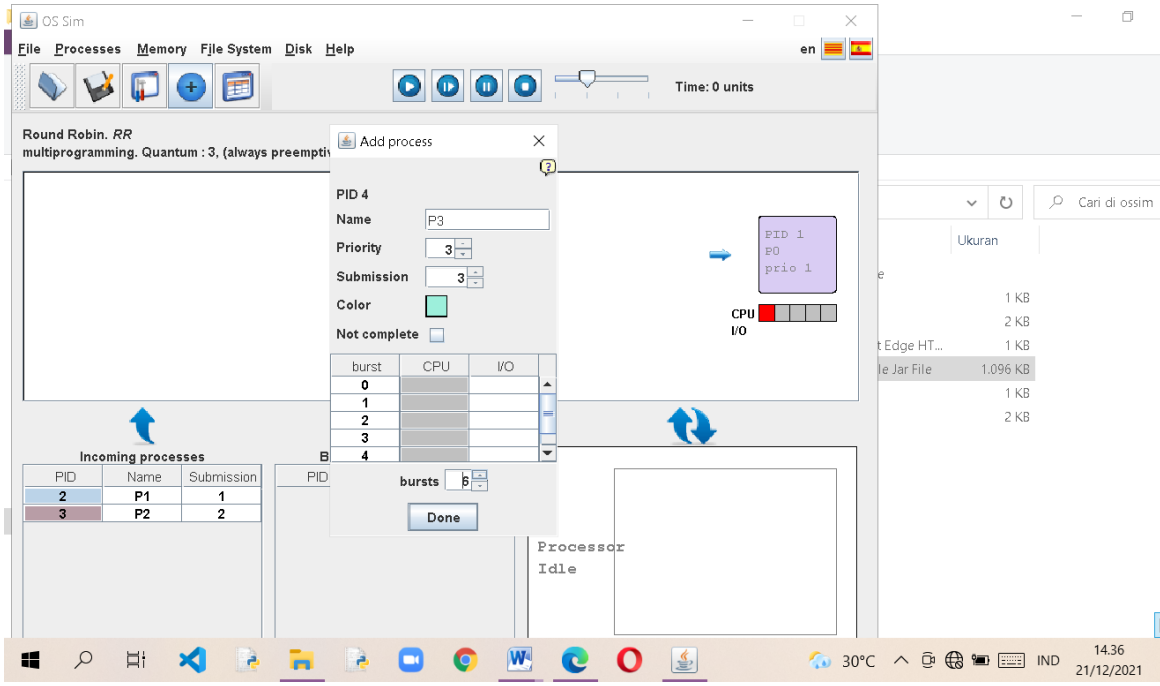


b. Lakukan input proses sesuai dengan tabel dengan memulai dengan P0 sebagai input proses yang pertama.

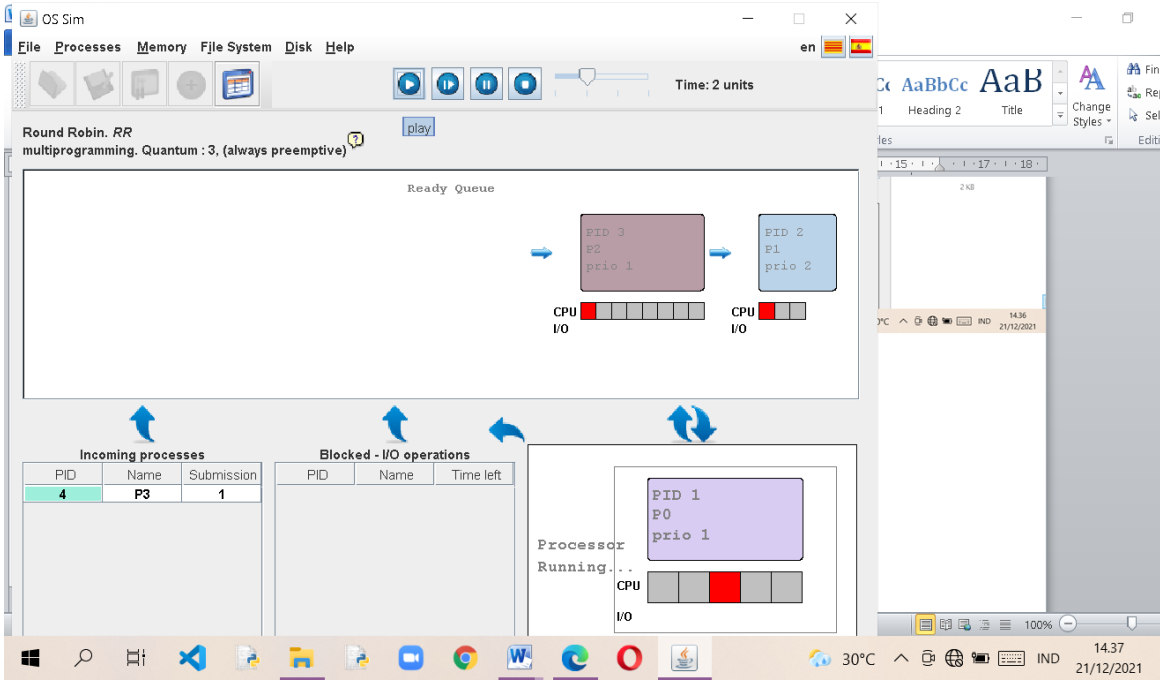


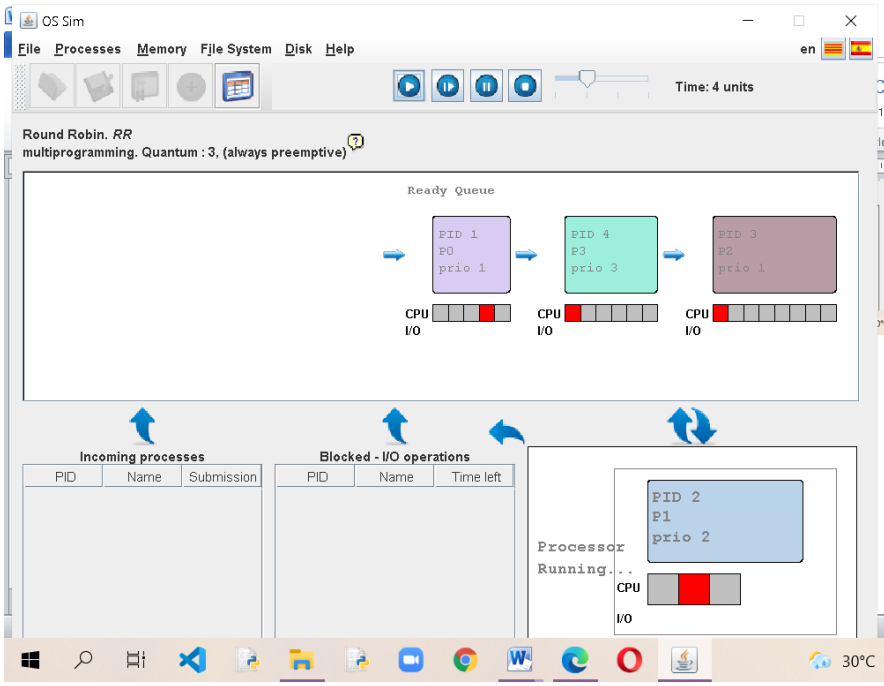

OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
Round Robin. RR
multiprogramming. Quantum : 3, (always preempti
Add process
PID 2
Name P1
Priority 2
Submission 1
Color
Not complete
burst CPU I/O
0
1
2
bursts 3
Done
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Processor
Idle
30°C
IND
14.35
21/12/2021

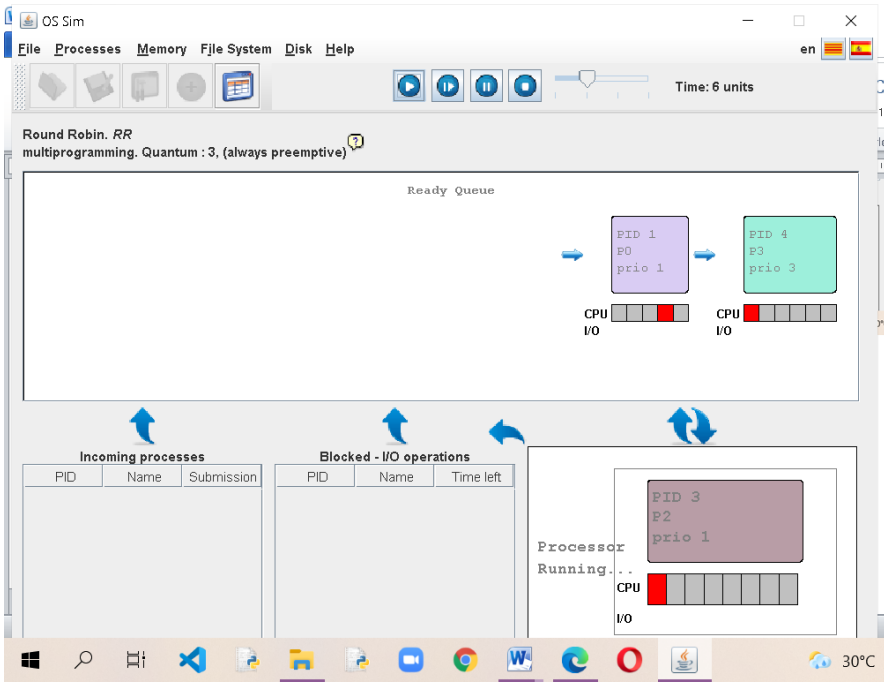

OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
Round Robin. RR
multiprogramming. Quantum : 3, (always preempti
Add process
PID 3
Name P2
Priority 1
Submission 2
Color
Not complete
burst CPU I/O
0
1
2
3
4
bursts 8
Done
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
2 P1 1
Processor
Idle
30°C
IND
14.36
21/12/2021

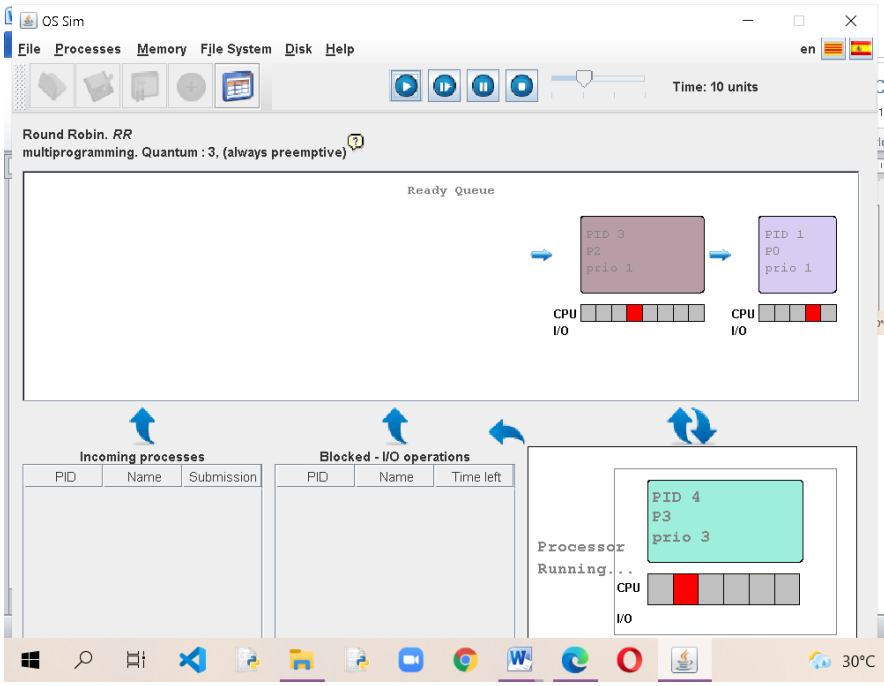


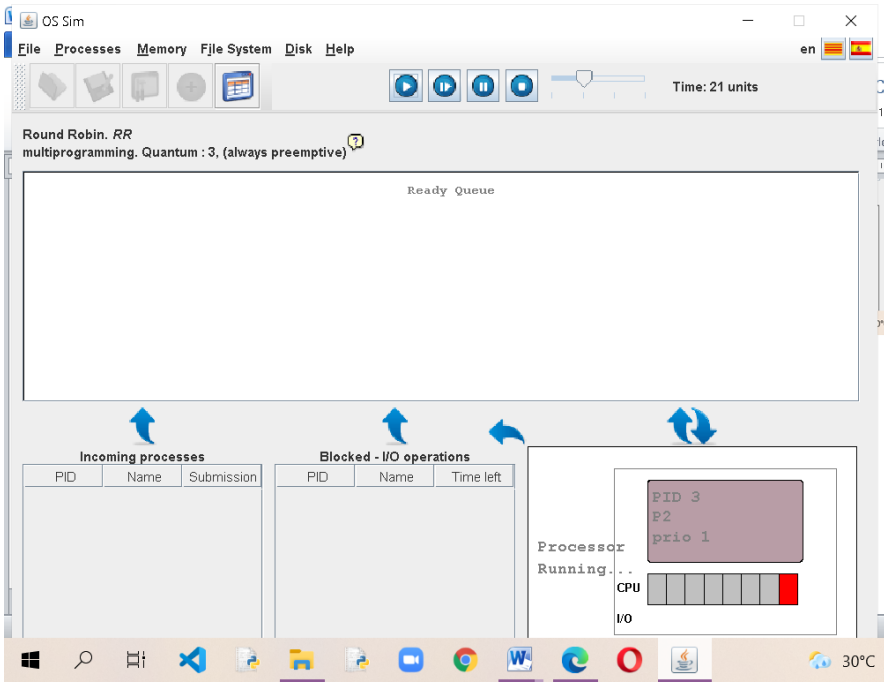
c. Memulai proses.

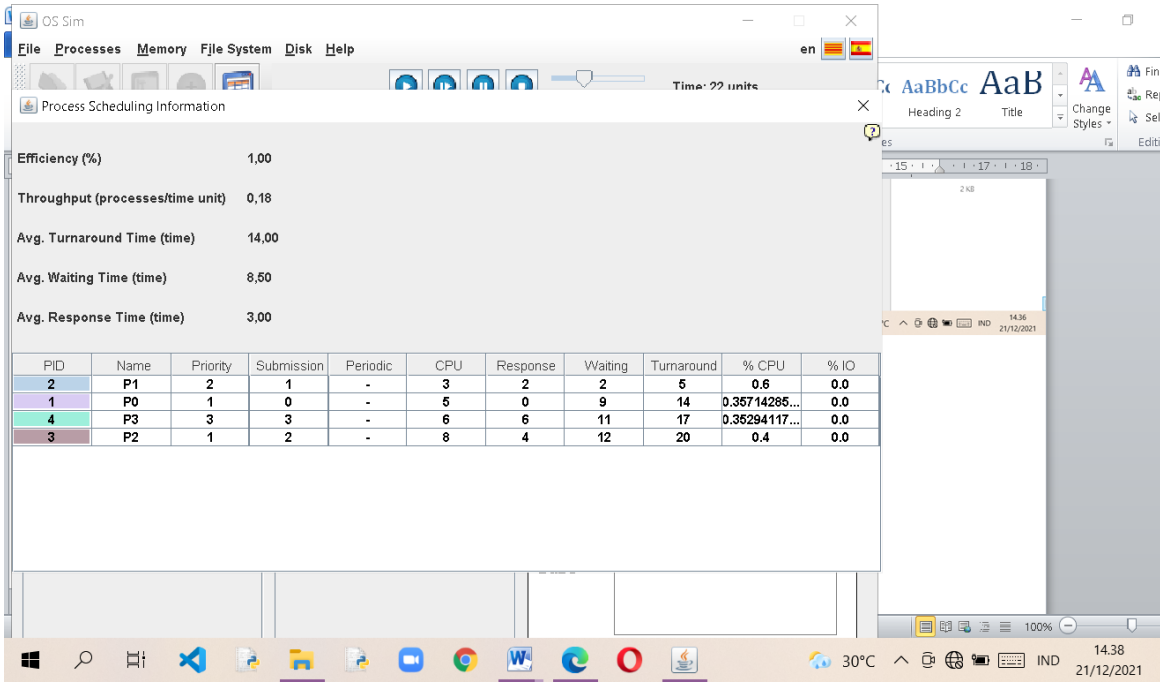



OS Sim
File Processes Memory File System Disk Help
en
Time: 4 units
Round Robin. RR
multiprogramming. Quantum : 3, (always preemptive)
Ready Queue
PID 1
P0
prio 1
PID 4
P3
prio 3
PID 3
P2
prio 1
CPU
I/O
Incoming processes
PID
Name
Submission
Blocked - I/O operations
PID
Name
Time left
Processor
Running...
PID 2
P1
prio 2
CPU
I/O
30°C
IND
14.37
21/12/2021


OS Sim
File Processes Memory File System Disk Help
en
Time: 6 units
Round Robin. RR
multiprogramming. Quantum : 3, (always preemptive)
Ready Queue
PID 1
P0
prio 1
PID 4
P3
prio 3
CPU
I/O
Incoming processes
PID
Name
Submission
Blocked - I/O operations
PID
Name
Time left
Processor
Running...
PID 3
P2
prio 1
CPU
I/O
30°C
IND
14.37
21/12/2021


OS Sim
File Processes Memory File System Disk Help
en
Time: 10 units
Round Robin. RR
multiprogramming. Quantum : 3, (always preemptive)
Ready Queue
PID 3
P2
prio 1
PID 1
P0
prio 1
CPU
I/O
CPU
I/O
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
Processor
Running...
PID 4
P3
prio 3
CPU
I/O
30°C
IND
14.37
21/12/2021


OS Sim
File Processes Memory File System Disk Help
en
Time: 21 units
Round Robin. RR
multiprogramming. Quantum : 3, (always preemptive)
Ready Queue
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
Processor
Running...
PID 3
P2
prio 1
CPU
I/O
30°C
IND
14.38
21/12/2021



d. Mengisi Table.

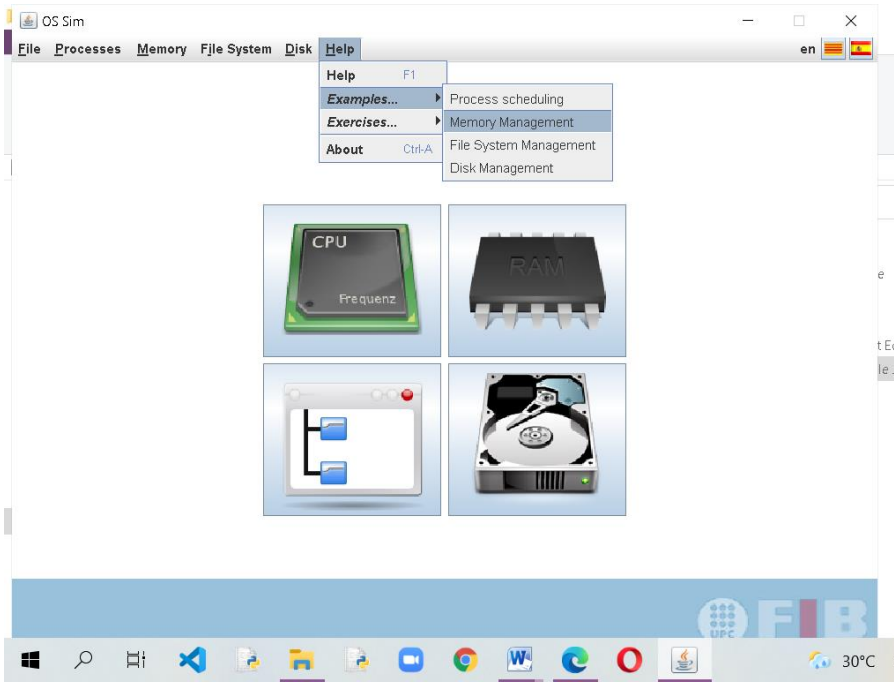
Process	Wait Time : Service Time – Arrival Time
P0	2
P1	9
P2	11
P3	12
Av Wait Time	6.50

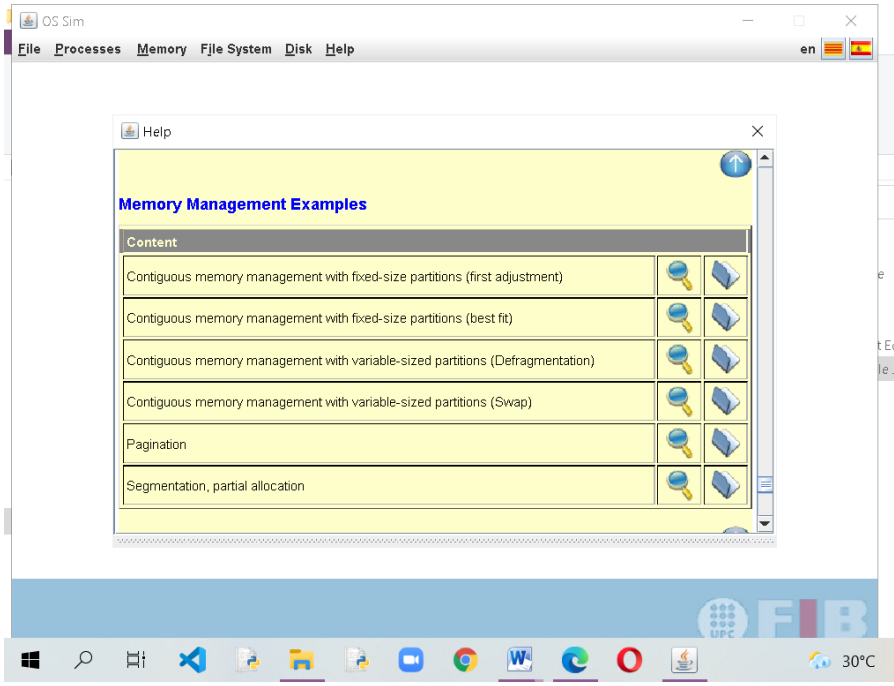
e. Pada algoritma ini, semua proses memiliki jatah waktu eksekusinya sehingga akan adil. Akan tetapi dikarenakan memiliki jatah eksekusi dan terjadi banyak switch saat waktu jatahnya habis, rata-rata waktu tunggunya menjadi semakin besar

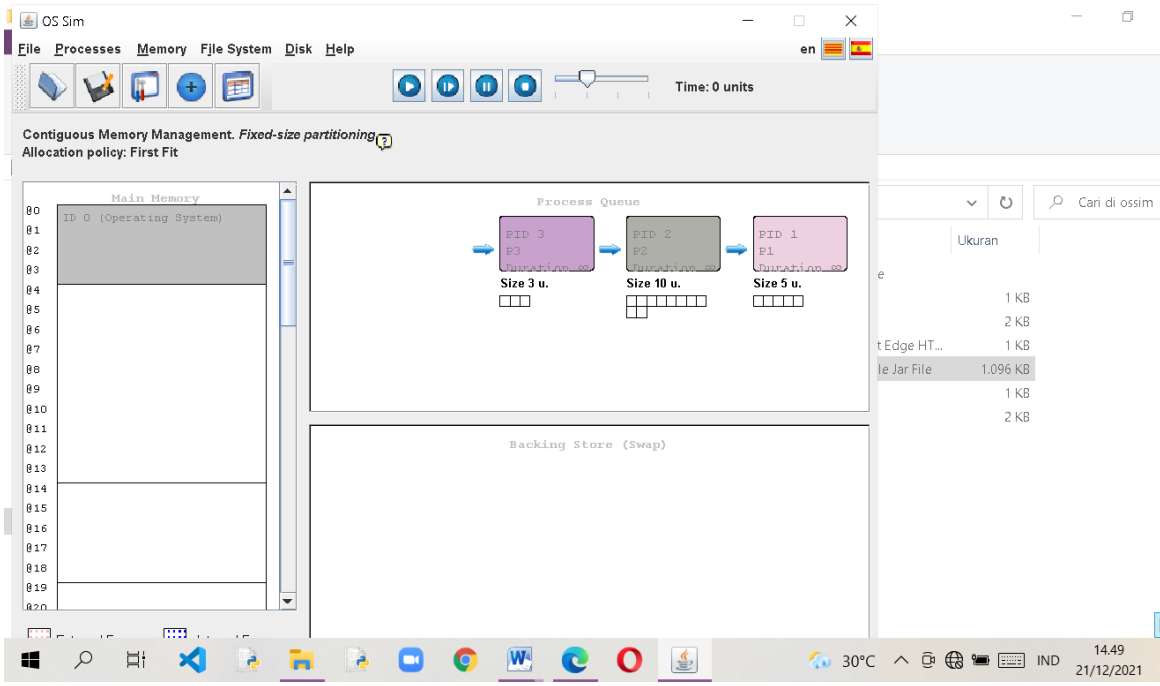
2. Kegiatan 2: Manajemen Memori

2.1 Contiguous Memory Management dengan menggunakan fixed-size partition dan aturan first fit.

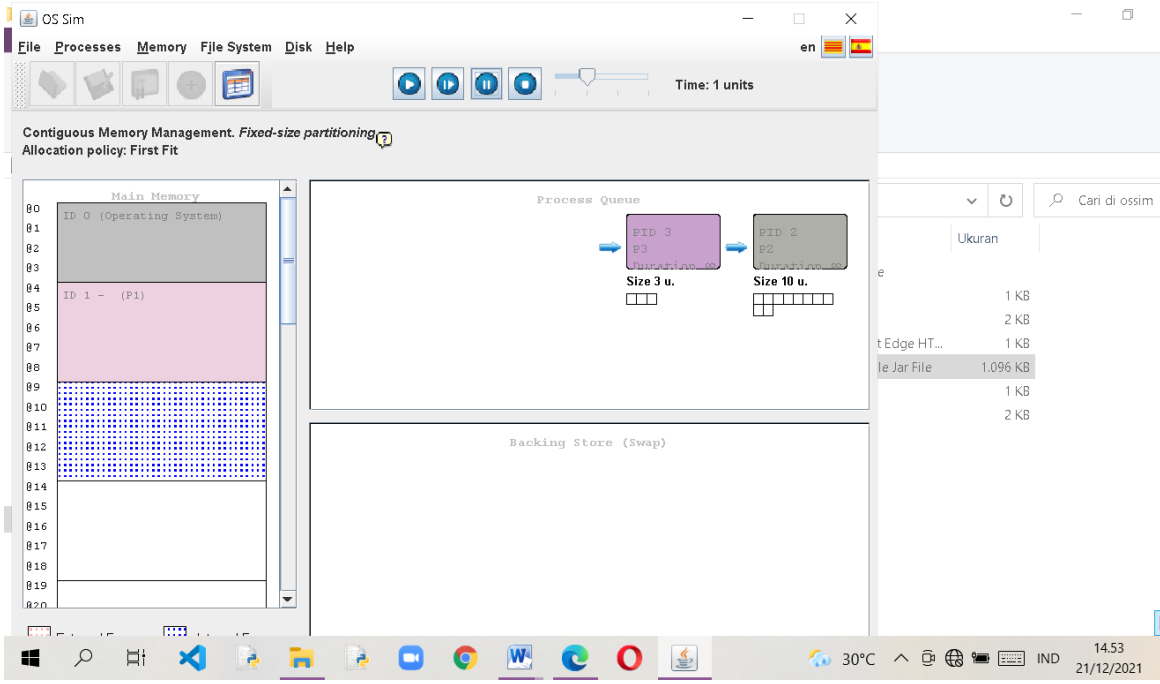
a. Memasuki folder dengan langkah: help>examples>memory management>contiguous memory management with fixed-size partition (first adjustment).


OS Sim
File Processes Memory File System Disk Help en
Help F1
Examples...
Exercises...
About Ctrl-A
Process scheduling
Memory Management
File System Management
Disk Management
CPU
Frequenz
RAM

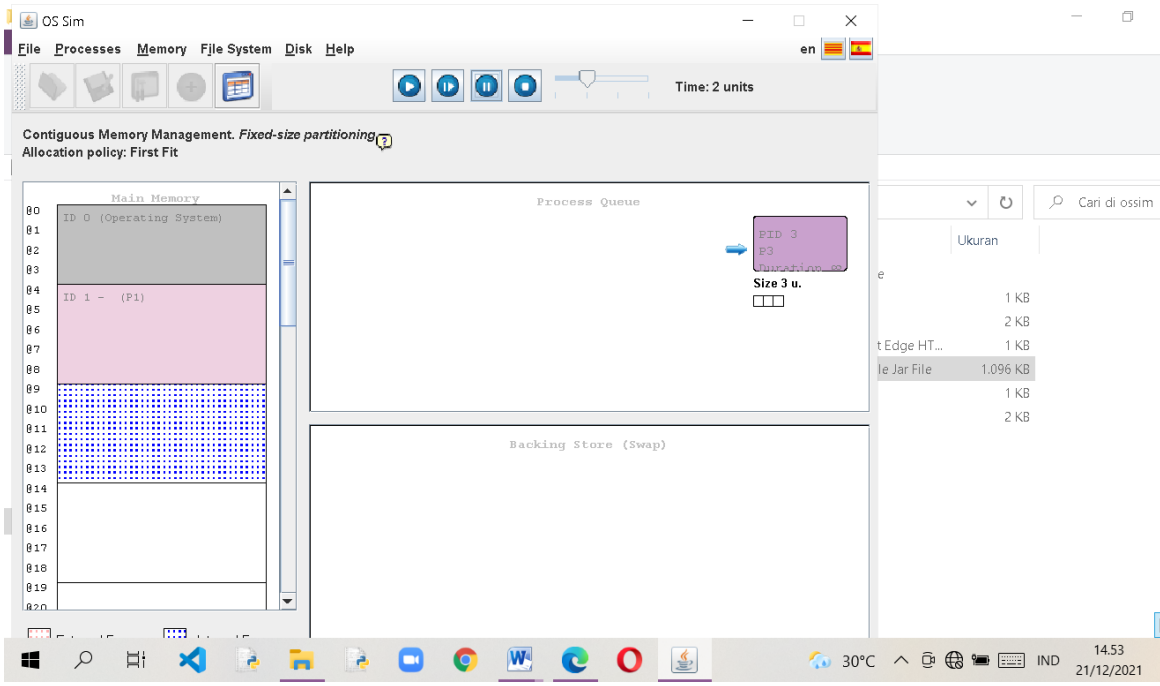

OS Sim
File Processes Memory File System Disk Help en
Help
Memory Management Examples
Content
Contiguous memory management with fixed-size partitions (first adjustment)
Contiguous memory management with fixed-size partitions (best fit)
Contiguous memory management with variable-sized partitions (Defragmentation)
Contiguous memory management with variable-sized partitions (Swap)
Pagination
Segmentation, partial allocation



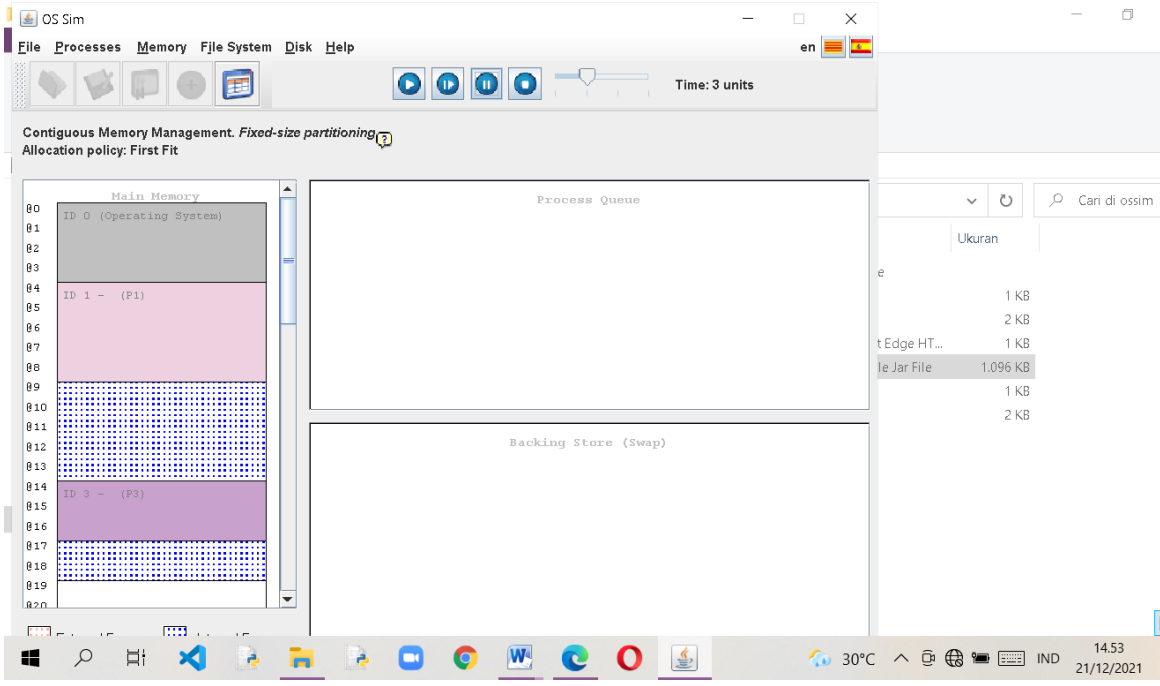
b. Menekan tombol step by step (time: 1 units).



c. Menekan tombol step by step (time: 2 units).



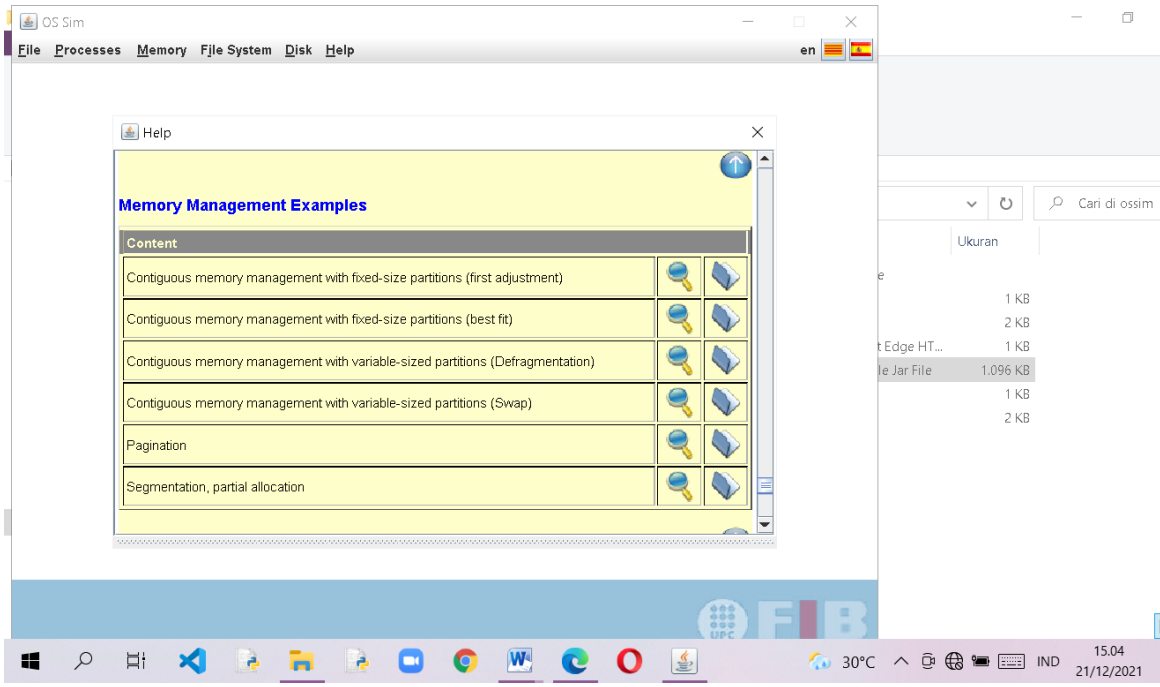
d. Menekan tombol step by step (time: 3 units).

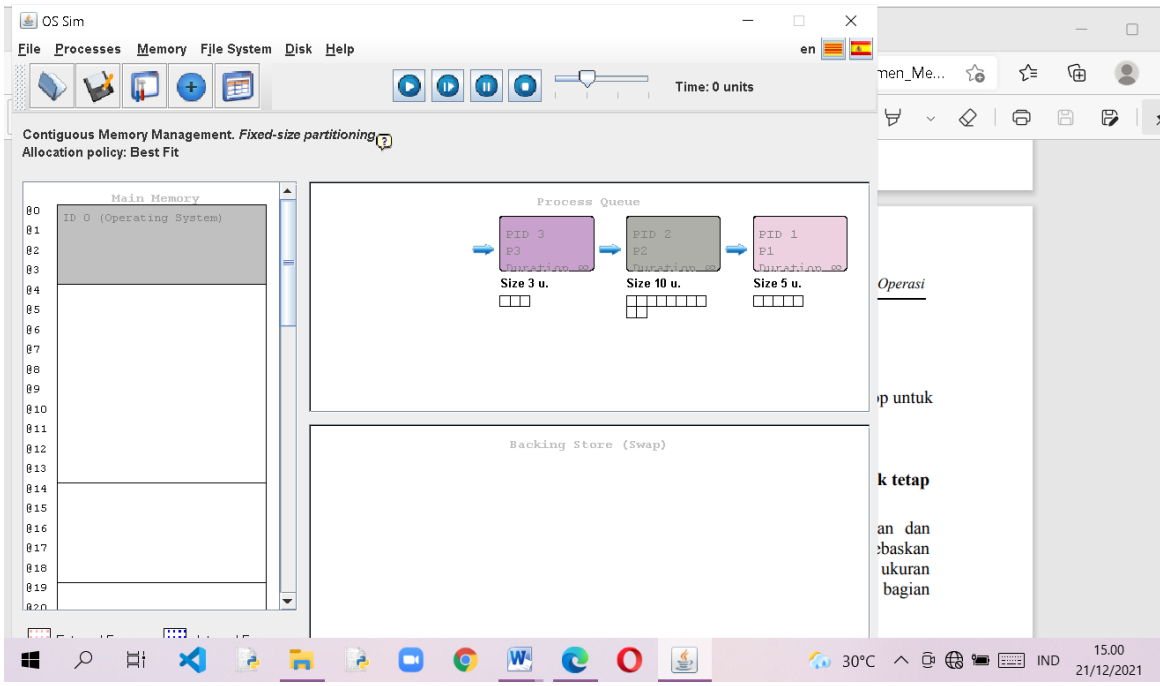


e. Dengan aturan first fit, banyak terjadi internal fragmentasi. Ini terjadi karena ketika suatu proses menemukan partisi yang cukup untuk dimuat, maka ia akan dimuat di partisi tersebut tanpa memandang apakah sisa partisi banyak atau tidak.

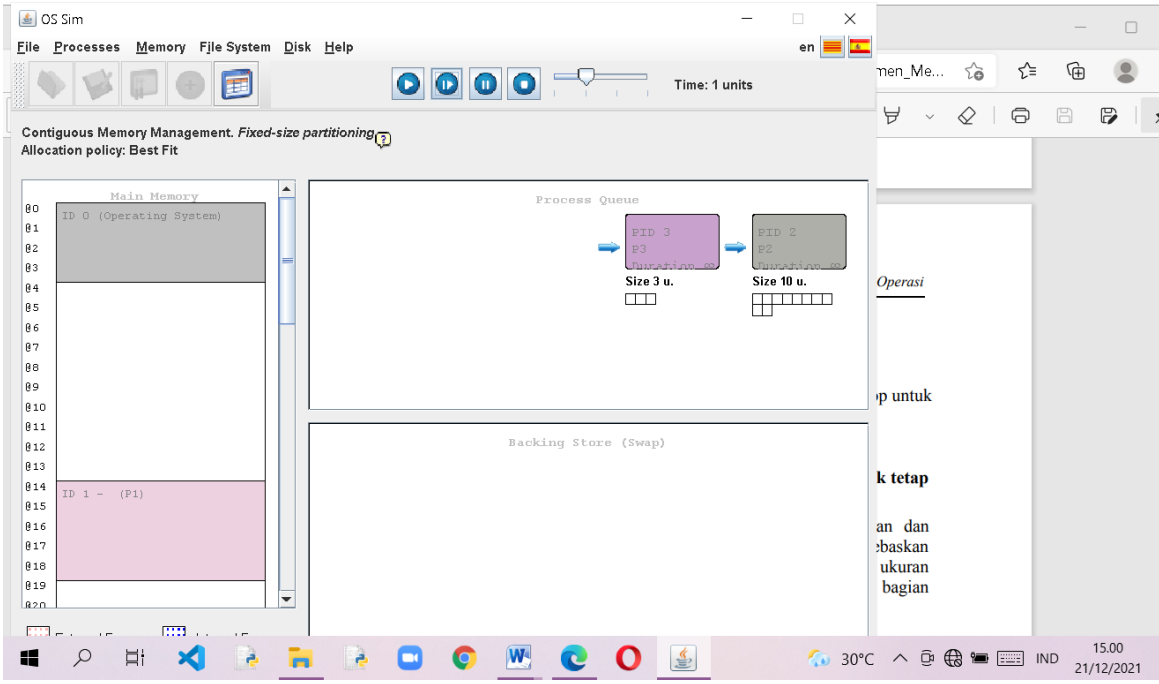
2.2 Contiguous Memory Management dengan menggunakan fixed-size partition dan aturan best fit.

a. Mengulangi langkah sebelumnya dan mengganti ke Contiguous Memory Management fixed-size partition with best fit.

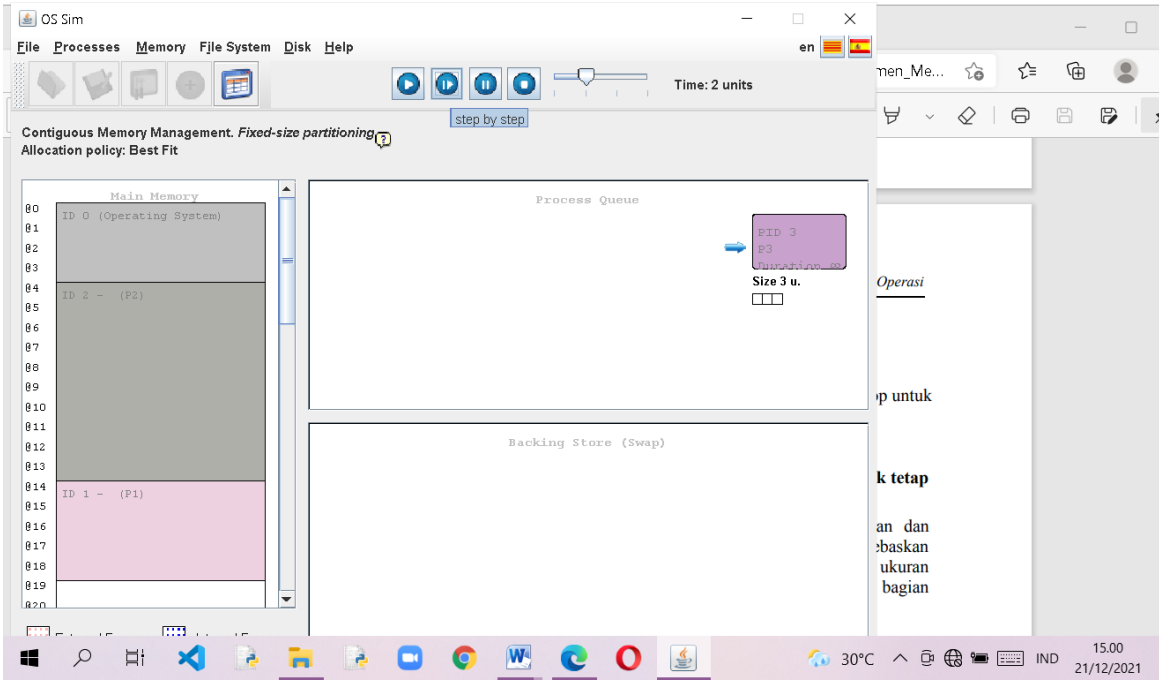




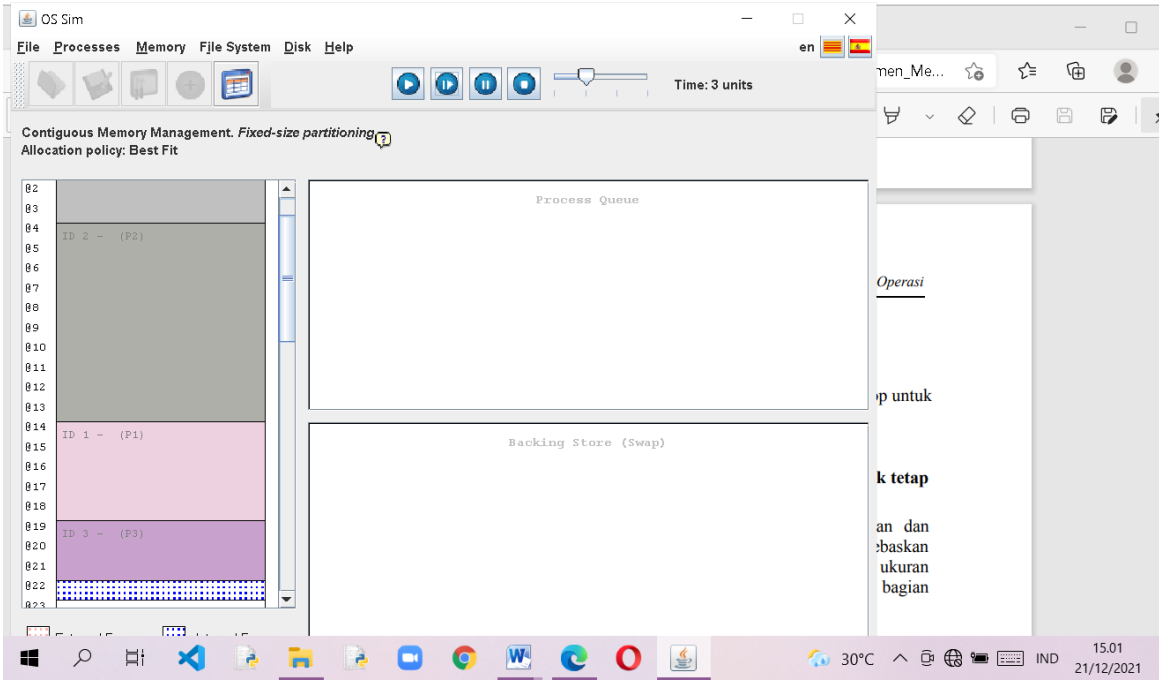
b. Menekan tombol step by step (time: 1 units).



c. Menekan tombol step by step (time: 2 units).



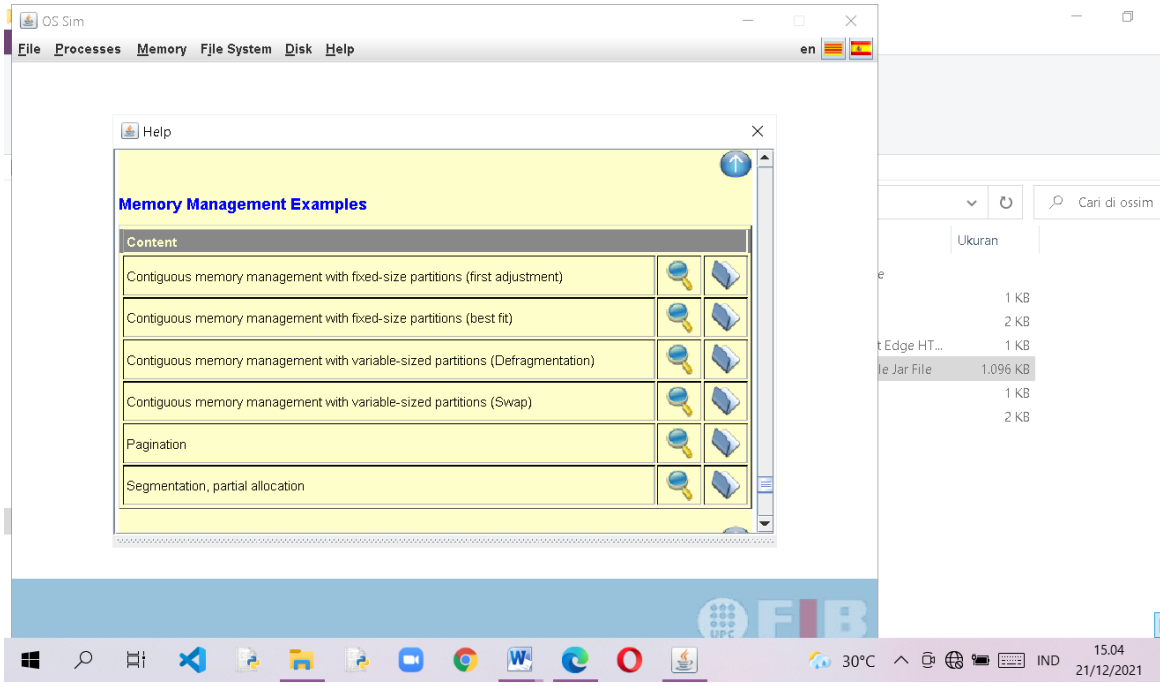
d. Menekan tombol step by step (time: 3 units).

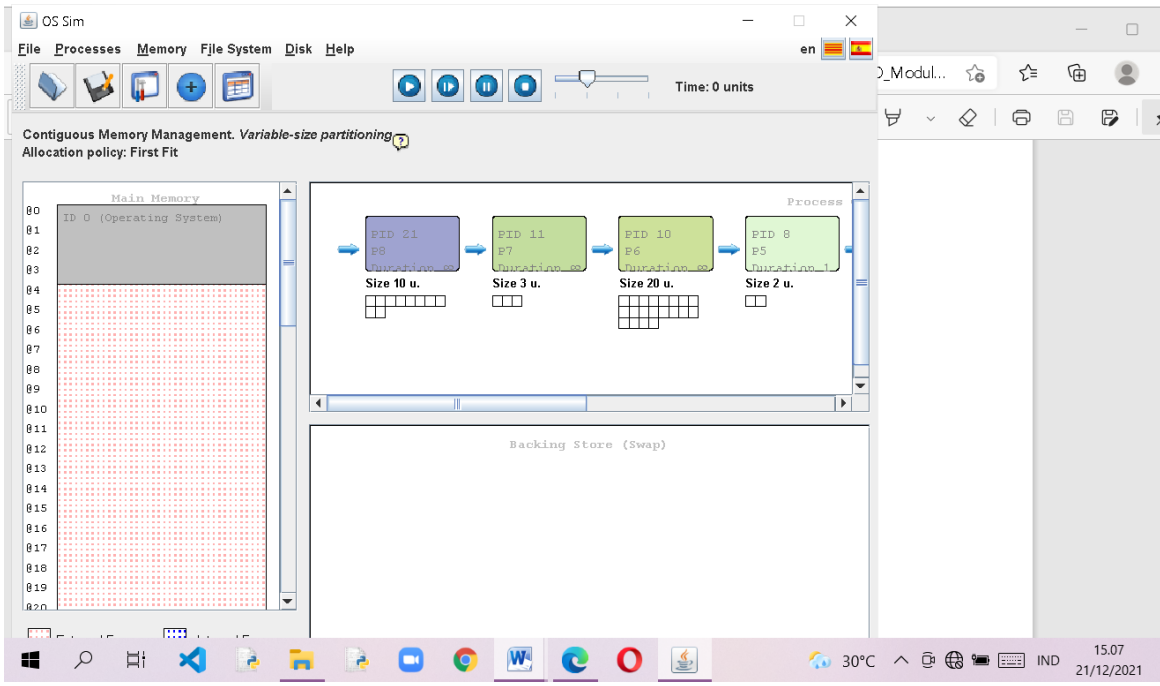


e. Dengan menggunakan aturan best fit, internal fragmentasi berkurang. Ini terjadi karena proses dimuat pada ukuran partisi terkecil akan tetapi masih dapat memuat proses tersebut sehingga internal fragmentasi dapat diminimalisasi.

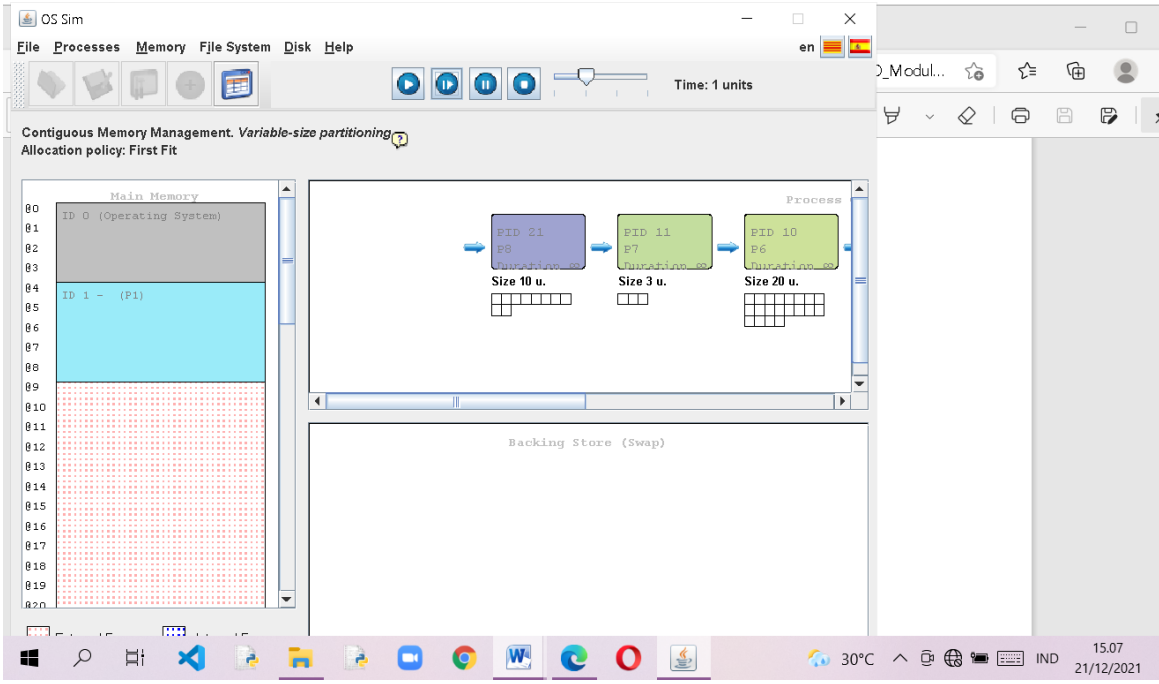
2.3Contiguous Memory Management dengan menggunakan variable-size partition dengan defragmentasi

a. Mengubah tipe manajemen memori ke partisi berukuran tidak tetap dengan defragmentasi.

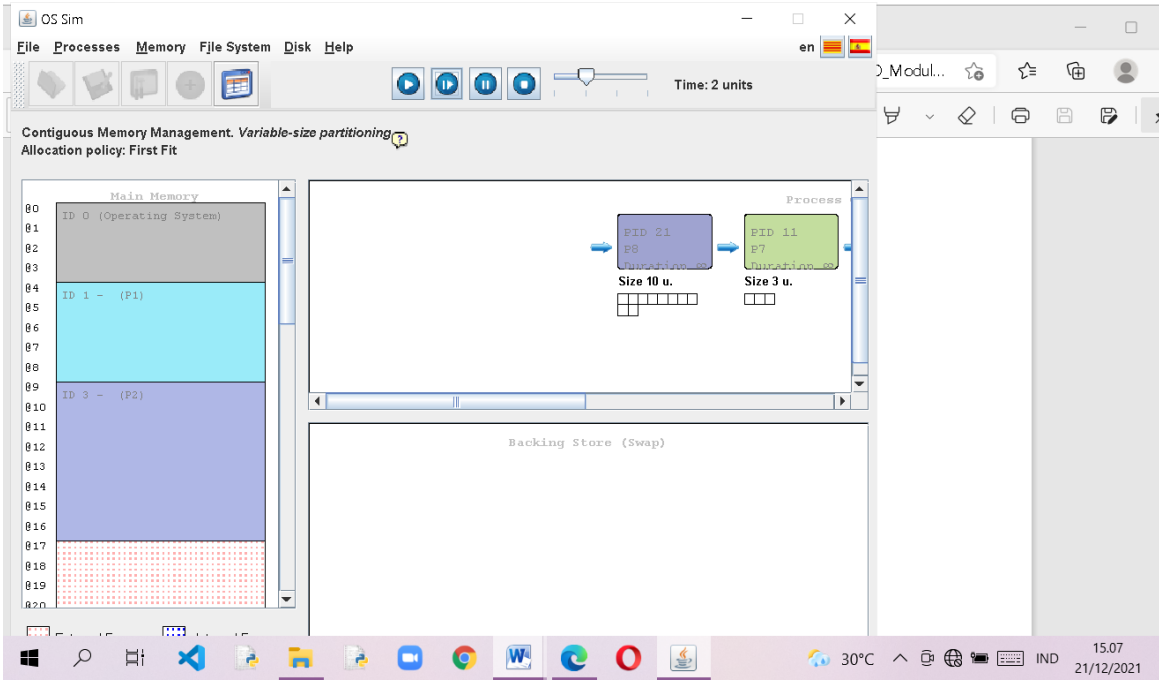




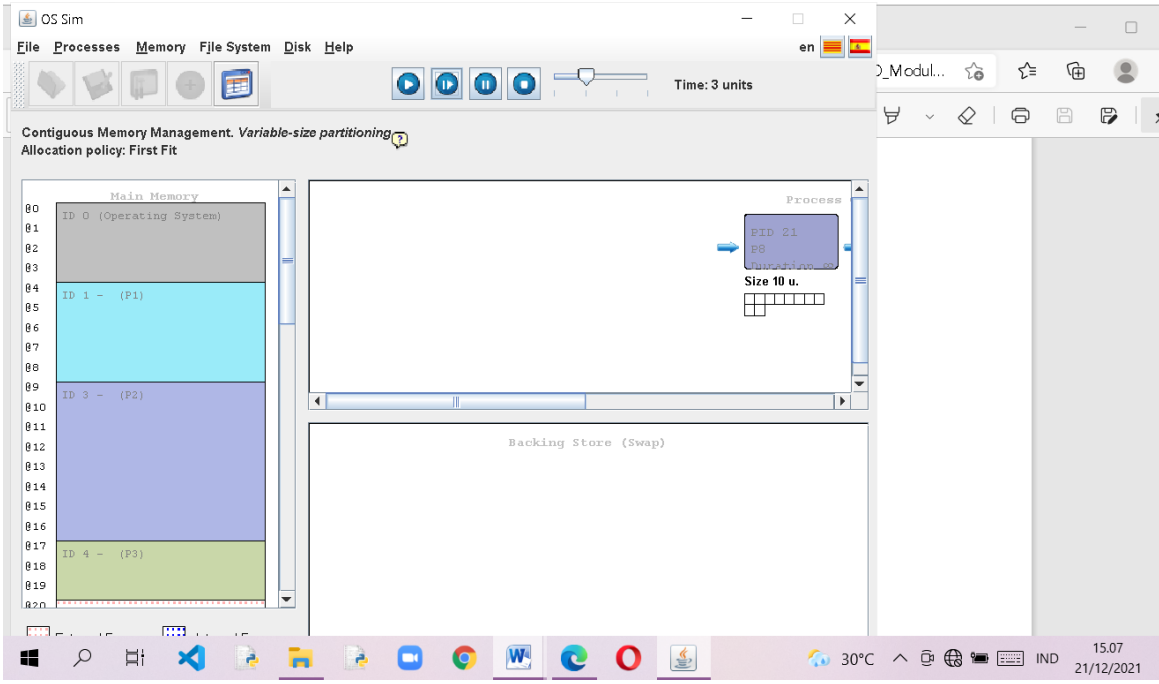
b. Menekan tombol step by step (time: 1 units).



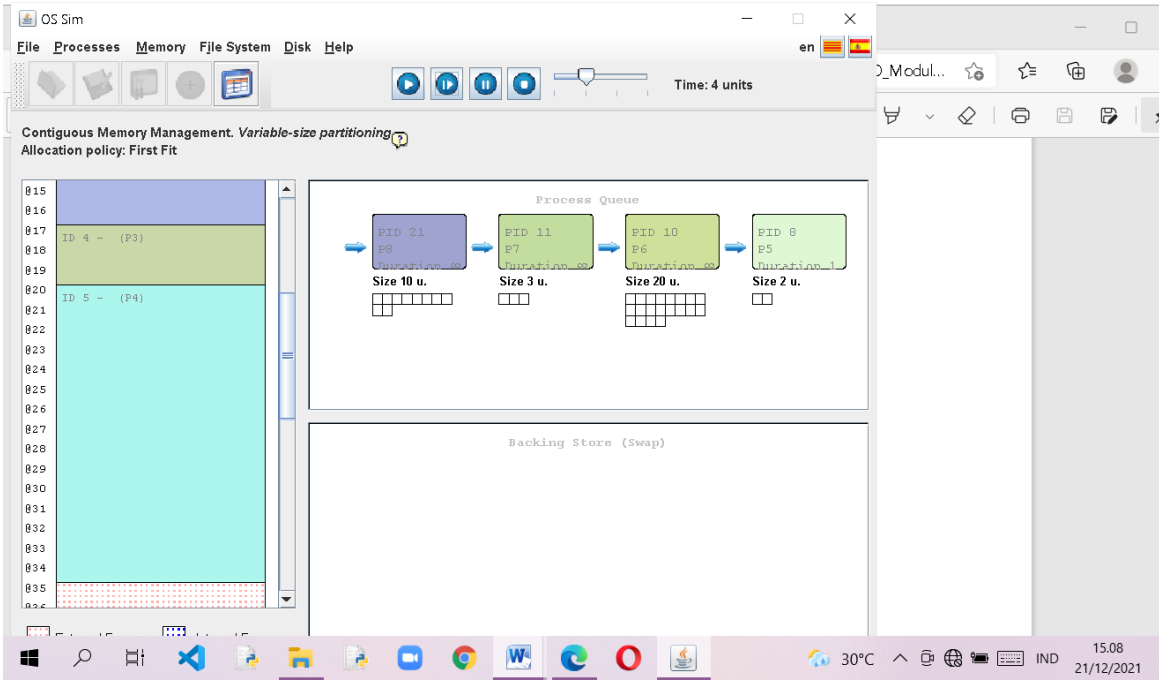
c. Menekan tombol step by step (time: 2 units).



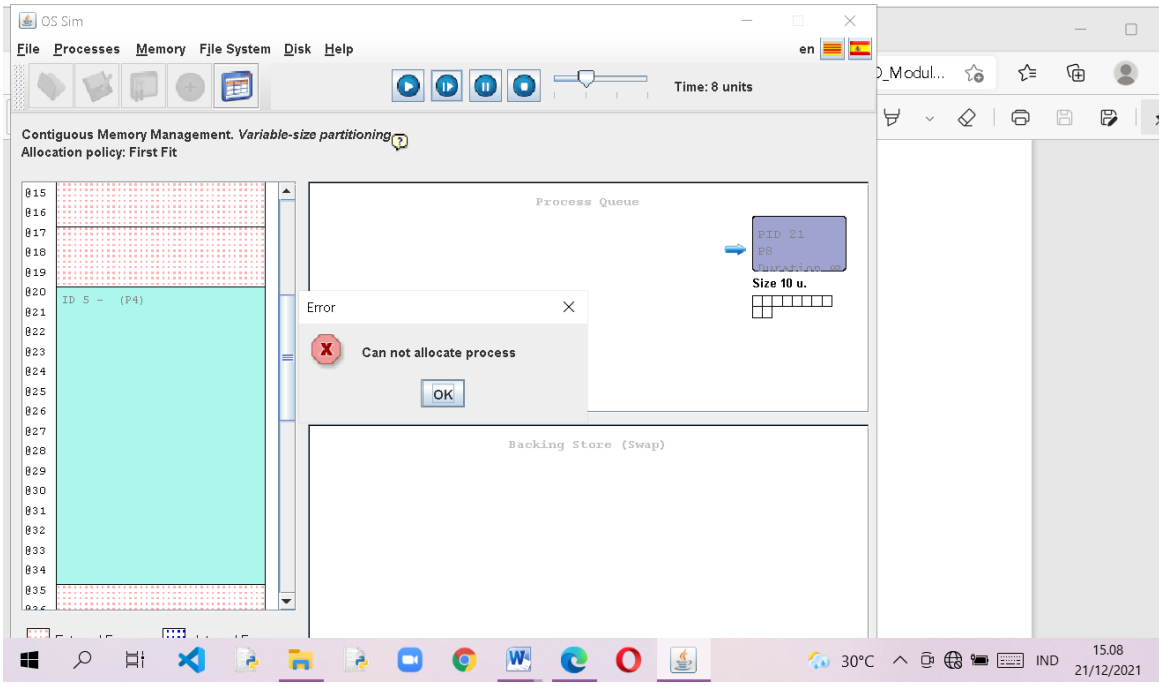
d. Menekan tombol step by step (time: 3 units).



e. Menekan tombol step by step (time: 4 units).



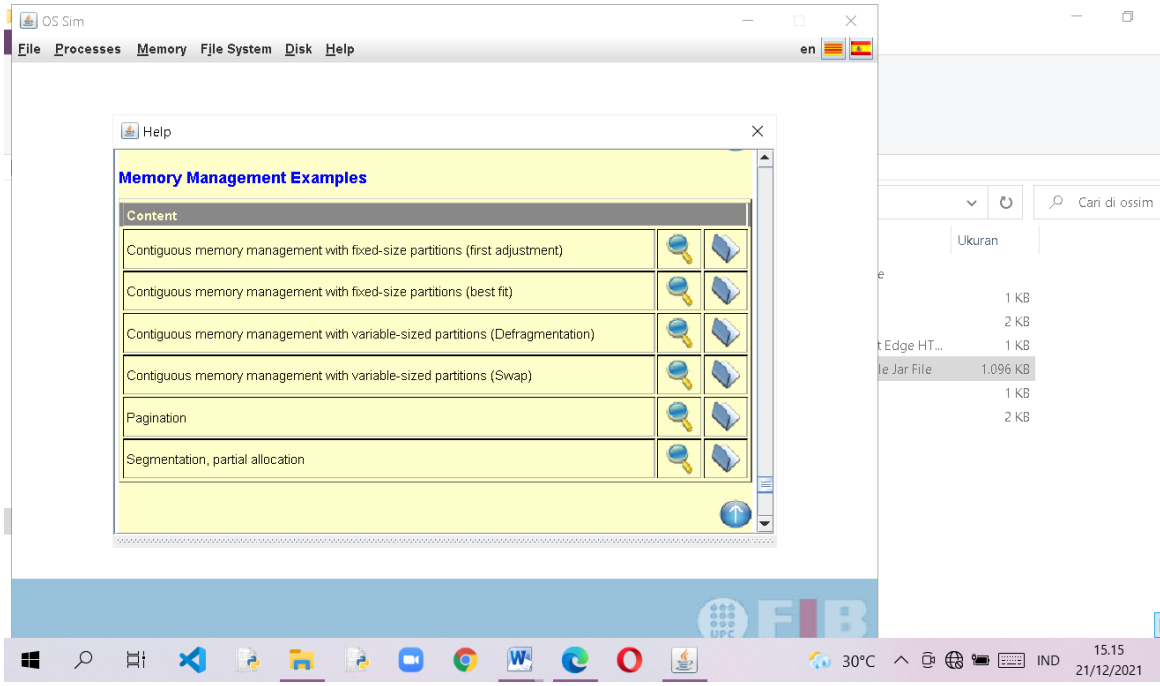
f. Menekan tombol step by step (time: 8 units).

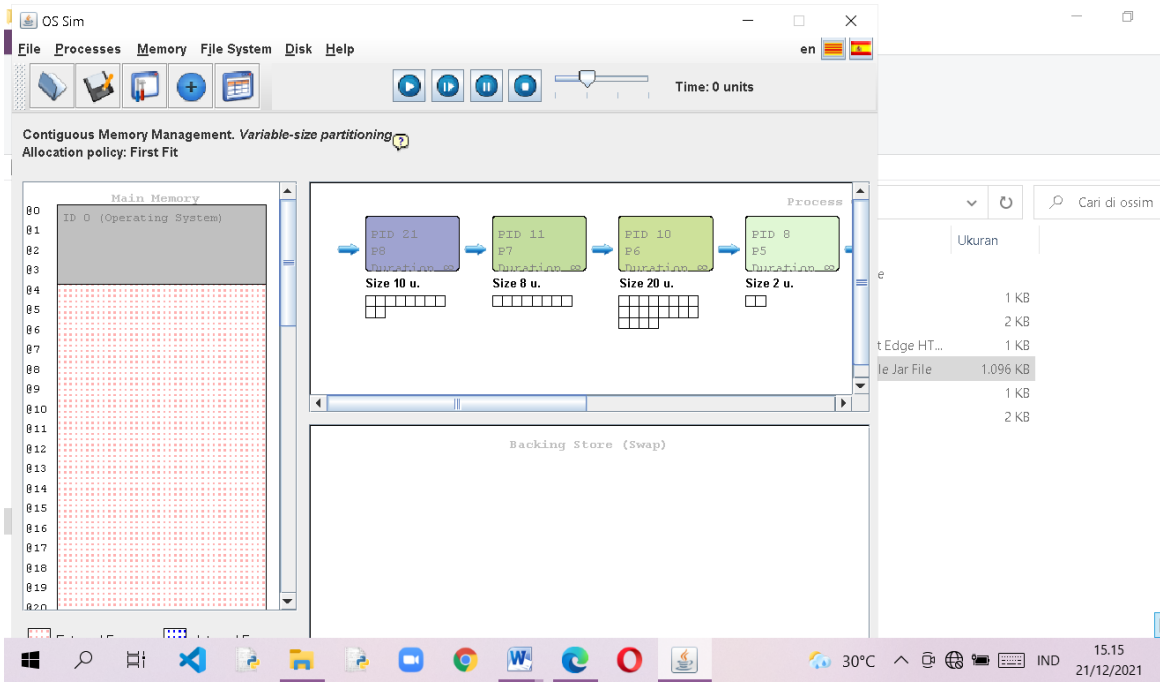


g. Pada step ke-8 terdapat sebuah proses yang tidak dapat dimuat. Ini terjadi karena pembuatan partisi berdasarkan proses yang dimuat pertama kali. Dan ternyata tidak ada partisi yang dapat memuat proses terakhir tersebut.

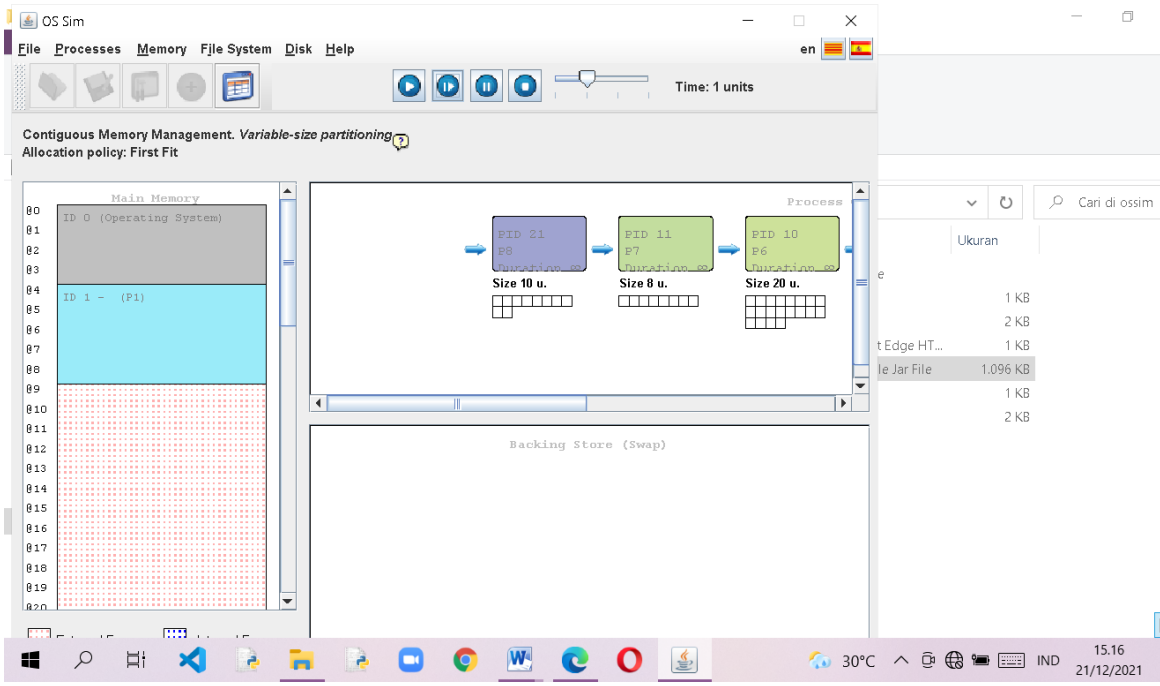
2.4Contiguous Memory Management dengan menggunakan variable-size partition dengan swap.

a. Mengubah tipe manajemen memori ke partisi berukuran tidak tetap dengan swap.

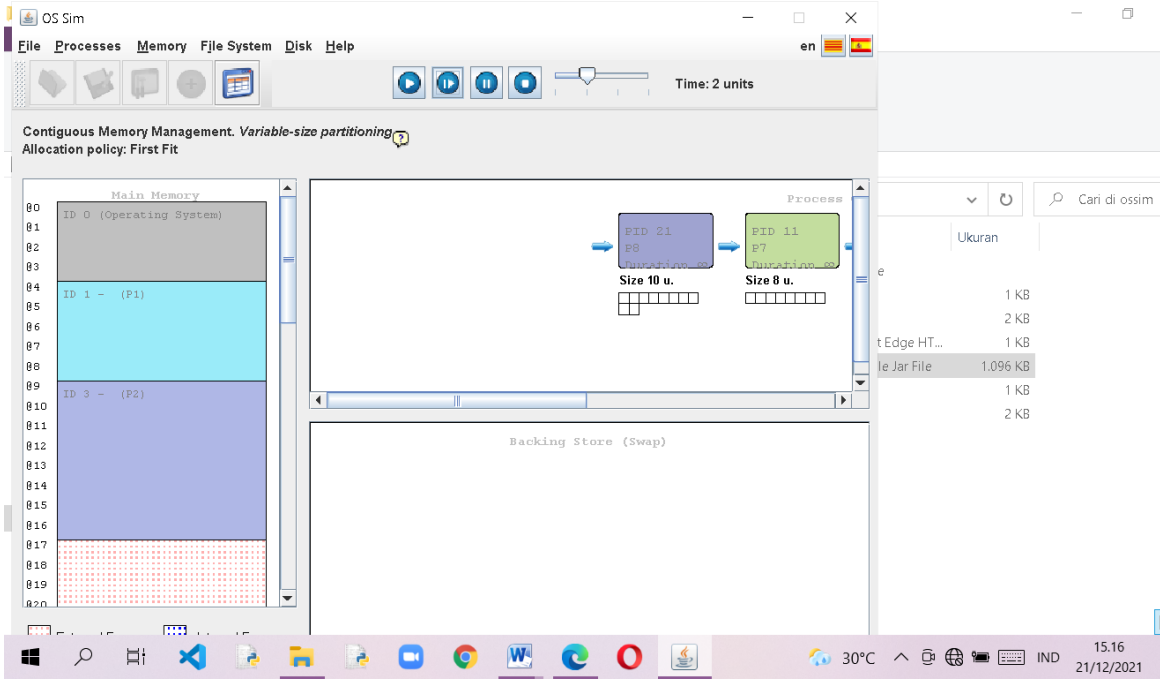




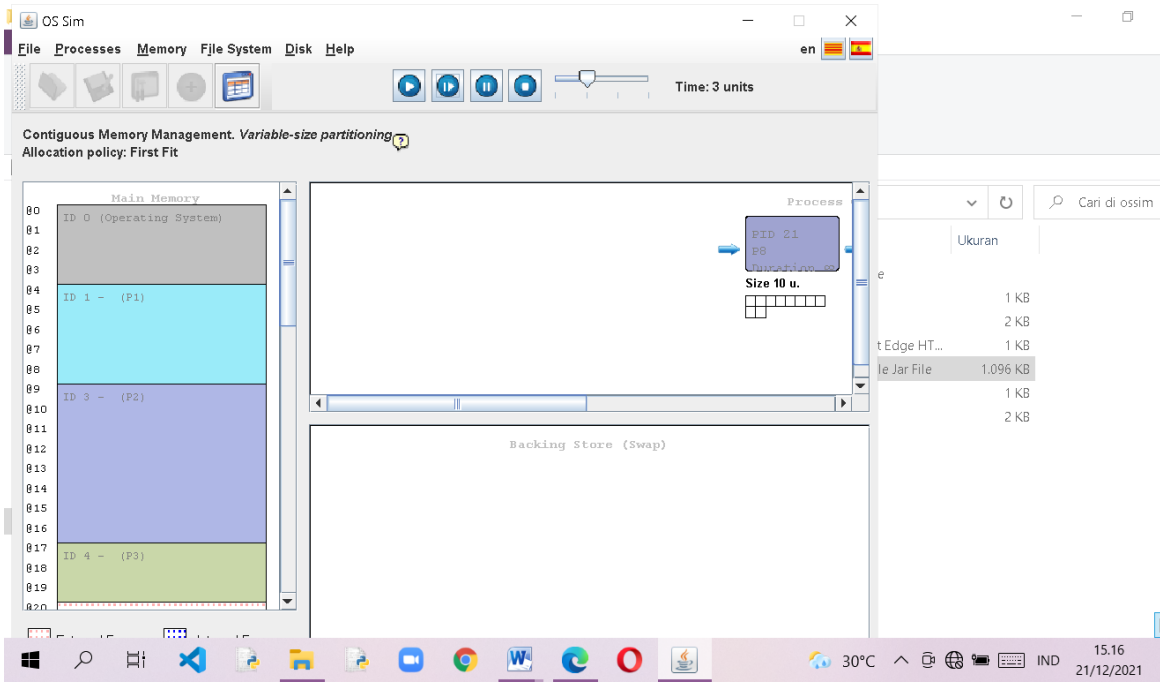
b. Menekan tombol step by step (time: 1 units).



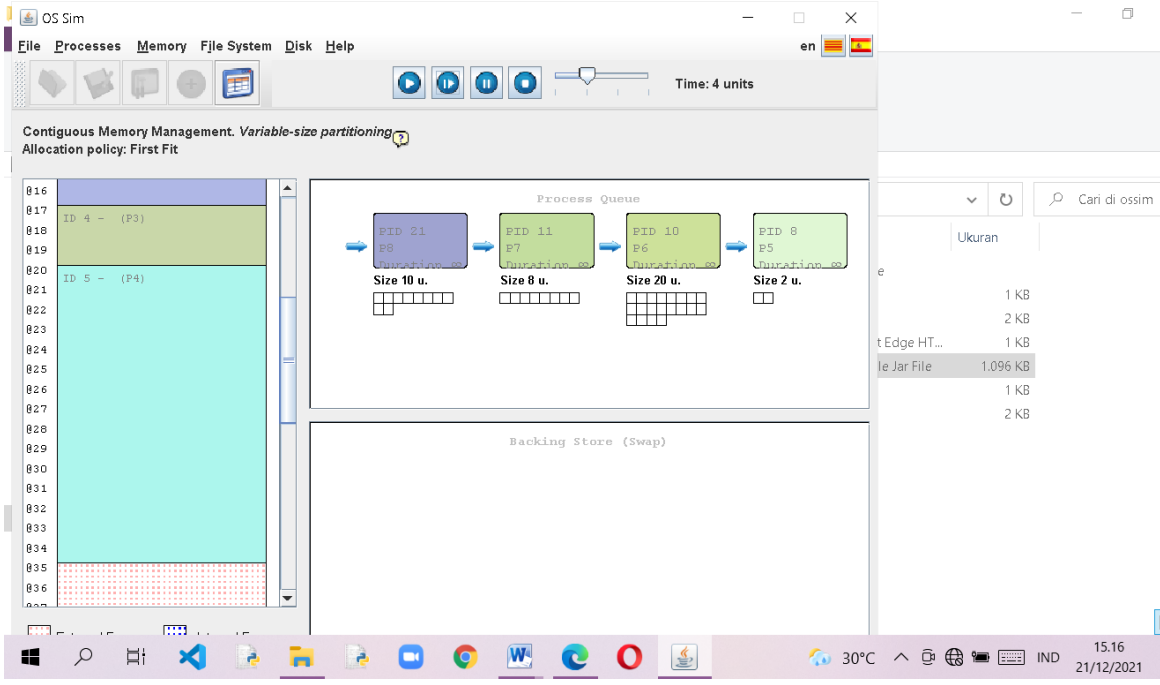
c. Menekan tombol step by step (time: 2 units).



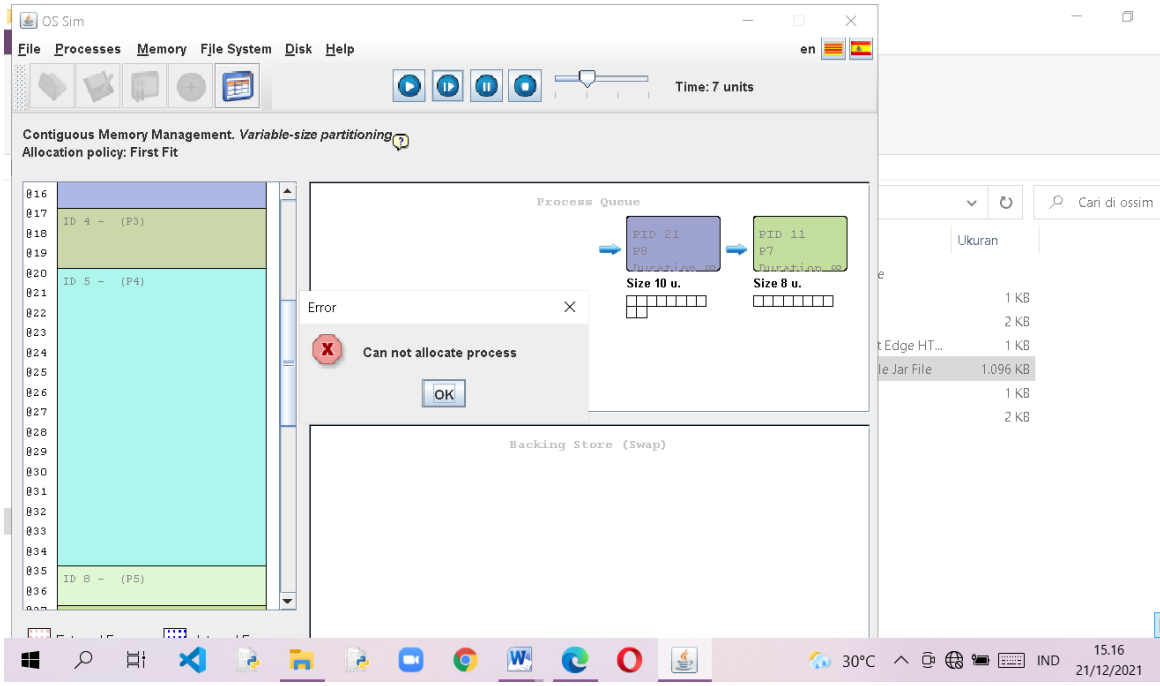
d. Menekan tombol step by step (time: 3 units).



e. Menekan tombol step by step (time: 4 units)



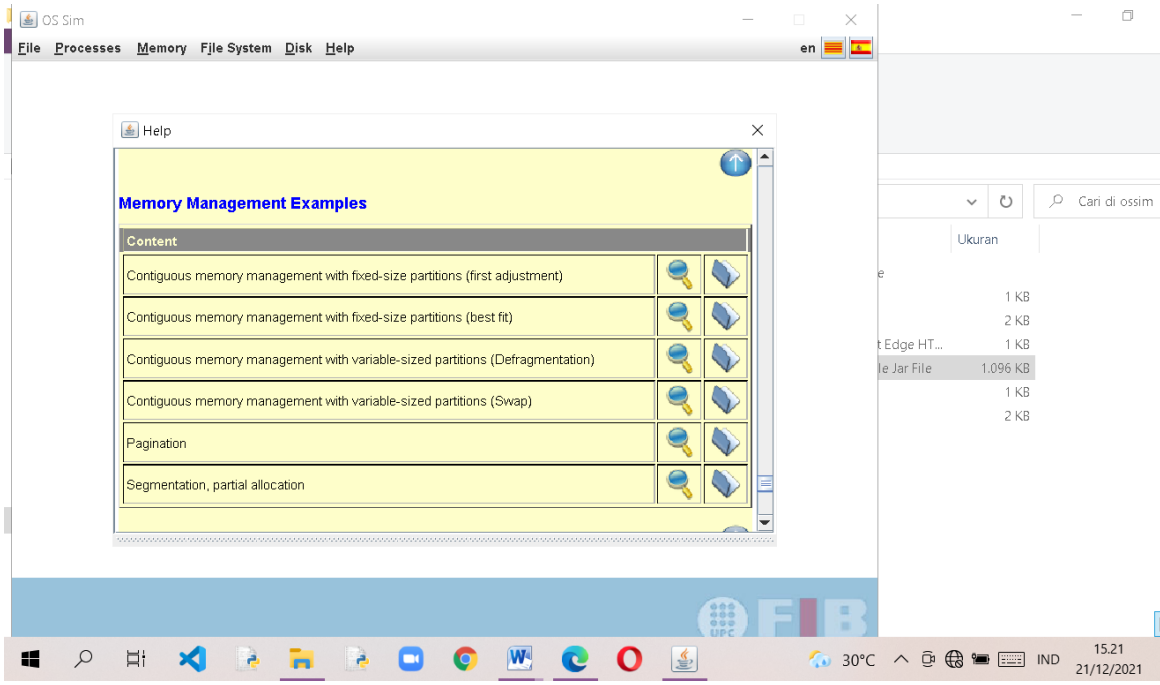
f. Menekan tombol step by step (time: 7 units).

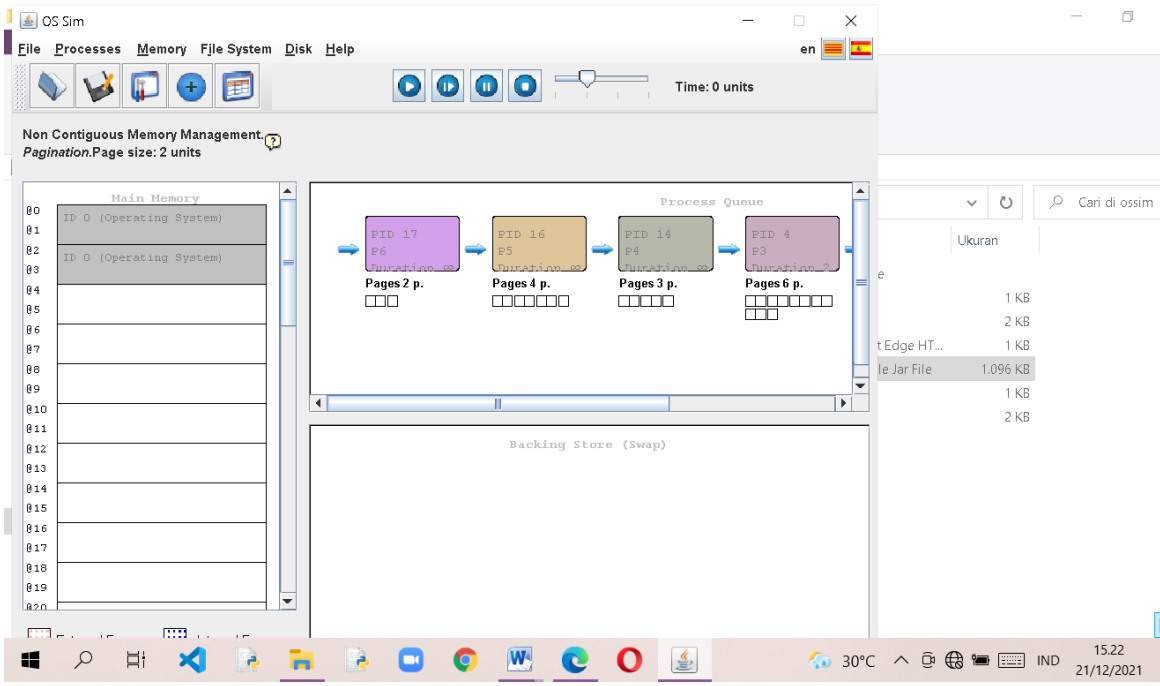


g. Pada step ke-7 terdapat 2 proses yang tidak dapat dialokasikan. Hal ini mungkin terjadi karena tidak yang tidak aktif sehingg tidak terjadi swap dan partisi tidak ada yang mencukupi untuk memuat proses yang tersisa.

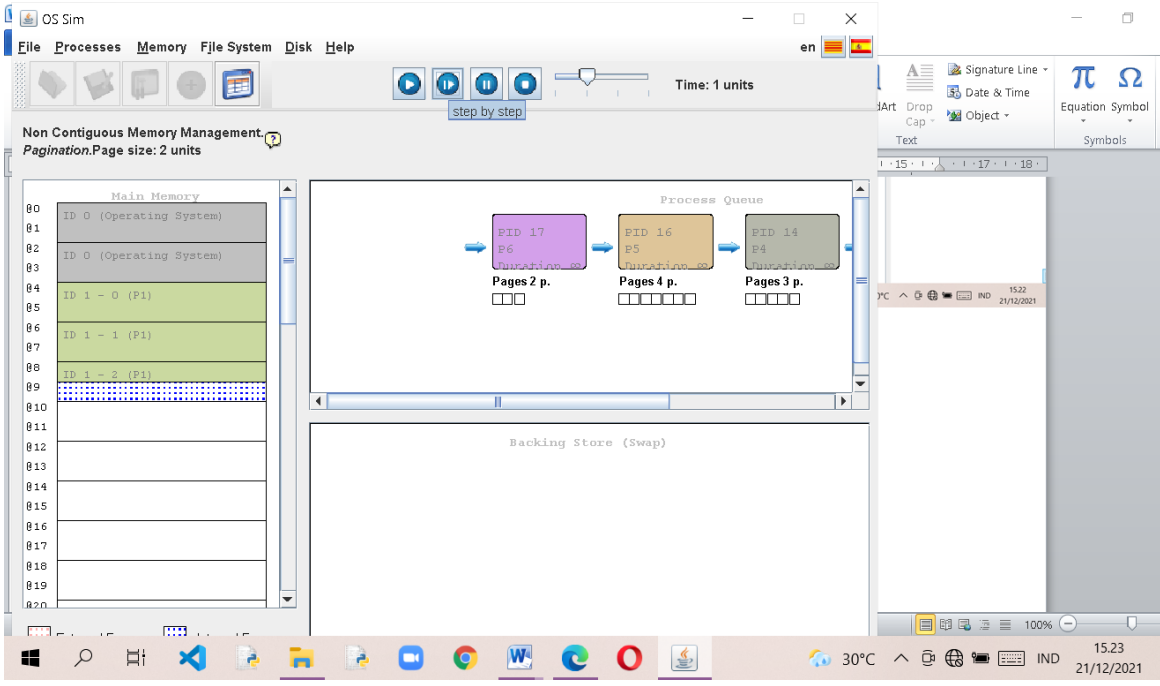
2.5Pagination (Ukuran page 2 unit).

a. Mengubah tipe manajemennya ke Pagination.

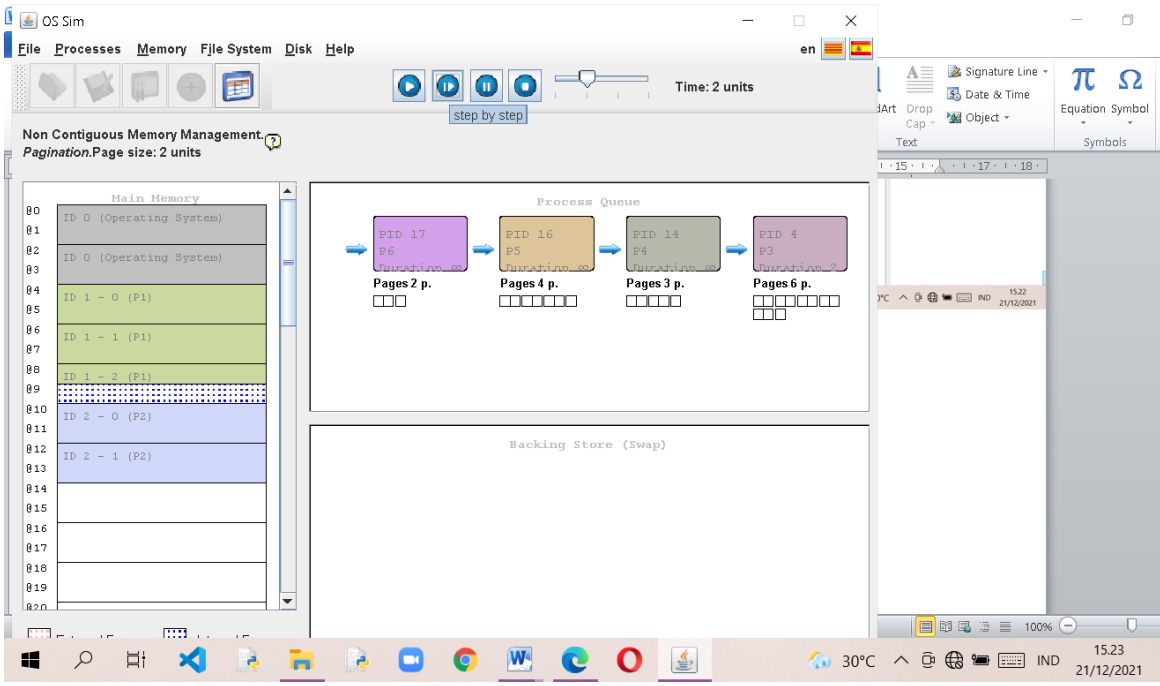




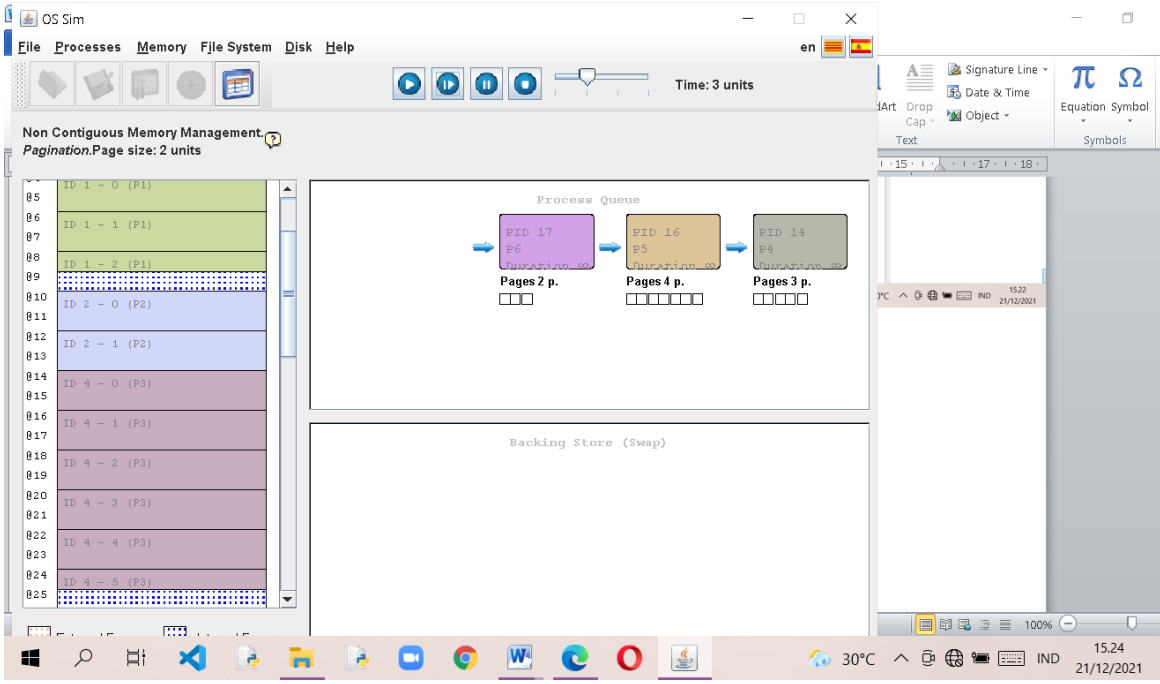
b. Menekan tombol step by step (time: 1 units).



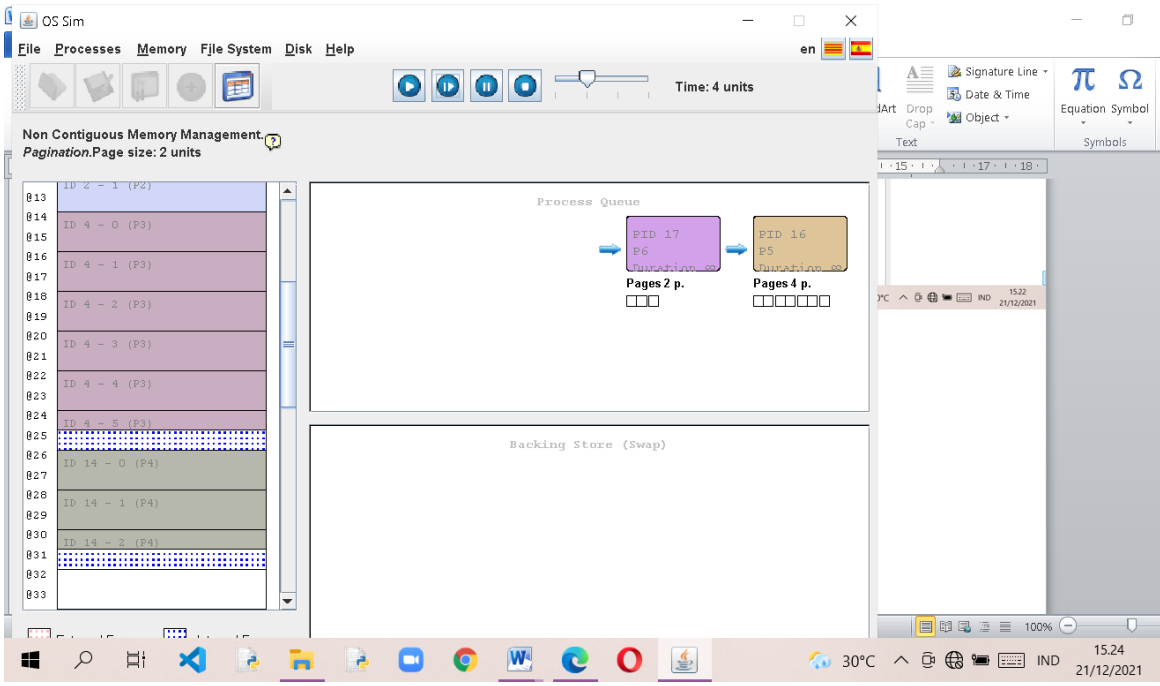
c. Menekan tombol step by step (time: 2 units).



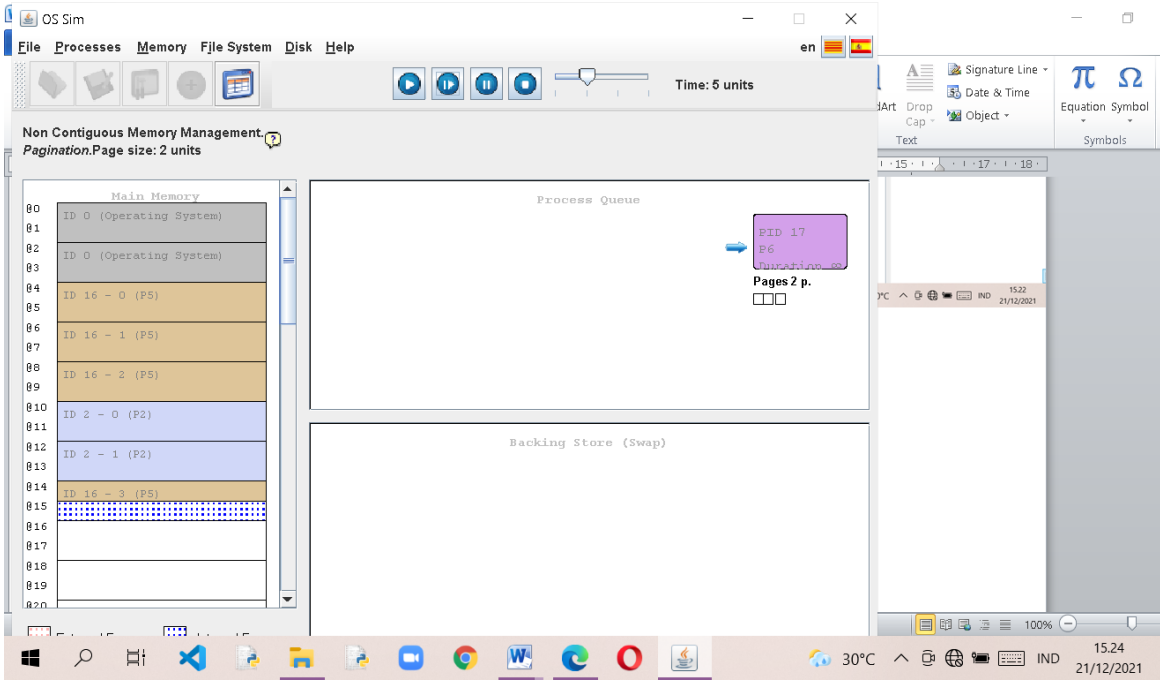
d. Menekan tombol step by step (time: 3 units).



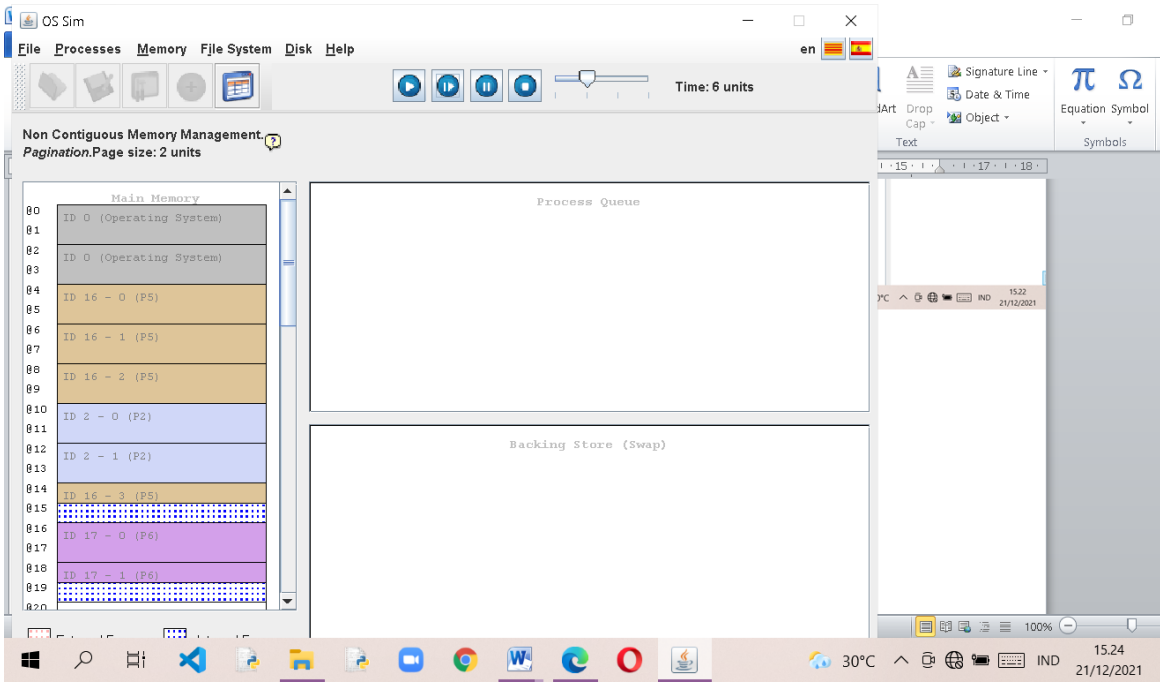
e. Menekan tombol step by step (time: 4 units).



f. Menekan tombol step by step (time: 5 units).



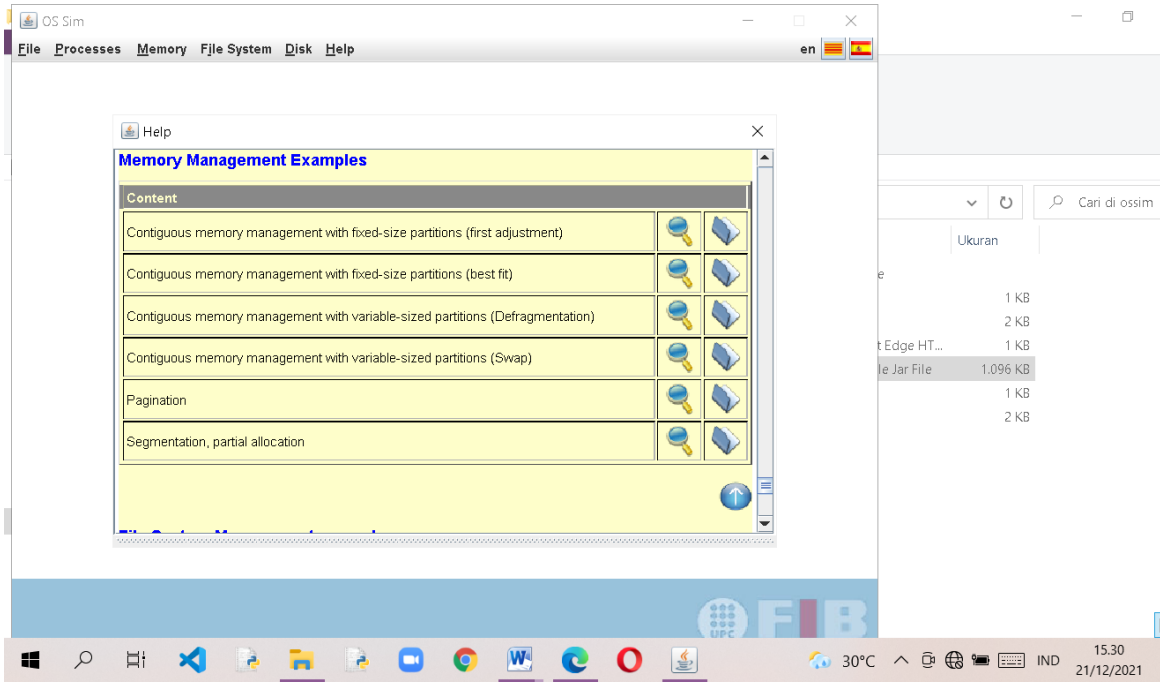
g. Menekan tombol step by step (time: 6 units).

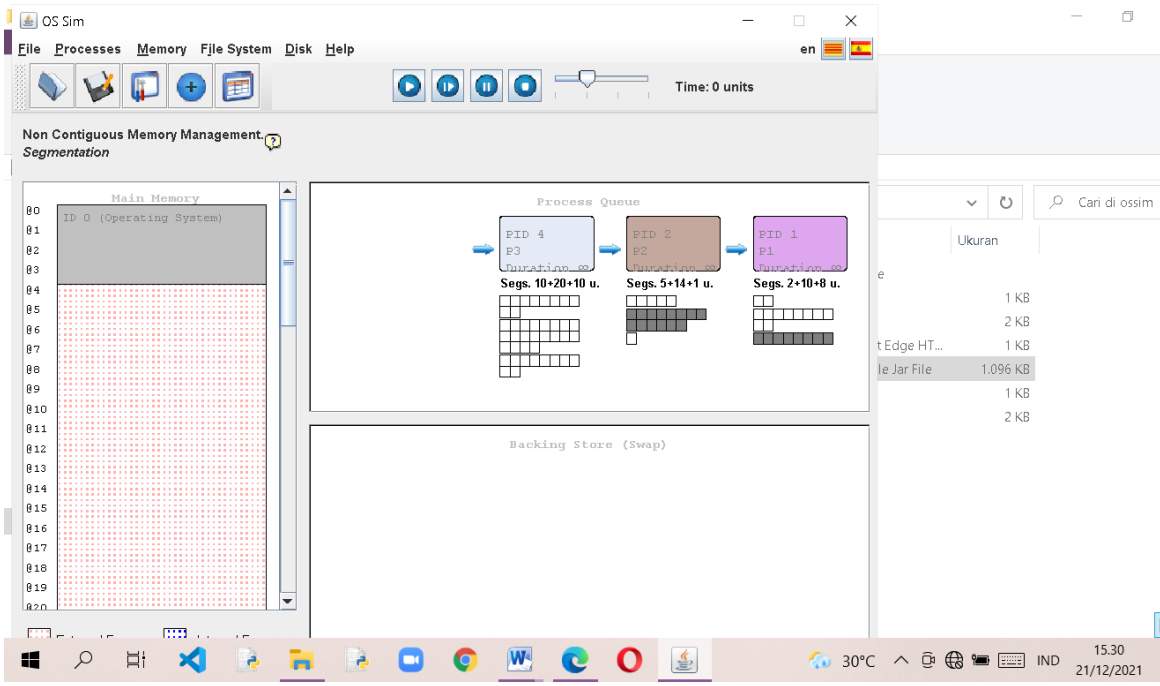


h. Terlihat pada percobaan ini bahwa setiap proses dibagi menjadi beberapa page dengan ukuran tetap setiap pagenya, yaitu 2 unit. Dan terlihat pada gambar tersebut bahwa terjadi internal fragmentasi pada proses yang memiliki unit ganjil karena terdapat page yang seharusnya berisi 2 unit akan tetapi hanya terisi 1 unit saja.

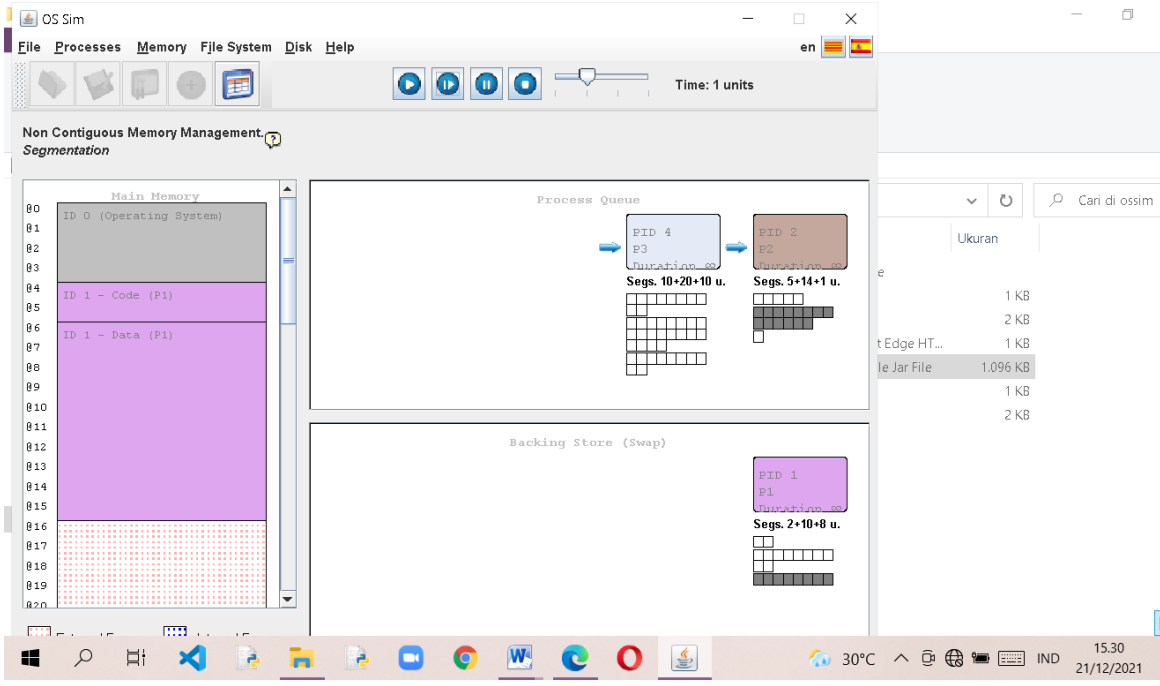
2.6Segmentation (Alokasi parsial)

a. Mengubah tipe manajemen memori ke Segmentation.

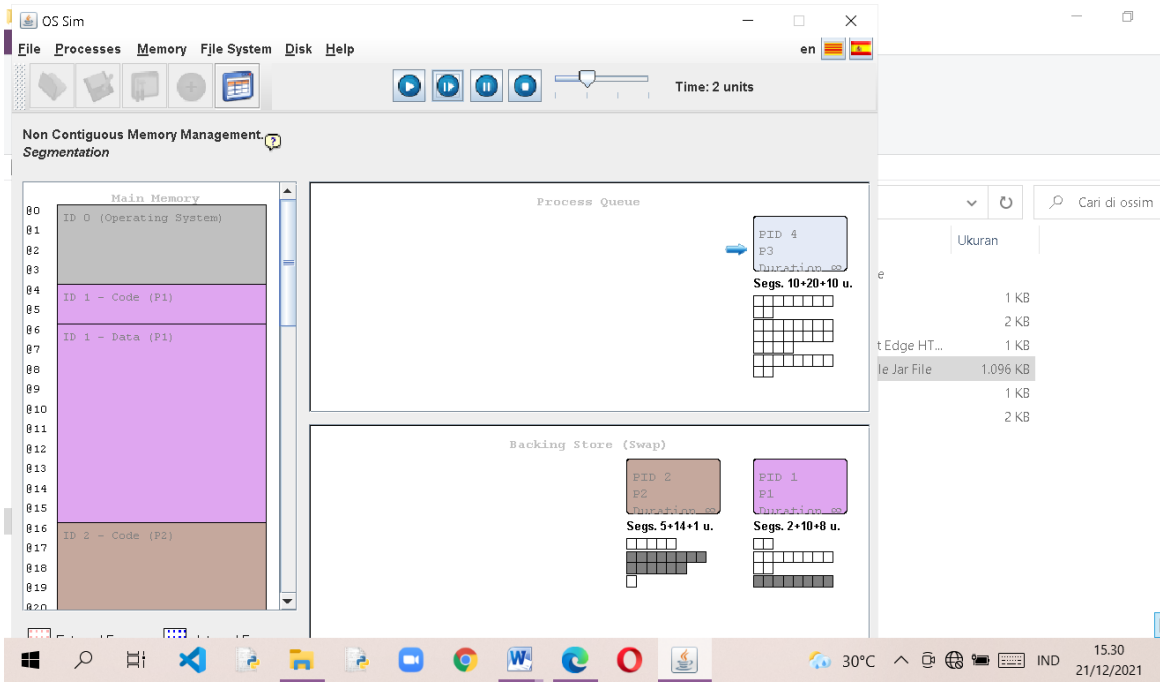




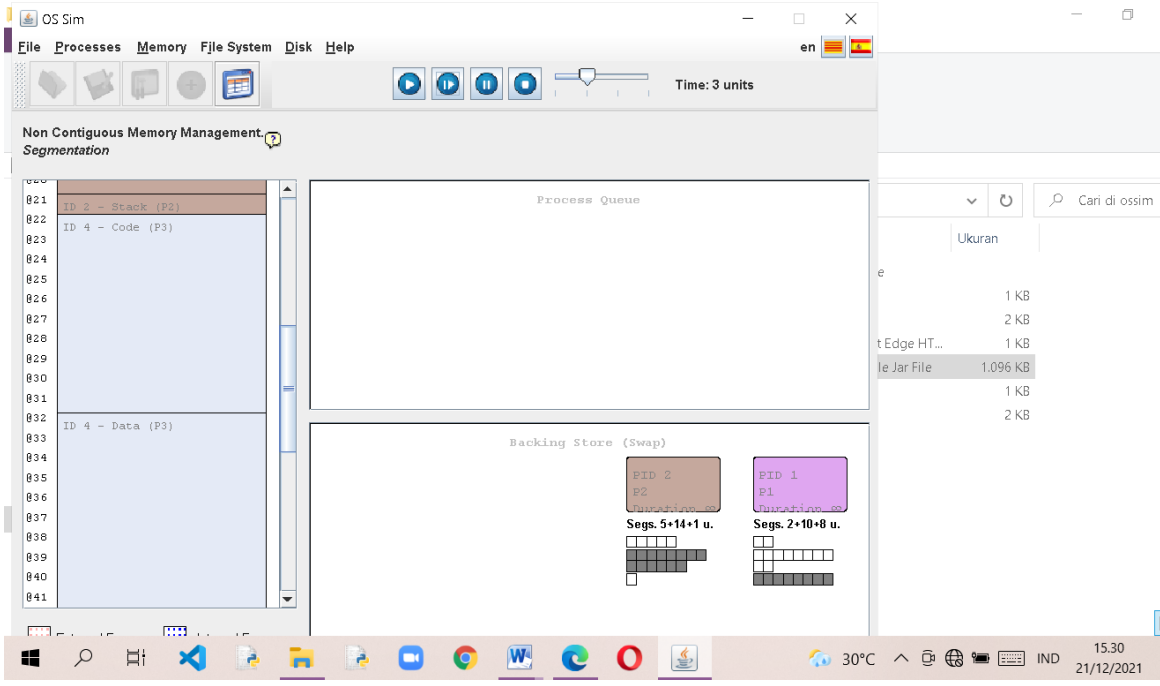
b. Menekan tombol step by step (time: 1 units).



c. Menekan tombol step by step (time: 2 units).



d. Menekan tombol step by step (time: 3 units).



e. Pada metode terakhir ini, 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.