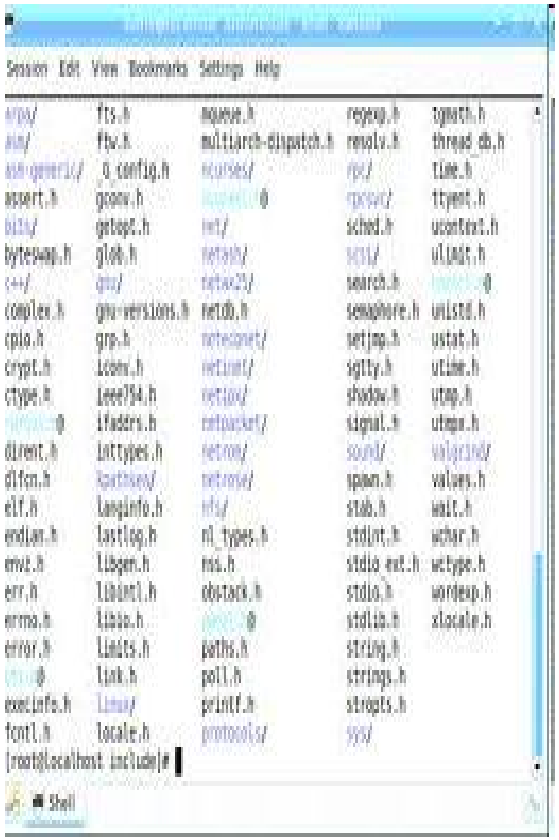
Penjadwalan Proses

DESKRIPSI PENJADWALAN PROSES

- Kumpulan kebijaksanaan dan mekanisme
- Urutan kerja yang dilakukan sistem komputer
- Mengatur :
  - Proses yang harus berjalan
  - Kapan & selama berapa lama proses itu berjalan



Kriteria untuk mengukur dan optimasi kinerje penjadwalan

- Adil (*fairness)*
- Efisiensi (*eficiency)*
- Waktu tanggap (response time)
- *Turn around time*

Adil (*fairness*)

Adalah proses-proses yang diperlakukan sama, yaitu mendapat jatah waktu pemroses yang sama dan tak ada proses yang tak kebagian layanan pemroses sehingga mengalami kekurangan waktu.

Efisiensi (*eficiency)*

Efisiensi atau utilisasi pemroses dihitung dengan perbandingan (rasio) waktu sibuk pemroses.

Waktu tanggap (response time)

- Untuk Sistem interaktif didefinisikan sebagai waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukkan atau transaksi sampai hasil pertama muncul di layar. Waktu tanggap ini disebut terminal *response time.*
- *Untuk sistem waktu nyata Didefinisikan sebagai waktu dari saat* kejadian (internal atau eksternal) sampai instruksi pertama rutin layanan yang dimaksud dieksekusi, disebut *event.*

Turn around time

Adalah waktu yang dihabiskan dari saat program atau job mulai masuk ke sistem sampai proses diselesaikan sistem.

Waktu yang dimaksud adalah waktu yang dihabiskan di dalam sistem, diekspresikan sebagai penjumlah waktu eksekusi (waktu pelayanan job) dan waktu menunggu, yaitu :

Turn arround time = Burst time + Wait time.

Throughput

Adalah jumlah kerja yang dapat diselesaikan dalam satu unit waktu.

Cara untuk mengekspresikan throughput adalah dengan jumlah job pemakai yang dapat dieksekusi dalam satu unit/interval waktu.

Tipe-tipe Penjadwalan

- Penjadwalan jangka pendek
  (short-term scheduller)

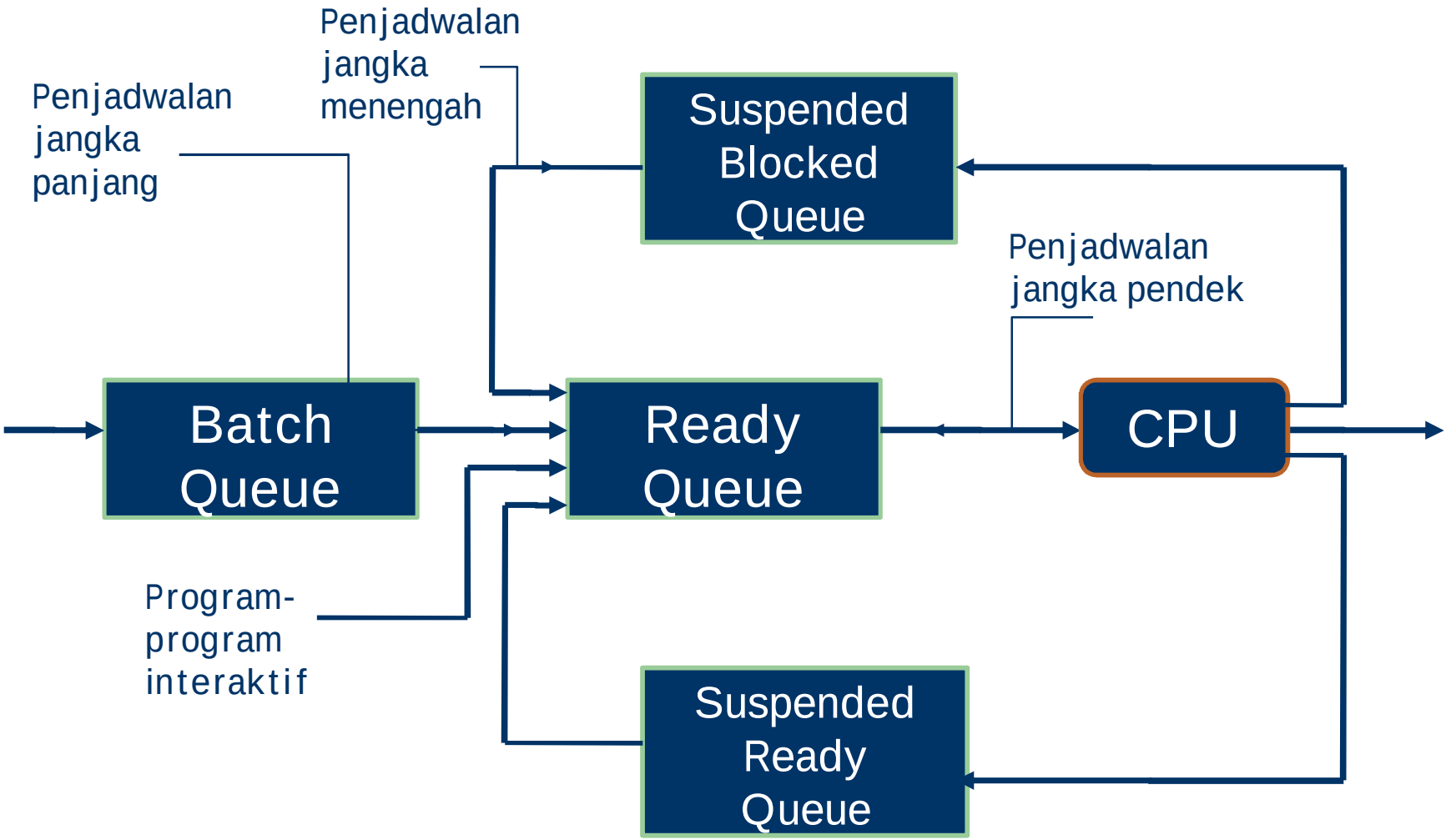
  Penjadwalan ini bertugas menjadwalkan alokasi pemroses diantara proses-proses ready di memori utama.

- Penjadwalan jangka menengah (medium term scheduller)

  Penjadwalan jangka menengah adalah menangani proses-proses swapping (aktivitas pemindahan proses yang tertunda dari memory utama ke memory sekunder).

- Penjadwalan jangka panjang (long-term scheduller)

  Penjadwalan jangka panjang bekerja terhadap antrian batch (proses – proses dengan penggunaan sumberdaya yang intensif) dan memilih batch berikutnya yang harus di eksekusi.



Menujukan posisi dari tipe-tipe penjadwalan yang terdapat pada satu sistem operasi

strategi penjadwalan

- Penjadwalan Non Preemptive

  Jika proses sedang menggunakan CPU → proses tersebut akan membawa CPU sampai proses tersebut melepaskannya (berhenti dalam keadaan wait).

- Penjadwalan Preemptive

  Pada saat proses sedang menggunakan CPU → CPU dapat diambil alih oleh proses lain.

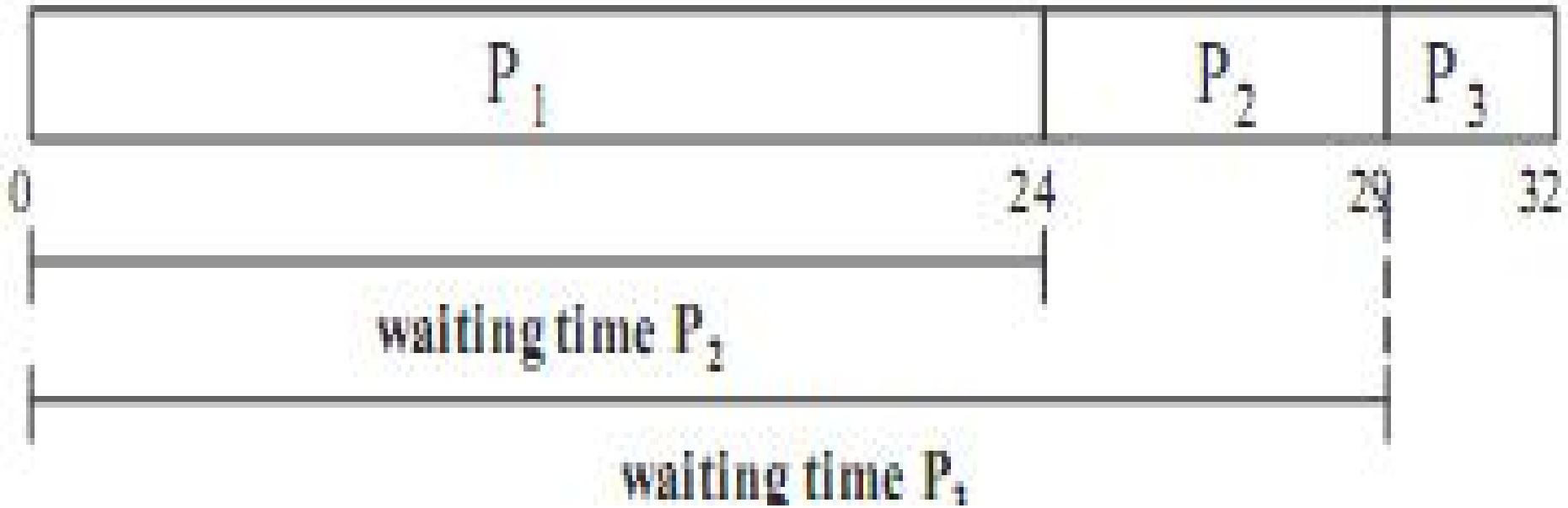
  Dalam hal ini harus selalu dilakukan perbaikan data.

FCFS *(First Come First Served)* Algorithm

- Penjadwalan ini merupakan penjadwalan *Non Preemptive.*
- Dalam penjadwalan FCFS *(First Come First Serve) :*
  - Proses yang pertama kali minta jatah waktu untuk menggunakan CPU akan dilayani terlebih dahulu.
  - Begitu proses mendapatkan jatah waktu CPU → proses dijalankan sampai selesai/ sampai proses tersebut melepaskannya, yaitu jika proses tersebut berhenti atau meminta I/O.

Contoh

Ada 3 buah proses yang datang secara berurutan yakni, P_1 selama 24 ms, P_2 selama 5 ms, P_3 selama 3 ms. Maka *Gantt Chart*-nya :



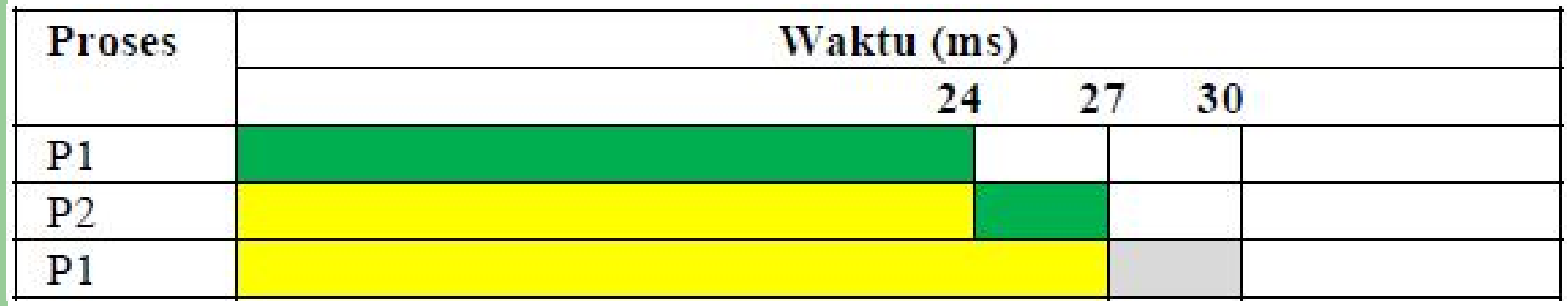
Contoh

- Diketahui proses-proses sebagai berikut

Process	*Arrival Time*	*Burst Time*
P1	0	24
P2	0	3
P3	0	3

- Hitunglah waiting time rata-rata dan *turnaround time (burst time + waiting time) dari* ketiga proses

- Urutan kedatangan adalah P1, P2 , P3
- gantt chart untuk urutan ini adalah:



- *Waiting time*

  P1=0 ms

  P2=24 ms

  P3=37 ms

- Rata-rata = (0 ms + 24 ms +27 ms)/3 = 17ms

- *Turnaround time:*

  P1 = 24 ms

  P2 =27 ms (dihitung dari awal kedatangan P2 hingga selesai dieksekusi)

  P3 = 30 ms.

- Rata-rata = (24 ms + 27 ms +30 ms)/3 = 27 ms.

SJF (Shortest Job First)

- Mendahulukan proses dengan Burst-Time terkecil.
- Ada 2 Tipe :

  Jika ada proses P1 yang datang pada saat P0 sedang berjalan → akan dilihat CPU burst P1 →

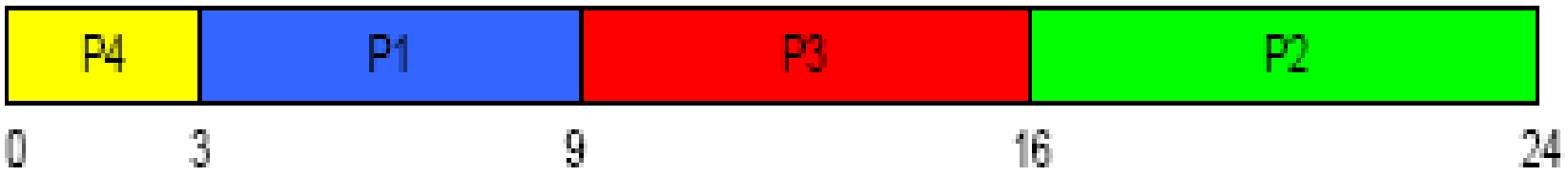
  - Preemptive, Jika CPU burst P1 lebih kecil dari sisa waktu yang dibutuhkan oleh P0 → CPU ganti dialokasikan untuk P1.

    Shortest-Remaining- Time-First scheduling

  - Non Preemptive, Akan tetap menyelesaikan P0 sampai habis CPU burstnya.

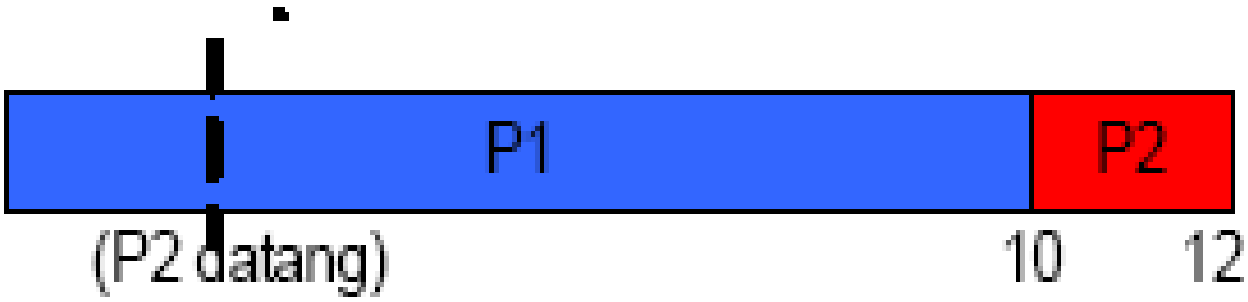
- Contoh SJF Scheduling → Non Preemptive
  - Waktu kedatangan sama

Proses	Waktu	Urutan	Kedatangan
P1	6	1	0
P2	8	2	0
P3	7	3	0
P4	3	4	0



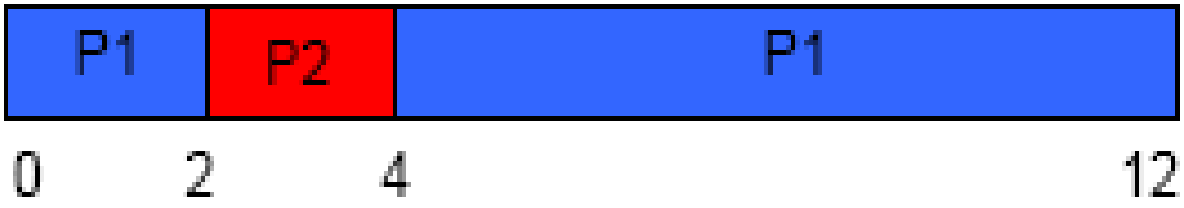
- Contoh SJF Scheduling → Non Preemptive
  - Waktu kedatangan tidak sama

Proses	Waktu	Urutan	Kedatangan
P1	10	1	0
P2	2	2	2



- Contoh SJF Scheduling → Preemptive
  - Waktu kedatangan tidak sama

Proses	Waktu	Urutan	Kedatangan
P1	10	1	0
P2	2	2	2



Soal

- Diketahui proses-proses sebagai berikut,

Process	*Arrival Time*	*Burst Time*
P1	0	40
P2	0	50
P3	0	10
P4	0	20

- Hitunglah waiting time rata-rata dan *turnaround time dari* ketiga proses secara non preemptive

Priority Scheduling

- Priority Scheduling merupakan algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi.
- Setiap proses memiliki prioritasnya masing-masing.
- Tiap proses diberi skala prioritas, proses yang mendapatkan prioritas tertinggi mendapat jatah waktu pemroses

Prioritas meliputi

- Waktu
- Memori yang dibutuhkan
- Banyaknya file yang dibuka
- Perbandingan antara rata-rata I/O Burst dengan rata-rata CPU Burst

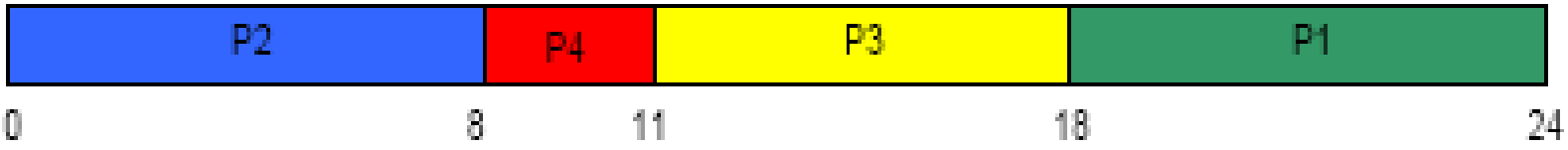
- Algoritma Priority Scheduling dapat bersifat *Preemptive* atau *Non Preemptive.*

  Jika ada proses P1 yang datang pada saat P0 sedang berjalan → akan dilihat prioritas P1, Jika prioritas P1>P0, maka :

  - Pada *Non Preemptive,* Algoritma tetap akan menyelesaikan P0 sampai habis CPU burstnya dan meletakkan P1 pada posisi head queue.
  - Pada *Preemptive,* P0 akan dihentikan dulu dan CPU ganti dialokasikan untuk P1.

Contoh Priority Scheduling

Proses	Waktu	Prioritas	Kedatangan
P1	6	4	0
P2	8	1	0
P3	7	3	0
P4	3	2	0



Soal

- Diketahui proses-proses sebagai berikut,

Process	*Arrival Time*	*Burst Time*	*Priority*
P1	0	10	1
P2	0	10	2
P3	0	5	1
P4	0	7	0
P5	0	8	0

- Hitunglah waiting time rata-rata dan *turnaround time dari* proses tersebut secara non preemptive

Soal

- Diketahui proses-proses sebagai berikut,

Process	*Arrival Time*	*Burst Time*	*Priority*
P1	0	10	1
P2	5	10	2
P3	10	5	1
P4	15	7	0
P5	15	8	0

- Hitunglah waiting time rata-rata dan *turnaround time dari* proses tersebut secara non preemptive

Round Robin

- Algoritma ini menggilir proses yang ada di antrian. Proses akan mendapat jatah sebesar time quantum. Jika time quantum-nya habis atau proses sudah selesai, CPU akan dialokasikan ke proses berikutnya.

- Konsep dasar algoritma ini menggunakan time sharing
- Pada dasarnya, prinsip hampir sama dengan FCFS, tapi bersifat *preemptive*

Contoh

Diketahui proses-proses sebagai berikut,

Proses	Durasi	Urutan	Kedatangan
P1	3	1	0
P2	4	2	0
P3	3	3	0

Hitunglah waiting time rata-rata dan *turnaround time dari* proses tersebut jika quantum time = 1

soal

- Diketahui proses-proses sebagai berikut,

Process	*Arrival Time*	*Burst Time*
P1	0	40
P2	0	50
P3	0	10
P4	0	20

- Hitunglah waiting time rata-rata dan *turnaround time dari* proses tersebut jika quantum time = 10

soal

- Diketahui proses-proses sebagai berikut,

Process	*Arrival Time*	*Burst Time*
P1	0	40
P2	20	50
P3	50	10
P4	70	20

- Hitunglah waiting time rata-rata dan *turnaround time dari* proses tersebut jika quantum time = 10

HIGHEST RATIO NEXT (HRN)

- Penjadwalan berprioritas dinamis.
- strategi penjadwalan dengan prioritas proses tidak hanya merupakan fungsi waktu layanan tetapi juga jumlah waktu tunggu proses. Begitu proses mendapat jatah pemroses, proses berjalan sampai selesai.

- Prioritas dinamis HRN dihitung berdasarkan rumus :

- Prioritas = (waktu tunggu + waktu layanan) / waktu layanan

contoh

Nama proses	AT	BT
A	0	4
B	1	2
C	2	5
D	3	8
E	4	4

Pada saat 0 : hanya ada A, A diolah

Pada saat 4 : A rampung, B, C, D, E telah tiba

perhitungan rasio penalti

Proses	WT	Rasio penalti
B	4 – 1 = 3	(3 + 2)/2 = 2,5
C	4 – 2 = 2	(2 + 5)/5 = 1,2
D	4 – 3 = 1	(1 + 8)/8 = 1,125
E	4 – 4 = 0	(0 + 4)/4 = 1

Rasio penalti tertinggi pada B, B diolah

Pada saat 6 : A dan B telah rampung

perhitungan rasio penalti

Proses	WT	Rasio penalti
C	2 + 2 = 4	(4 + 5)/5 = 1,8
D	1 + 2 = 3	(3 + 8)/8 = 1,375
E	0 + 2 = 0	(2 + 4)/4 = 1,5

Rasio penalti tertinggi pada C, C diolah

Pada saat 11 : A, B, dan C telah rampung

perhitungan rasio penalti

Proses	WT	Rasio penalti
D	3 + 5 = 8	(8 + 8)/8 = 2
E	2 + 5 = 7	(7 + 4)/4 = 2,75

Rasio penalti tertinggi pada E, E diolah

Pada saat 15 : A, B, C, dan E telah rampung, D diolah