

BAB II

PENJADWALAN PROSES

MATERI

1. Istilah-istilah dalam penjadwalan proses
2. Penjadwalan Satu Tingkat
3. Penjadwalan Multi Tingkat

STANDAR KOMPETENSI

Mengetahui teknik penjadwalan proses dan mengetahui istilah-istilahnya

CAPAIAN PEMBELAJARAN

Mahasiswa mengetahui teknik penjadwalan proses dan mengetahui istilah-istilahnya

1. Istilah-istilah dalam penjadwalan proses

o **Pekerjaan / Job**

User menyerahkan pekerjaan mereka pada komputer. Ada pekerjaan yang panjang, sedang dan pendek. Untuk dapat diolah oleh komputer harus dianalisis dahulu untuk mengetahui bagaimana sebaiknya pekerjaan itu dilakukan.

o **Terobosan / Troughput**

Pekerjaan dalam komputer dilaksanakan langkah demi langkah, sehingga dapat dikatakan bahwa troughput adalah banyaknya pekerjaan yang dapat diselesaikan oleh sistem komputer dalam satu satuan waktu.

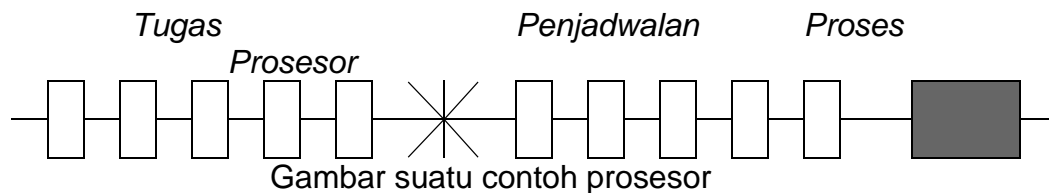
o Tugas / Task

Program yang terdiri dari banyak bagian program dan bagian program yang dikerjakan merupakan tugas bagi sistem operasi.

- *Tugas yang sedang dilaksanakan*, adalah tugas yang menemukan semua sumber daya yang diperlukannya, termasuk prosesor.
- *Tugas yang siap dilaksanakan*, adalah tugas yang menemukan semua sumber daya yang diperlukannya, kecuali prosesor.
- *Tugas yang belum dilaksanakan*, adalah tugas yang belum menemukan sumber daya yang diperlukannya.

o Proses / Process

Tugas yang telah dijadwalkan untuk menemukan prosesor / tugas yang telah diterima oleh penjadwalan.



Keterangan :

- **GK** = galur kendali / *control bus*
- **GD** = galur data / *data bus*, tempat informasi data berlalu lintas antar register
- **GA** = galur alamat / *address bus*, tempat informasi alamat memori berlalu lintas antar register karena setiap kali hanya ada 1 informasi dalam galur ini, maka lalu lintas diatur oleh tanda waktu dari kunci waktu. Tanda waktu dibentuk oleh basis waktu
- **A** = akumulator, register serba guna yang dapat menerima berbagai informasi
- **SLA** = satuan logika dan aritmatika / ALU, tempat pengolahan, contohnya untuk menghitung $3+4=?$, maka data 3 diletakkan di

register A dan data di register B. proses penjumlahan terjadi di SLA / ALU, hasil = 7 diletakkan kembali di register A

- **B** = register, register yang menerima informasi yang akan diolah bersama oleh isi register A
 - **PT** = pencacah program / *program counter*, program terletak dalam memori kerja pada alamat memori tertentu. PT menerima alamat memori awal saat diolah oleh prosesor dan juga menyaring dan menentukan alamat memori mana saja yang isinya akan dibawa ke prosesor dan hanya alamat yang sesuai dengan letak program yang sedang diolah yang diperbolehkan PT masuk ke prosesor. PT mencacah dirinya sebesar 1 cacahan dan dengan cara yang sama akan melayani alamat berikutnya sampai alamat terakhir.
 - **RI** = register instruksi, membawa dan menerima informasi dari memori kerja dan diperiksa.
 - **SK** = satuan kendali / *control unit*, menerima informasi instruksi program dan sesuai dengan isi instruksi SK mengendalikan kegiatan dalam prosesor.
 - **RA** = register alamat / *address register*, mencatat alamat memori yang isinya akan dicapai alamat itu di memori kerja. Jika pembacaan data dari memori kerja, data / isi dari memori kerja dibaca dan disalin ke RD, tetapi jika pada penulisan data ke memori kerja, maka isi RD disalin ke alamat memori kerja.
 - **RD** = register data.
 - RD dan RA bekerja sama dalam kegiatan pengambilan dan pengiriman data dari dan ke memori kerja.
- **2 siklus kerja prosesor :**
- ✓ *Siklus Jemput*
 - PT berisi alamat awal program pada memori kerja yang disesuaikan dengan isinya di memori kerja

- RA = PT, alamat memori PT diteruskan ke RA
- RA \Rightarrow RD, isi program diambil dari memori kerja dan diletakkan di RD
- RI = RD, dari RD program diteruskan ke RI
- PT = PT + 1, pencacah untuk melanjutkan siklus berikutnya
- Go to point b
- ✓ *Siklus Kerja*
 - RI = SK, isi program di RI akan diteruskan ke SK, RI berisi instruksi untuk melaksanakan sesuatu
 - SK \Rightarrow seluruh kegiatan di dalam prosesor dikendalikan sehingga kerja yang dimaksud oleh bagian program itu dirampungkan.
- **2 jenis kerja / proses :**
 - *Tugas / proses sistem*, berasal dari program sistem yaitu dari sistem bahasa, utilitas, operasi, dimana prosesor melayani sistem (prosesor dikuasai sistem / kontek sistem)
 - *Tugas / proses aplikasi*, berasal dari program aplikasi, dimana prosesor melayani aplikasi (prosesor dikuasai aplikasi / kontek aplikasi)
- **Proses Serentak / Concurrent Process :**

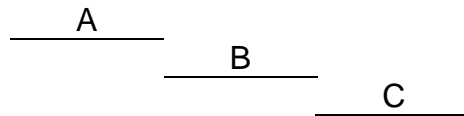
Yaitu prosesor menghadapi banyak tugas dan proses.

 - *Multiprogramming*, sistem komputer lebih dari 1 program sekaligus dalam pelaksanaan proses.
 - *Multitasking*, banyak bagian program yang dipersiapkan untuk diolah oleh prosesor namun belum sempat dijadwalkan untuk memperoleh prosesor.
 - *Multiprocessing*, sejumlah tugas yang telah dijadwalkan untuk menggunakan prosesor.

- *Multiplexing*, menggunakan pertukaran kendali dalam selang waktu terpisah-pisah.
- *Time sharing / rentang waktu*, secara bersamaan sejumlah pemakai dapat menggunakan 1 sistem komputer, sehingga setiap pemakai merasa bahwa seluruh sistem komputer seolah-olah untuk dirinya.

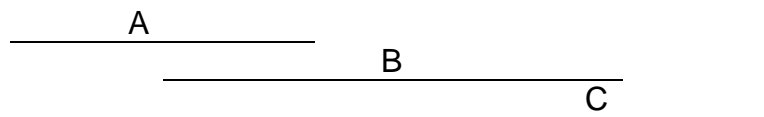
o **Proses Berurutan :**

Sejumlah proses berlangsung secara berselingan dan tidak ada diantara mereka yang bertumpang tindih waktu, sebelum 1 proses selesai, proses berikutnya belum bekerja.



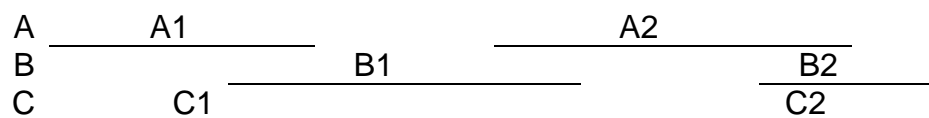
o **Proses Paralel :**

- pada proses tunggal, proses serentak bukan proses paralel karena proses tersebut di gali 1 demi 1, sepenggal demi sepenggal.
- Pada proses jamak, proses dapat dilaksanakan secara serempak diantara banyak prosesor sehingga disebut proses paralel



o **Proses Serentak Berpenggalan :**

Ada penggalan dari 1 proses yang berselingan dengan penggalan dari proses lain dan ada penggalan dari proses pertama yang bertumpang tindih waktu dengan penggalan proses kedua.



- **Scheduler** adalah bagian sistem operasi yang mengatur penjadwalan eksekusi proses-proses.
- **Algoritma penjadwalan (scheduling algorithm)** adalah algoritma yang digunakan.
- **Penjadwalan Proses**
 - *Antrian*, karena banyak proses yang muncul secara serentak maka dibuat antrian di depan prosesor, yang berada dalam keadaan siap dan hanya ada 1 proses yang berada dalam status kerja
 - *Prioritas*, mendahulukan pada antrian proses karena tidak semua proses sama pentingnya, sehingga dibuat suatu prioritas. Dalam prioritas, pekerjaan pada prosesor diselesaikan dahulu baru proses berprioritas akan di proses
 - *Preempsi*, sama dengan prioritas, tetapi pada preempsi jika ada proses yang mendapatkan preempsi maka preempsi akan menghentikan kerja prosesor dan mengeluarkan pekerjaan di dalam prosesor itu, sehingga proses berpreempsi dapat dilayani prosesor. Dan setelah proses berpreempsi selesai dilaksanakan, prosesor akan melaksanakan sisa proses yang dikeluarkan dari pekerjaannya tadi
- **Jangka penjadwalan**

Semua antrian dan penantian (contohnya yang dikeluarkan karena preempsi.

 - *Penjadwalan jangka pendek / short term scheduling / low level scheduling*, yaitu mengurus masuknya antrian siap ke prosesor serta antrian siap ke alat peripheral I/O, yang mengurus prioritas dan preempsi.
 - *Penjadwalan jangka media / medium term scheduling / intermediate level scheduling*, yaitu mengurus terhadap proses yang dikeluarkan

dari prosesor yang belum rampung dikerjakan dan melanjutkan pekerjaan proses tersebut di prosesor.

- *Penjadwalan jangka panjang / long term scheduling / high level scheduling*, yaitu mengurus masuknya pekerjaan baru berupa penentuan pekerjaan baru mana yang boleh diterima dan tugas disini diubah menjadi proses.
- **Tujuan penjadwalan** / kriteria baik tidaknya suatu algoritma penjadwalan
 - *Fairness* / pelayanan yang adil untuk semua pekerjaan
 - *Throughput* / memaksimalkan throughput
 - *Efficiency* / memaksimalkan pemakaian prosesor
 - *Overhead* / meminimumkan waktu tunggu
 - Pemakaian sumber daya seimbang
 - Tidak terjadi penundaan waktu tak hingga
 - Kegiatan sumber daya dapat dideteksi terlebih dahulu
- **Perhitungan kerja prosesor**
 - **t** adalah lama proses pada prosesor, lama waktu sesungguhnya yang diperlukan untuk mengolah proses dalam prosesor.
 - **T** adalah lama tanggap pada prosesor, lama waktu yang diperlukan oleh prosesor sejak tiba sampai dengan rampung diolah oleh prosesor, terdapat waktu tunggu dalam antrian / dalam preempsi.
 - Lama tanggap *turn around time*, yaitu memperhitungkan lama waktu yang diperlukan oleh proses untuk keluaran
 - Lama tanggap *respon time*, yaitu tidak memperhitungkan lama waktu yang diperlukan oleh proses untuk keluaran
 - **Tr** adalah lama tanggap rata-rata, yaitu perbandingan lama tanggap setiap proses (T_i) dengan jumlah proses serentak yaitu $Tr = \sum T_i / N$

- **S** adalah waktu sia-sia, waktu yang terbuang di dalam antrian / selama terkena preempsi yaitu selisih antara lama tanggap dengan lama proses ($T-t$)
- **Rt** adalah rasio tanggap, perbandingan antara lama proses terhadap lama tanggap, $Rt = t / T$
- **Rp** adalah rasio penalti, perbandingan antara lama tanggap terhadap lama proses, $Rp = T/t$ dan karena $t \leq T$, maka $Rt < 1$ dan $Rp > 1$

2. Penjadwalan Satu Tingkat

Penjadwalan satu tingkat terdiri dari :

- Pertama Tiba Pertama Dilayani (PTPD)
 - Proses Terpendek Dipertamakan (PTD)
 - Proses Terpendek Dipertamakan Preempsi (PTDP)
 - Rasio Penalti Tertinggi Dipertamakan (RPTD)
 - Putar Gelang (PG)
 - Putar Gelang Prioritas Berubah (PGPB)
- ✓ *Pertama Tiba Pertama Dilayani (PTPD) / First Come First Served (FCFS)*

Penjadwalan ini murni antrian, tanpa prioritas tanpa preempsi.

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	9	0	9	9
B	0	30	9	39	39
C	0	4	39	43	43
D	0	8	43	51	51
E	0	12	51	63	63
↓ serentak				ΣTi	205
				Tr	41

Tr cukup besar, jika dibandingkan dengan lama tanggap dari masing-masing proses.

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	4	0	4	4
B	1	7	4	11	10
C	3	3	11	14	11
D	7	8	14	22	15
				ΣTi	40
				Tr	10

Lama tanggap \neq saat rampung karena perbedaan saat tiba.

✓ *Proses Terpendek Dipertamakan (PTD) / Shortest Job First (SJF)*

Penjadwalan ini adalah antrian dengan prioritas tanpa preempsi, yang menjadi prioritas adalah proses yang terpendek (tersingkat), makin pendek proses, makin tinggi prioritasnya.

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	11			
B	0	8			
C	0	10			
D	0	3			
E	0	5			

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
D	0	3	0	3	3
E	0	5	3	8	8
B	0	8	8	16	16
C	0	10	16	26	26
A	0	11	26	37	37
				ΣTi	90
				Tr	18

↓ serentak
↓ prioritas

Tr lebih singkat, jika dibandingkan dengan PTPD

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	5			
B	3	7			
C	5	2			
D	6	4			

Disusun urutan proses berdasarkan prioritasnya, yaitu CDAB, tetapi pada saat tiba = 0, proses terpendek C belum datang, maka mengerjakan proses A dahulu, setelah selesai pada saat 5, proses C sudah datang, sehingga C dikerjakan dan seterusnya seperti :

	A		B		C	D													
	↓		↓		↓	↓													
Saat	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Proses	A	A	A	A	A	C	C	D	D	D	D	D	B	B	B	B	B	B	B

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	5	0	5	5
C	5	2	5	7	2
D	6	4	7	11	5
B	3	7	11	18	15
$\Sigma=18$				ΣTi	27
				Tr	6.75

Keuntungan : memperkecil rata-rata lama tanggap

Kelemahan : layanan terhadap proses panjang bisa tidak terlayani jika proses pendek datang terus.

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	8			
B	4	5			
C	6	3			
D	9	8			
E	14	4			

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	8	0	8	8
C	6	3	8	11	5
B	4	5	11	16	12
E	14	4	16	20	6
D	9	8	20	28	19
				ΣTi	50
				Tr	10

✓ Proses Terpendek Dipertamakan Preempsi (PTDP)

Preemptive Shortest Job First (PSJF)

Penjadwalan ini dengan prioritas dengan preempsi, yang menjadi prioritas adalah sisa proses. Proses yang terpendek bisa didahulukan dengan cara membandingkan sisa waktu proses yang sedang dilaksanakan dengan proses yang tiba, dengan preempsi mengeluarkan proses yang sedang diolah untuk melaksanakan proses yang lebih pendek / singkat.

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	7			
B	2	3			
C	4	9			
D	5	4			

	A	B	C	D																			
	↓	↓	↓	↓																			
Saat	0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2
Proses	A	A	B	B	B	D	D	D	D	A	A	A	A	A	C	C	C	C	C	C	C	C	C

- Pada saat 0, proses A dikerjakan karena A datang pertama
- Pada saat 2, B datang, B=3, sisa A=5, B<A maka A dikeluarkan dari proses, B dikerjakan
- Pada saat 4, C datang, C=9, sisa A=5, sisa B=1, B<A<C, maka B dikerjakan
- Pada saat 5, D datang, D=4, sisa A=5, sisa C=9, D<A<C, maka D dikerjakan sampai selesai karena tidak ada proses yang datang lagi
- A<C maka A dikerjakan
- C dikerjakan
- Selesai

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	7	0	14	14
B	2	3	2	5	3
C	4	9	15	23	19
D	5	4	5	9	4
$\Sigma=23$				ΣTi	40
				Tr	10

Karena preempsi, PTDP lebih baik dari PTD dalam hal kecilnya Tr, tetapi untuk proses panjang lebih lama dibandingkan PTD.

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	8			
B	2	5			
C	4	7			
D	5	1			

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	8	0	14	14
B	2	5	2	8	6
C	4	7	14	21	17
D	5	1	5	6	1
				ΣTi	38
				Tr	9.5

- ✓ *Rasio Penalti Tertinggi Dipertamakan (PTD) / Highest Penalti Ratio Next (HPRN)*

Penjadwalan dengan prioritas tanpa preempsi. Yang menjadi prioritas adalah besarnya rasio pinalti. Tetap mendahulukan proses pendek ditambah dengan mempertimbangkan rasio penaltinya, yang ditentukan berdasarkan lama waktu antriannya. Sehingga prioritas proses panjang akan turut meningkat melalui peningkatan rasio pinalti, sehingga pada suatu saat proses panjang pada antrian yang telah lama menunggu akan menyusul proses pendek. $S = (T-t) ; Rp = T/t = (s+t)/t$

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	4			
B	1	2			
C	2	5			
D	3	8			
E	4	4			

Nama Proses	Waktu sia-sia	Rasio Penalti	
B	$4 - 1 = 3$	$(3 + 2) / 2 = 2,5$	Rp>>
C	$4 - 2 = 2$	$(2 + 5) / 5 = 1,4$	
D	$4 - 3 = 1$	$(1 + 8) / 8 = 1,125$	
E	$4 - 4 = 0$	$(0 + 4) / 4 = 1$	

Nama Proses	Waktu sia-sia	Rasio Penalti	
C	$6 - 2 = 4$	$(4 + 5) / 5 = 1,8$	Rp>>
D	$6 - 3 = 3$	$(3 + 8) / 8 = 1,375$	
E	$6 - 4 = 2$	$(2 + 4) / 4 = 1,5$	

Nama Proses	Waktu sia-sia	Rasio Penalti	
D	$11 - 3 = 8$	$(8 + 8) / 8 = 2$	
E	$11 - 4 = 7$	$(7 + 4) / 4 = 2,75$	Rp>>

	A	B	C	D	E																				
Saat	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
Proses	A	A	A	A	B	B	C	C	C	C	C	E	E	E	E	D	D	D	D	D	D	D	D	D	D

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	4	0	4	4
B	1	2	4	6	5
C	2	5	6	11	9
D	3	8	15	23	20
E	4	4	11	15	11
$\Sigma = 23$				ΣT_i	49
				Tr	9.8

✓ *Putar Gelang (PG) / Round Robin (RR)*

Penjadwalan ini tanpa prioritas, dengan preempsi. Secara bergiliran berdasarkan antrian (tanpa prioritas) prosesor melayani sejenak setiap

proses tergantung besarnya quantum waktu. Secara berturut-turut proses yang dilayani prosesor dan belum rampung akan kembali ke akhir antrian yang ada, sehingga pergiliran ini berputar seperti gelang. Dan hanya proses yang telah rampung terlayani yang meninggalkan prosesor dan antrian tersebut. Jadi setiap proses dilayani selama quantum waktu tertentu secara bergiliran.

Quantum waktu : waktu sejenak yang digunakan oleh prosesor untuk melayani setiap proses. Perubahan quantum waktu membedakan hasil layanan terhadap antrian yang sama.

Contoh dengan $Q = 3$

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	7			
B	0	5			
C	0	8			
D	0	2			
E	0	6			

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	7	0	26	26
C	0	5	3	19	19
B	0	8	6	28	28
E	0	2	9	11	11
D	0	6	11	25	25
$\Sigma = 28$				ΣT_i	109
				Tr	21.8

Dengan $Q = 2$

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	5			
B	1	3			
C	5	7			
D	6	1			
E	7	6			

		A	B				C	D	E																															
		↓	↓				↓	↓	↓																															
S		0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	1	2	2	2															
t																																								
Pro		A	A	B	B	A	A	B	C	C	D	E	E	A	C	C	E	E	C	C	E	E	C	E	E	C														
s																																								

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap
A	0	5	0	13	13
B	1	3	2	7	6
C	5	7	7	22	17
D	6	1	9	10	4
D	7	6	10	21	14
				ΣTi	54
				Tr	10,4

Pada saat 0, A dikerjakan 2x, saat 2, B sudah datang, dikerjakan 2x, saat 4, belum ada yang datang maka kerjakan A dan B lagi. Kemudian kerjakan C, D, E yang sudah datang, tetapi di antrian masih ada A, A dikerjakan kemudian E, sisa C, sisa E dan sisa C sampai selesai.

Secara merata, penjadwalan putar gelang memperlambat proses yang tiba dan secara merata pula semua proses dilayani oleh prosesor, sehingga biasanya digunakan pada proses interaktif jika semua proses menuntut dikerjakan segera.

3. Penjadwalan Multi Tingkat

✓ Metode Simulasi

Metode ini berdasarkan sejumlah variabel yang disimulasikan sistem komputer yang digunakan, memerlukan data masukan melalui bilangan acak.

Keuntungan : jika ditangani dengan baik maka metode ini cukup baik.

Kelemahan : mahal, karena banyak menggunakan jam prosesor dan ruang memori.

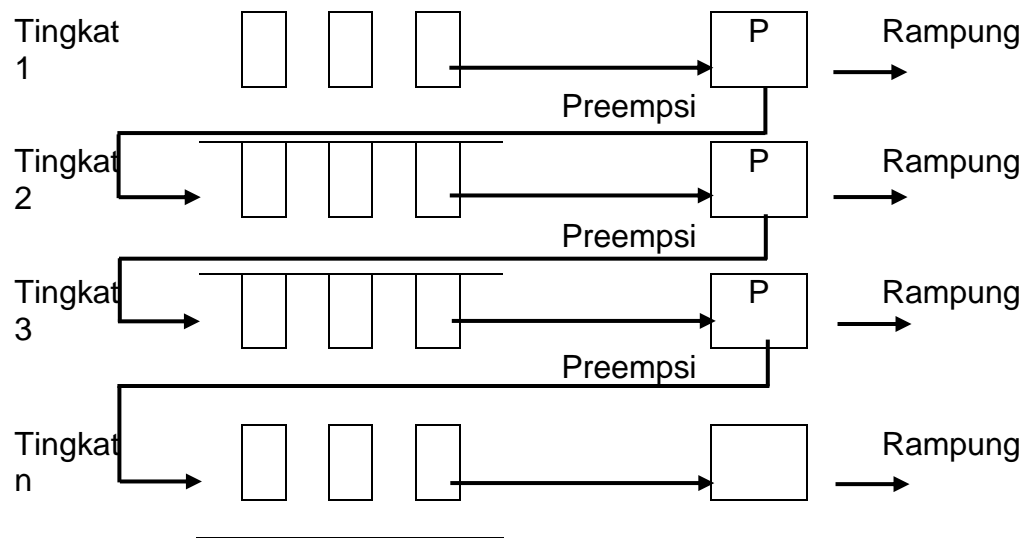
✓ Metode Implementasi

Metode ini bekerja dengan cara mengamati hasil dari implementasi setiap penjadwalan / menerapkan berbagai penjadwalan pada pekerjaan yang sesungguhnya.

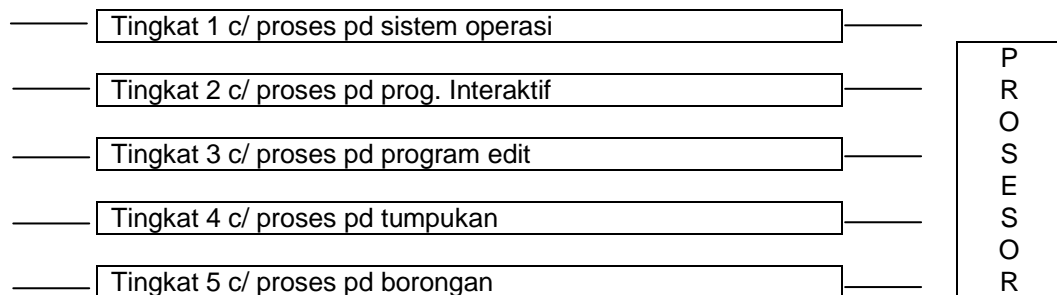
Keuntungan : cukup cermat.

Kelemahan : perubahan macam penjadwalan mengganggu pemakai komputer dan perubahan jenis pekerjaan mengganggu penjadwalan.

✓ Antrian Multi Tingkat Berbalikan / Feedback multi level queue



✓ Antrian Multi Tingkat (Multi level queue)



Mengumpulkan proses-proses yang berkepentingan sama dalam 1 tingkat. Proses dikerjakan tingkat demi tingkat dari tingkat tertinggi

sampai dengan tingkat terendah dimana setiap tingkat telah rampung dikerjakan

✓ evaluasi analitik

- pemodelan deterministik

pekerjaan sudah ditetapkan terlebih dahulu, menerapkan berbagai penjadwalan dan dievaluasi hasilnya.

Nama proses	Saat tiba	Lama proses
A	0	12
B	0	30
C	0	2
D	0	8
E	0	10

Dari tabel tersebut dapat kita cari :

Macam penjadwalan	Rerata lama tanggap	Rerata waktu sia-sia
PTPD	42,4	30
PTD	29,2	14
PG (Q=8)	38,4	28

Keuntungan : ketepatan untuk menunjukkan rata-rata lama tanggap / rata-rata waktu sia-sia

Kelemahan : hasil evaluasi hanya berlaku untuk proses yang telah ditetapkan

- analisis model antrian

menganggap sistem komputer sebagai suatu jaringan alat layan, menggunakan rumus distribusi probabilitas untuk memperkirakan bentuk antrian dan bentuk pelaksanaan proses. Sasaran rumus distribusi probabilitas adalah untuk lama proses, saat tiba dan kecepatan layan. Sehingga dapat menentukan analisis jaringan antrian, dengan ditetapkan rumus litte, yaitu : $n = u \cdot s$, dengan $n =$

rerata panjang antrian, s = rerata waktu sia-sia / waktu tunggu, u
= rerata kecepatan tiba proses baru.

Kelemahan : Pengambilan model cukup rumit untuk sistem tertentu
dan model tidak selalu cocok dengan keadaan yang sebenarnya

CONTOH SOAL

1. Sebutkan dan jelaskan tipe-tipe penjadwalan.

- **Penjadwalan jangka pendek (short-termscheduller)**

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

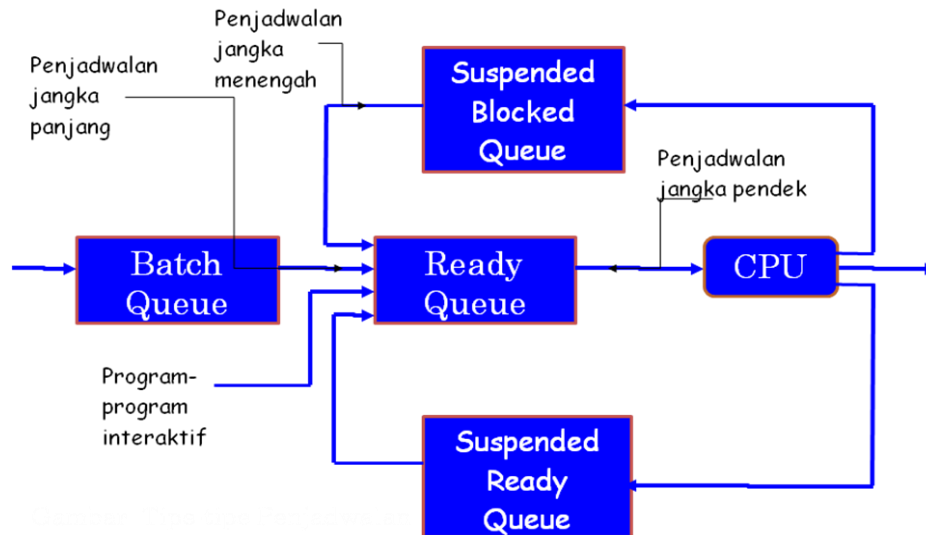
- **Penjadwalan jangka menengah (medium termscheduller)**

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

- **Penjadwalan jangka panjang (long-termscheduller)**

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

Gambar tipe-tipe penjadwalan:



2. Sebutkan dan jelaskan tentang strategi penjadwalan

- **Penjadwalan *Nonpreemptive***

Begitu proses diberi jatah waktu pemroses maka pemroses tidak dapat diambil alih oleh proses lain sampai proses itu selesai

- **Penjadwalan *Preemptive***

Saat proses diberi jatah waktu pemroses maka pemroses dapat diambil alih oleh proses lain sehingga proses disela sebelum selesai dan harus dilanjutkan menunggu jatah waktu pemroses tiba kembali pada proses itu

3. Sebutkan algoritma-algoritma yang menerapkan strategi preemptive dan nonpreemptive.

- Preemptive
 - o **RR** (*Round-Robin*).
 - o **SRF** (*Shortest-Remaining-First*).
 - o **PS** (*Priority Scheduling*).
 - o **GS** (*Guaranteed Scheduling*).

- Nonpreemptive
 - o **FIFO** (*First-in, First-out*).
 - o **SJF** (*Shortest Job First*).
 - o **HRN** (*Highest Ratio Next*).
 - o **MFQ** (*Multiple Feedback Queues*).

EVALUASI

1. Sebutkan istilah apa saja yang ada dalam penjadwalan proses, berikan contohnya
2. Jelaskan tentang parameter yang digunakan untuk menghitung kerja prosesor
3. Sebutkan macam-macam penjadwalan satu tingkat, berikan contohnya
4. Jelaskan apa yang dimaksud dengan penjadwalan multi tingkat
5. Jelaskan apa yang dimaksud dengan evaluasi analitik
6. Sebutkan keuntungan dan kerugian dari masing-masing metode simulasi maupun metode implementasi

DAFTAR PUSTAKA :

- Harvey M Deitel dan Paul J Deitel. 2005. *Java How To Program*. Sixth Edition. Prentice Hall.
- Bambang Hariyanto. 1997. *Sistem Operasi* . Buku Teks Ilmu Komputer . Edisi Kedua. Informatika. Bandung.
- John L Hennessy dan David A Patterson. 2002. *Computer Architecture* . A Quantitative Approach . Third Edition. Morgan Kaufman. San Francisco.
- Randall Hyde. 2003. *The Art of Assembly Language*. First Edition. No Strach Press.

- Kenneth H Rosen. 1999. *Discrete Mathematics and Its Application*. McGraw Hill.
- Ronald L Krutz dan Russell D Vines. 2001. *The CISSP Prep Guide Mastering the Ten Domains of Computer Security*. John Wiley & Sons.
- Sri Kusumadewi. 2000. *Sistem Operasi* . Edisi Dua. Graha Ilmu. Yogyakarta.
- Robert Love. 2005. *Linux Kernel Development* . Second Edition. Novell Press.
- Larry L Peterson dan Bruce S Davie. 2000. *Computer Networks A Systems Approach*. Second Edition. Morgan Kaufmann.
- Riri Fitri Sari dan Yansen. 2005. *Sistem Operasi Modern* . Edisi Pertama. Andi. Yogyakarta.
- Betha Sidik. 2004. *Unix dan Linux*. Informatika. Bandung.
- Abraham Silberschatz, Peter Galvin, dan Greg Gagne. 2002. *Applied Operating Systems*. Sixth Edition. John Wiley & Sons.
- Avi Silberschatz, Peter Galvin, dan Grag Gagne. 2005. *Operating Systems Concepts*. Seventh Edition. John Wiley & Sons.
- William Stallings. 2001. *Operating Systems: Internal and Design Principles*. Fourth Edition. Edisi Keempat. Prentice-Hall International. New Jersey.
- Andrew S Tanenbaum dan Albert S Woodhull. 1997. *Operating Systems Design and Implementation*. Second Edition. Prentice-Hall.
- Andrew S Tanenbaum. 2001. *Modern Operating Systems*. Second Edition. Prentice-Hall.