

**Jawaban UAS Sistem Operasi TA 2021/2022**  
**By Diklat From HMIF Universitas Diponegoro**

1. Sebutkan 3 esensi kebutuhan *storage* (terkait untuk penyimpanan dan mengambil suatu informasi).

**Jawab:**

- 1) Mampu menyimpan data besar.
- 2) Mampu menjaga data walaupun yang mengaksesnya dihentikan.
- 3) Proses yang bersamaan dapat mengakses data secara bersamaan tanpa ada yang terganggu.

2. Berikan suatu contoh pengalaman saudara mengenai:

- a. Implementasi *file System* (Keamanan file, akses file, dsb).
- b. Implementasi piranti *input/output* (software/hardware).

**Jawab :**

- a) Implementasi *file system*

Saat menggunakan CLI pada linux jika akan mengakses file untuk *user* kedua oleh *user* pertama maka bergantung pada kategori *user* pertama termasuk yang diijinkan atau tidak. Admin dapat mengakses dan *user* pertama dapat mengakses jika diberi ijin oleh *user* kedua. Akses file tersebut diantaranya adalah *read*, *write*, *execute*.

- b) Implementasi piranti *input/output*

Melakukan restart device saat program lag atau menyuruh restart ulang, menginstal *driver* secara manual jika belum terdapat dalam *device*, menggunakan *mouse* dalam pengoperasian *device*, menggunakan keyboard untuk menginput, menggunakan *speaker* dan *headset* untuk output suara, menggunakan *printer* dan *scanner* untuk menghasilkan salinan.

3. Piranti I/O dikategorikan menjadi 2 yaitu *block device* dan *character device*. Jelaskan dan berikan contohnya.

**Jawab :**

- 1) *Block Device*

Adalah perangkat yang menyimpan informasi dalam sebuah blok yang ukurannya tertentu dan memiliki alamat masing – masing.

Contoh : *hard disk*, *blue – ray disc*, *USB sticks*

- 2) *Character Device*

Adalah perangkat yang mengirim atau menerima sebarisan karakter tanpa menghiraukan struktur blok, tidak memiliki alamat dan operasi pencarian.

Contoh : *printers*, *network interfaces*, *mice*, *rais*

4. Pikirkan suatu disk dengan 40 silinder, saat ini membaca blok silinder 11, kemudian disk request datang secara berturutan untuk membaca cylinder 1, 35, 21, 18, 34, 8, 4, 19, 23 dan 15. Misalkan *seek time* per *cylinder* adalah 5 msec, gambarkan skenario dan berapa seek time yang diperlukan, jika *disk arm scheduling algorithm* yang digunakan adalah *First Come First Served*, *Shortest Seek First*, dan Elevator Algorithm (periksa bila elevator bergerak naik). Menurut saudara apa kelebihan dan kelemahan ketiga metode tersebut.

**Jawab :**

Kelebihan dan kekurangan metode:

a. FCFS

Kelebihan: Setiap permintaan mendapat kesempatan yang adil , tidak ada penundaan yang tidak terbatas.

Kekurangan : Tidak mencoba mengoptimalkan pencarian, mungkin tidak emberikan layanan terbaik.

b. SSF

Kelebihan: Waktu respon rata-rata berkurang, *through out* meningkat.

Kekurangan : *Overhead* untuk menghitung waktu pencarian di mulai dapat menyebabkan kelaparan untuk permintaan jika memiliki waktu pencarian lebih tinggi dibandingkan dengan permintaan yang masuk, varians waktu respons yang tinggi karena hanya mendukung beberapa permintaan.

c. Elevator Algorithm

Kelebihan: *throughput* tinggi, varians waktu respon yang rendah, waktu respon rata-rata.

Kekurangan : waktu tunggu yang lama untuk permintaan lokasi yang baru saja dikunjungi oleh lengan *disk*.

5. Pada suatu saat sebuah system memiliki status sbb:

Proses	Resource					
	R1	R2	R3	R4	R5	R6
P1			<i>Hold</i>		<i>Want</i>	
P2			<i>Want</i>	<i>Hold</i>	<i>Want</i>	
P3		<i>Hold</i>	<i>Want</i>			
P4		<i>Want</i>		<i>Hold</i>		
P5			<i>Want</i>		<i>Hold</i>	
P6		<i>Want</i>		<i>Want</i>		<i>Hold</i>
P7	<i>Hold</i>	<i>Want</i>				

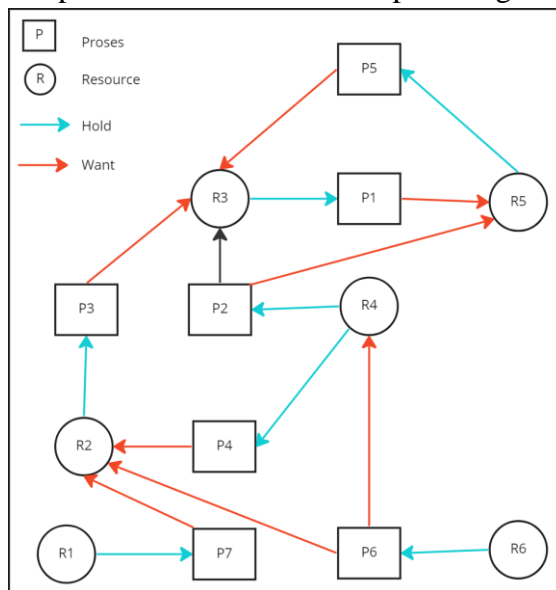
*Hold* = *Use* = menggunakan *resource*.

*Want* = *request* = meminta *request*.

Gunakan *deadlock modeling* untuk menganalisis apakah pada sistem tersebut terjadi *deadlock* atau tidak. Jika ya, tuntutkan proses yang mengalami *deadlock*.

**Jawab :**

Untuk menentukan terdapat *deadlock* atau tidak dapat kita gunakan graph.



Dari graph diatas dapat disimpulkan bahwa terjadi deadlock dikarenakan terjadinya **Circular Wait**. Deadlock tersebut terjadi pada proses 1 (P1) dan Proses 5 (P5). Proses 1 sedang menggunakan resource 3 dan menginginkan resource R5 dimana resource tersebut digunakan oleh proses 5(P5) yang merequest resource R3 yang sedang digunakan oleh proses R3.

6. Dengan menggunakan Banker Algorithm, tentukan apakah state berikut ini safe atau *unsafe*. Jika safe, tunjukkan bahwa ada cara agar semua proses dapat berjalan hingga akhirnya terminate.

Proses	Has	Max
A	1	9
B	1	3
C	2	5
D	4	9
Free = 2		

**Jawab :**

*Free (Source available to use)* saat ini : 2

Proses	Has	Max	Need	Source diberikan	Free (saat proses berjalan)	Free (Ketika proses selesai)
A	1	9	8	(4) 8	1	10
B	1	3	2	(1) 2	0	3
C	2	5	3	(2) 3	0	5
D	4	9	5	(3) 5	0	9

Keterangan :

(1) Pertama-tama, dilakukan pengecekan untuk proses yang membutuhkan source yang dapat dipenuhi oleh source yang available(Free) dimana diawal proses terdapat 2 source saja yang available untuk digunakan. Proses yang membutuhkan source yang memenuhi adalah proses B yang mana proses tersebut membutuhkan sebanyak 2 source lagi untuk melakukan proses sehingga proses B yang diberikan source terlebih dahulu. Karena proses B membutuhkan sebanyak 2 lagi source maka semua source yang available saat ini bersisa 0. Ketika proses telah selesai dijalankan maka proses B mengembalikan sebanyak 3 proses yang digunakan sebelumnya. Sehingga proses B berhasil dijalankan dan terdapat 3 source yang available.

(2) Selanjutnya, dicari lagi proses yang membutuhkan source yang dapat dipenuhi oleh source yang available saat ini yaitu proses C yang membutuhkan 3 source. Setelah proses C selesai maka akan mengembalikan sebanyak 5 source yang digunakan sebelumnya. Sehingga proses C berhasil dijalankan dan terdapat 5 source yang available.

(3) Kemudian, dicari lagi proses yang membutuhkan source yang dapat dipenuhi oleh source yang available saat ini yaitu proses D yang membutuhkan 5 source. Setelah proses D selesai maka akan mengembalikan sebanyak 9 source yang digunakan sebelumnya. Sehingga proses D berhasil dijalankan dan terdapat 9 source yang available.

(4) Kemudian karena proses A yang terakhir maka kita lakukan pengecekan apakah proses A membutuhkan source yang dapat dipenuhi oleh source yang available saat ini yaitu sebanyak 9 source. Karena proses A membutuhkan sebanyak 8 source lagi untuk prosesnya maka dapat diberikan sebanyak 8 source dari source yang available (Free) sehingga source yang available saat ini tersisa 1. Setelah proses A selesai dijalankan maka akan mengembalikan sebanyak 9 source. Sehingga proses A berhasil dijalankan dan terdapat 9 ditambah 1 source yang free yaitu 10 source yang available.

**Kesimpulan :**

Dengan menggunakan banker algorithm, state tersebut safe dengan cara proses dijalankan dengan urutan B-C-D-A.

7. Dengan menggunakan bankers algorithm, tentukan apakah state berikut ini safe atau unsafe. Jika safe, tunjukkan bahwa ada cara agar semua proses dapat berjalan hingga akhirnya terminate. Catatan: Memiliki 5 tape driver, 3 plotter, 3 printer, dan 22 CD ROM drive.

Existing resource(E) = (5 3 3 2)

Processed resource(P) = (2 1 3 1)

Available resource(A) = (3 2 0 1)

Resource assigned(C) =  $\begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$  Resource still needed(R) =  $\begin{bmatrix} 3 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 \end{bmatrix}$

**Jawab :**

misal tape drive(A), Plotter(B), Printer (C), dan CD ROM drive(D).

Proses	Resource Assigned(C)				Resource still needed (R)				Max = C + R			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	0	0	1	0	3	2	1	0	3	2	2	0
P2	2	0	1	1	0	1	1	1	2	1	2	2
P3	0	1	1	0	2	1	0	1	2	2	1	1

existing resource (A)				Available Resource (A)			
A	B	C	D	A	B	C	D
5	3	3	2	3	2	0	1

Untuk menjalankan prosesnya, kita harus cek terlebih dahulu apakah banyaknya kebutuhan resource pada suatu proses dapat dipenuhi oleh available resource.

cek : resource still needed <= available resource

1.	Proses	Resource Still need (R)					Available Resource (A)				
		A	B	C	D		A	B	C	D	
	P1	3	2	1	0	>=	3	2	0	1	tidak memenuhi
	P2	0	1	1	1	>=	3	2	0	1	tidak memenuhi
	P3	2	1	0	1	<=	3	2	0	1	memenuhi

Proses P3 memenuhi sehingga kita berikan resource yang availabel kepada P3

Proses	Resource Assigned(C)				Resource still needed (R)				Residual Resource			
	A	B	C	D	A	B	C	D	A	B	C	D
P3	0	1	1	0	2	1	0	1	1	1	0	1

Available Resource (A)

Now Max + Residual Res

A	B	C	D
3	3	1	2

2.

Proses	Resource Still needed(R)					Available Resource (A)				
	A	B	C	D		A	B	C	D	
P1	3	2	1	0	<=	3	3	1	2	memenuhi
P2	0	1	1	1	<=	3	3	1	2	memenuhi

Proses P1 dan P2 memenuhi sehingga kita berikan resource yang availabel nya terserah, misalnya kita berikan kepada P1 terlebih dahulu

Proses	Resource Assigned(C)				Resource still needed (R)				Residual Resource			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	0	0	1	0	3	2	1	0	0	1	0	2

Available Resource (A)

Now Max + Residual Res

A	B	C	D
3	3	2	2

Proses	Resource Assigned (C)				Resource still needed (R)				Residual Resource			
	A	B	C	D	A	B	C	D	A	B	C	D
P2	2	0	1	1	0	1	1	1	3	2	1	1

Available Resource (A)

Now Max + Residual Res

A	B	C	D
5	3	3	3

Proses berhasil dilakukan dan semua proses mendapatkan resource sesuai kebutuhan, sehingga dapat disimpulkan State tersebut Safe dengan cara proses dilakukan berdasarkan urutan : P3 - P1 - P2

8. Ada empat kondisi yang menyebabkan deadlock dapat terjadi, sebutkan dan bagaimana deadlock prevention dapat dilakukan.

Jawab :

➤ 4 Kondisi yang menyebabkan deadlock:

1) Mutual Exclusion

Adalah kondisi dimana setiap kondisi sumber daya diberikan tepat pada satu proses pada suatu waktu.

2) Hold and wait

Adalah situasi dimana proses sedang hold suatu source secara eksklusif dan ia menunggu mendapatkan resource resource lain (wait).

3) No Pre-emption

Adalah kondisi dimana sumber daya yang sedang digunakan oleh suatu proses tidak bisa sembarangan diambil dari proses tersebut atau harus menunggu sampai sumber daya tersedia.

4) Circular wait

Adalah kondisi dimana proses pertama membutuhkan sumber daya yang dimiliki oleh proses kedua, kemudian proses kedua membutuhkan sumber daya yang dimiliki proses ketiga dan begitu seterusnya sehingga membentuk rantai sirkular yang masing-masing menunggu resource dari proses berikutnya.

➤ Bagaimana deadlock prevention terjadi yaitu dengan meyakinkan bahwa paling sedikit satu dari kondisi deadlock tidak terjadi yaitu dengan:

a) Meniadakan mutual exclusion

Dengan melakukan spooling perangkat-perangkat yang didedikasikan ke suatu proses. Dengan spooling, permintaan-permintaan diantrikan di hardisk. Setiap job diantrian spooler akan dialayani satu persatu.

b) Meniadakan hold and wait

Dengan mengalokasikan semua sumber daya atau tidak sama sekali dan hold and release.

c) Meniadakan non pre-emption

d) Meniadakan circular wait

Yaitu dengan proses hanya diperbolehkan menggenggam satu sumber daya. Penomoran global semua sumber daya.

**Pembahasan dan jawaban ini hanyalah sebuah referensi**

**Jawaban belum tentu benar**

**Selamat UAS dan wish you all the best**