

NAMA : EDWARD                      Mata Kuliah                      : Operating System  
NIM : 2201741971                      Kode Mata Kuliah                      : COMP6153  
KELAS : LB-08                      Fakultas / Departemen                      : School of Computer Science

## I. ESSAY (100%)

Jawablah pertanyaan-pertanyaan dibawah ini dengan ringkas dan jelas. **Khusus untuk Penjelasan**, jawaban Anda harus **minimal 150 kata**. Jawaban diperbolehkan dalam bentuk **point**.

### 1. [15%] I/O Management

Permasalahan utama pada I/O Management adalah **efficiency** dan **generality**. Jelaskan yang dimaksud dengan **efficiency** dan **generality** dalam hal ini, dan bagaimana strategi penanganannya pada sistem operasi modern di saat ini.

**Jawaban (174 Kata):**

#### Efficiency

➔ Berfokus pada penggunaan CPU. Penting karena operasi I / O sering kali mengalami hambatan seperti sangat lambat dibandingkan dengan memori utama dan prosesor.

#### Generality

➔ Sesuai untuk pola akses rekaman tertentu seperti tape drive hanya dapat mengakses record yang ditempatkan tepat sebelum atau sesudah posisi Read/Write Saat ini.

➔ Menyediakan User Interface sederhana untuk proses pengguna: menentukan ID unik untuk perangkat ini dan perintah kontrol sederhana seperti Open, Close, Read, Write.

#### Strategi Penanganan

##### a) Variable Partitioning

Partisi pertama dicadangkan untuk sistem operasi. Ruang yang tersisa dibagi menjadi beberapa bagian. Ukuran setiap partisi akan sama dengan ukuran prosesnya. Ukuran partisi bervariasi sesuai dengan kebutuhan proses sehingga fragmentasi internal dapat dihindari.

##### b) Multilevel Paging

Apa pun level pagingnya, semua tabel halaman akan disimpan di memori utama. Setiap entri tabel halaman kecuali entri tabel halaman tingkat terakhir berisi alamat dasar dari tabel halaman tingkat berikutnya.

##### c) Segmentation

Suatu proses dibagi menjadi Segmen. Bagian-bagian yang terbagi dalam sebuah program yang tidak selalu berukuran sama disebut segmen. Segmentasi memberikan pandangan pengguna tentang proses yang tidak diberikan paging.

NAMA : EDWARD      Mata Kuliah : Operating System  
NIM : 2201741971      Kode Mata Kuliah : COMP6153  
KELAS : LB-08      Fakultas / Departemen : School of Computer Science

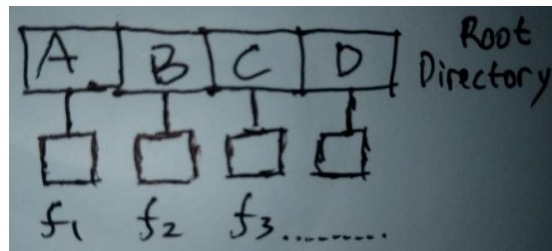
## 2. [15%] File Management

- a. [7%] Gambarkan dan bandingkan 3 jenis Directory System.

**Jawaban (186 Kata):**

### I) Single Level Directory Structures

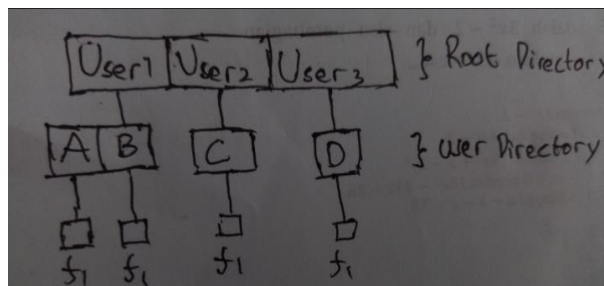
Hanya memiliki satu direktori yang disebut direktori root. Pengguna tidak diperbolehkan membuat subdirektori di bawah direktori root. Semua file yang dibuat oleh beberapa pengguna hanya ada di direktori root dan memiliki nama yang unik satu dengan yang lainnya.



kelemahannya yaitu pengguna tidak dapat menggunakan nama file yang sama yang digunakan oleh pengguna lain dalam sistem. Jika file dengan nama yang sama dibuat, file lama akan dihancurkan terlebih dahulu dan diganti dengan file baru yang memiliki nama yang sama.

### II) Two Level Directory Structures

Pengguna membuat direktori langsung di dalam direktori root. Tetapi sekali pengguna membuat direktori tersebut, selanjutnya dia tidak dapat membuat subdirektori di dalam direktori itu.

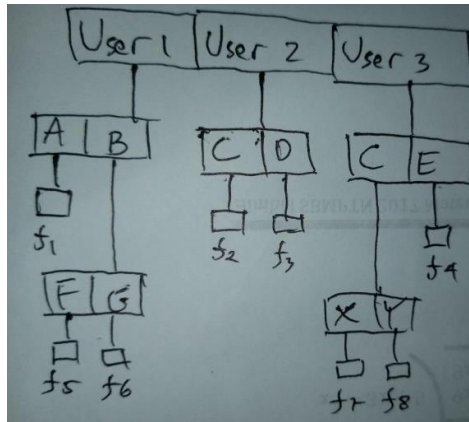


memungkinkan setiap pengguna untuk menyimpan file mereka secara terpisah di dalam direktorinya sendiri. Struktur ini memungkinkan untuk menggunakan nama yang sama untuk file tetapi di bawah direktori pengguna yang berbeda.

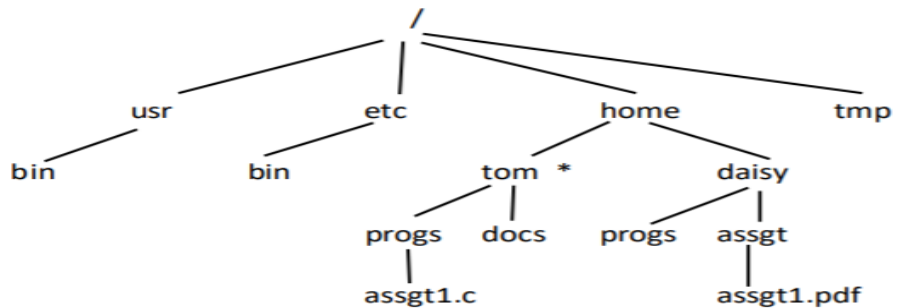
NAMA : EDWARD      Mata Kuliah : Operating System  
 NIM : 2201741971      Kode Mata Kuliah : COMP6153  
 KELAS : LB-08      Fakultas / Departemen : School of Computer Science

### III) Tree Directory Structures

Kecuali direktori root, setiap direktori atau file hanya memiliki satu direktori induk. Jadi, ada pemisahan total antara pengguna yang memberikan kebebasan penamaan lengkap. Di sini, jika seorang pengguna ingin mengakses file pengguna lain, ia harus melalui dua atau lebih direktori.



- b. [8%] Misalkan pada suatu Sistem Operasi terdapat directory dan file seperti tertulis dibawah ini:



Tanda (\*) menunjukkan posisi saat ini (**current directory**)

Bagaimana cara mengakses file-file berikut ini:

- i) assgt1.c menggunakan absolute pathname

**Jawaban:** /home/tom/progs/assgt1.c

- ii) assgt1c menggunakan relative directory

**Jawaban:** progs/assgt1.c

- iii) assgt1.pdf menggunakan absolute pathname

**Jawaban:** /home/daisy/assgt/assgt1.pdf

- iv) assgt1.pdf menggunakan relative pathname

**Jawaban:** ../daisy/assgt/assgt1.pdf

NAMA : EDWARD      Mata Kuliah : Operating System  
NIM : 2201741971      Kode Mata Kuliah : COMP6153  
KELAS : LB-08      Fakultas / Departemen : School of Computer Science

### 3. [20%] Memory Management

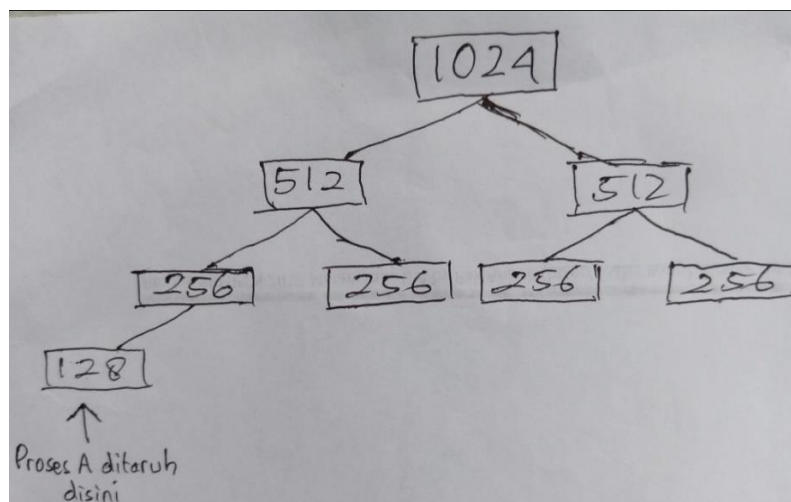
- a. [10%] Perusahaan XYZ mempunyai 1 program aplikasi Internal yang mempunyai 5 process yang berjalan bergantian sesuai permohonan terhadap alokasi memorinya. Process tersebut dijalankan pada Personal Computer yang menggunakan memori 1M. Berikut urutannya permohonan prosesnya:

- Process A Request 100k
- Process B Request 240k
- Process C Request 64k
- Process D Request 512k
- Release Process B
- Release Process D
- Process E Request 75k
- Release Process C
- Release Process A
- Release Process E

Gambarkan dan simulasikan permohonan dan pelepasan memori pada sistem tersebut, dan hitung internal fragmentasi yang terjadi pada setiap tahapnya menggunakan Buddy System.

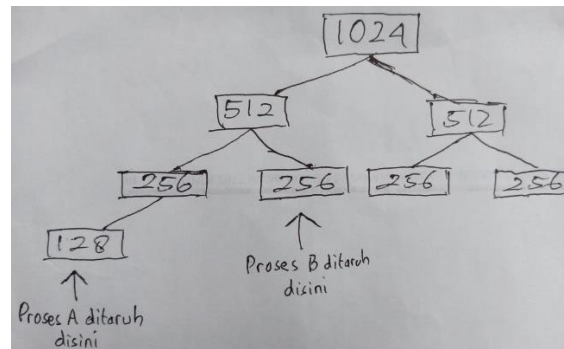
**Jawaban:**

#### D) Process A Request 100k



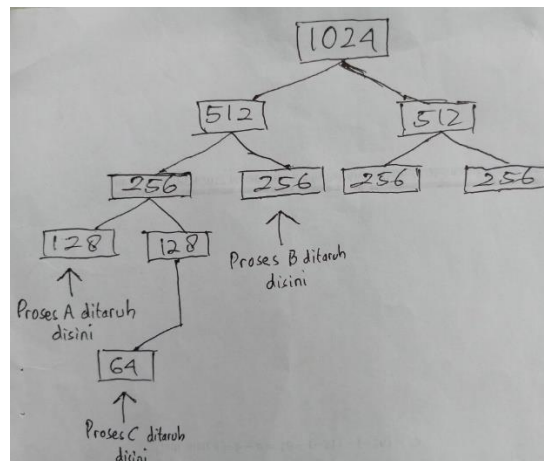
**Internal Fragmentasi yang terjadi berjumlah 28k**

**NAMA : EDWARD**      **Mata Kuliah**      **: Operating System**  
**NIM : 2201741971**      **Kode Mata Kuliah**      **: COMP6153**  
**KELAS : LB-08**      **Fakultas / Departemen**      **: School of Computer Science**  
**II) Process B Request 240k**



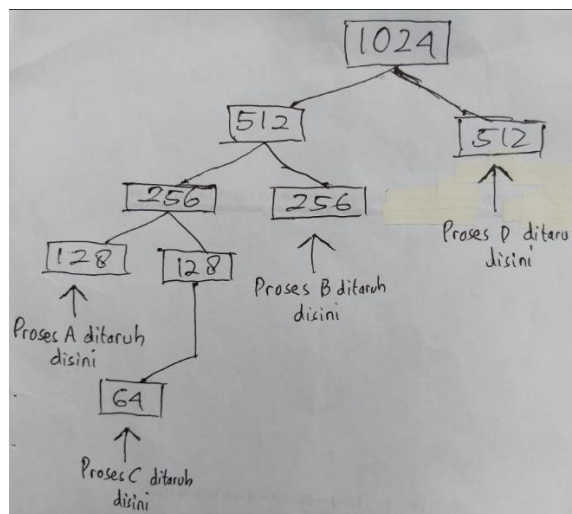
**Internal Fragmentasi yang terjadi berjumlah 16k**

**III) Process C Request 64k**



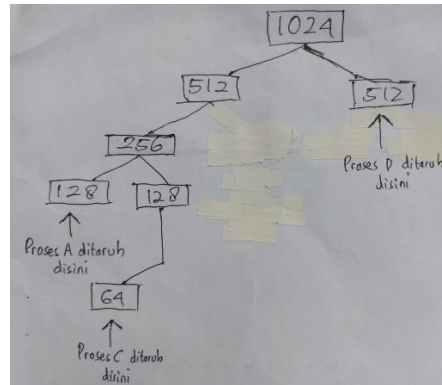
**Internal Fragmentasi yang terjadi berjumlah 0k**

**IV) Process D Request 512k**



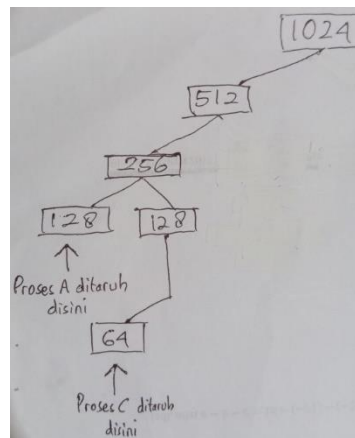
**Internal Fragmentasi yang terjadi berjumlah 0k**

**NAMA : EDWARD**      **Mata Kuliah : Operating System**  
**NIM : 2201741971**      **Kode Mata Kuliah : COMP6153**  
**KELAS : LB-08**      **Fakultas / Departemen : School of Computer Science**  
**V) Release Process B**



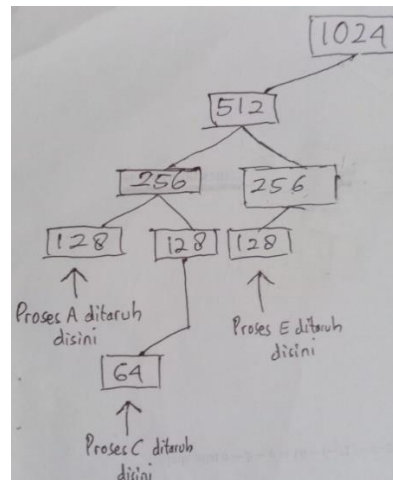
**Internal Fragmentasi yang terjadi berjumlah 0k**

**VI) Release Process D**



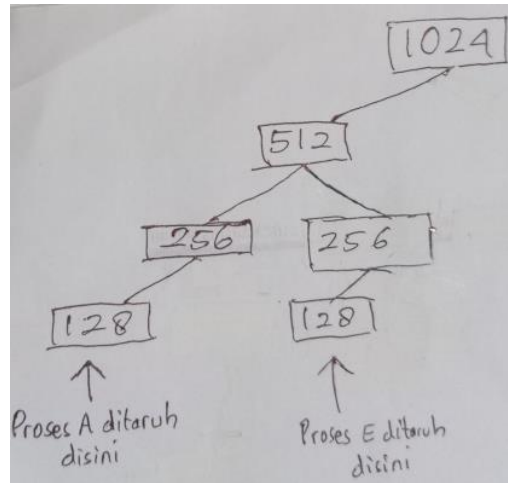
**Internal Fragmentasi yang terjadi berjumlah 0k**

**VII) Process E Request 75k**



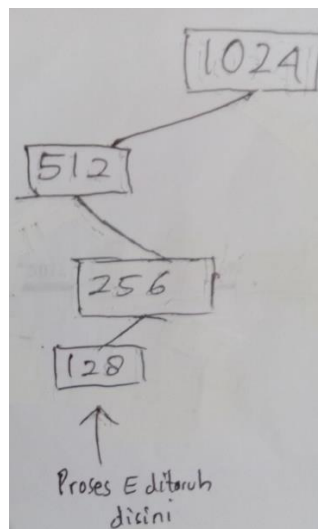
**Internal Fragmentasi yang terjadi berjumlah 53k**

**NAMA : EDWARD**      **Mata Kuliah**      **: Operating System**  
**NIM : 2201741971**    **Kode Mata Kuliah**    **: COMP6153**  
**KELAS : LB-08**        **Fakultas / Departemen**    **: School of Computer Science**  
**VIII) Release Process C**



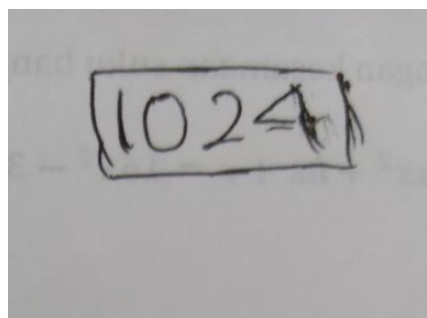
**Internal Fragmentasi yang terjadi berjumlah 0k**

**IX) Release Process A**



**Internal Fragmentasi yang terjadi berjumlah 0k**

**X) Release Process E**



**Internal Fragmentasi yang terjadi berjumlah 0k**

**NAMA : EDWARD**      **Mata Kuliah : Operating System**  
**NIM : 2201741971**      **Kode Mata Kuliah : COMP6153**  
**KELAS : LB-08**      **Fakultas / Departemen : School of Computer Science**  
 b. [10%] Misalkan pada suatu saat kondisi memori sbb:

				Pointer															
Partition Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	13		
Size (KB)	32	64	256	128	256	160	128	512	512	256	128	128	128	512	256	128	512		

Partisi berwarna abu-abu adalah partisi yang sudah terisi.  
 Berikut ini adalah urutan proses yang membutuhkan memori:

P1	100 KB
P2	30 kb
P3	50 KB
P4	32 KB
P5	300 KB
P6	500 KB

Tentukan pada partisi mana memori akan dialokasikan untuk proses-proses tersebut jika menggunakan algoritma:

**i) First Fit**

**Jawaban:**

Mencari sepanjang daftar yang berisi besarnya ukuran memori yang dibutuhkan oleh proses dalam antrian beserta ukuran memori yang tersedia pada saat itu. **Dari tabel, Partisi Ganjil selalu terisi penuh, maka cek Partisi Genap saja dikarenakan partisi genap tersedia ruang untuk dialokasikan.**

- P1 berukuran 100KB.** Ketika cek dengan partisi 0, maka tidak bisa dialokasikan di partisi ini. **Partisi 2 memiliki ruang 256, maka P1 bisa ditempatkan disana dengan Sisa 156. Maka P1 akan dialokasikan di Partisi 2.**
- P2 berukuran 30KB.** Cek Partisi 0. **Partisi 0 memiliki ruang 32, maka P2 bisa ditempatkan disana dengan sisa 2. Maka P2 akan dialokasikan di Partisi 0.**
- P3 berukuran 50KB.** Cek Partisi 0, sudah terisi oleh P2. Cek partisi 2, sudah diisi oleh P1. Cek Partisi 4. **Partisi 4 memiliki ruang 256, maka P3 bisa ditempatkan disana dengan sisa 206. Maka P3 akan dialokasikan di partisi 4.**



NAMA : EDWARD      Mata Kuliah : Operating System  
NIM : 2201741971      Kode Mata Kuliah : COMP6153  
KELAS : LB-08      Fakultas / Departemen : School of Computer Science

- **P4 berukuran 32KB.** Partisi 0 sudah terisi oleh P2 , maka tidak bisa. Cek Partisi 2, tidak bisa ditempatkan disana karena sudah terisi oleh P1. Partisi 4 juga sudah dialokasikan untuk P3. Ketika cek dengan Partisi 6, **Partisi 6 memiliki ruang 128, maka P4 bisa ditempatkan disana dengan sisa 96. Maka P4 akan dialokasikan di Partisi 6.**
- **P5 berukuran 300KB.** Cek Partisi 0, terisi oleh P2. Partisi 2 tidak bisa ditempatkan karena sudah terisi oleh P1. Ketika cek Partisi 4, sudah terisi oleh P3. Selanjutnya cek Partisi 6, sudah terisi juga oleh P4, maka tidak bisa disana. Cek Partisi 8. **Partisi 8 memiliki ruang 512, maka P5 bisa ditempatkan disana dengan sisa 212. Maka P5 akan dialokasikan di Partisi 8.**
- **P6 berukuran 500KB.** Cek Partisi 0, sudah terisi oleh P2. Cek partisi 2, sudah diisi oleh P1. Cek Partisi 4 terisi oleh P3. Ketika Cek Partisi 6, sudah terisi oleh P4. Tidak bisa dialokasikan di Partisi 8 karena sudah diisi oleh P5. Cek Partisi 10, tidak bisa karena memori kurang dari yang dibutuhkan. Begitu pula dengan Partisi 12 sama dengan Partisi 10.Selanjutnya Cek Partisi 14, tidak bisa dialokasikan disana karena tidak mencukupi. Cek Partisi 16. **Partisi 16 memiliki ruang 512, maka P6 bisa ditempatkan disana dengan sisa 12. Maka P6 akan dialokasikan di Partisi 16.**

NAMA : EDWARD      Mata Kuliah : Operating System  
 NIM : 2201741971      Kode Mata Kuliah : COMP6153  
 KELAS : LB-08      Fakultas / Departemen : School of Computer Science  
 ii) Best Fit

**Jawaban:**

Mencari Partisi dengan ukuran yang hampir sama dengan yang dibutuhkan oleh proses untuk menampung proses yang bersangkutan. **Dari tabel, Partisi Ganjil selalu terisi penuh, maka cek Partisi Genap saja dikarenakan partisi genap tersedia ruang untuk dialokasikan.**

- **P1 berukuran 100KB.**

Partisi	Sisa Memori
0	$32 - 100 = -68$ (TIDAK BISA)
2	$256 - 100 = 156$
4	$256 - 100 = 156$
<b>6</b>	<b><math>128 - 100 = 28</math></b>
8	$512 - 100 = 412$
10	$128 - 100 = 28$
12	$128 - 100 = 28$
14	$256 - 100 = 156$
16	$512 - 100 = 412$

**Maka P1 akan dialokasikan di Partisi 6.**

- **P2 berukuran 30KB.**

Partisi	Sisa Memori
<b>0</b>	<b><math>32 - 30 = 2</math></b>
2	$256 - 30 = 226$
4	$256 - 30 = 226$
<b>6</b>	<b><math>128 - 100 = 28</math></b>
8	$512 - 30 = 482$
10	$128 - 30 = 98$
12	$128 - 30 = 98$
14	$256 - 30 = 226$
16	$512 - 30 = 482$

**Maka P2 akan dialokasikan di Partisi 0.**

NAMA : EDWARD      Mata Kuliah : Operating System  
 NIM : 2201741971      Kode Mata Kuliah : COMP6153  
 KELAS : LB-08      Fakultas / Departemen : School of Computer Science

- **P3 berukuran 50KB.**

Partisi	Sisa Memori
0	$32 - 30 = 2$
2	$256 - 50 = 206$
4	$256 - 50 = 206$
6	$128 - 100 = 28$
8	$512 - 50 = 462$
10	$128 - 50 = 78$
12	$128 - 50 = 78$
14	$256 - 50 = 206$
16	$512 - 50 = 462$

Maka P3 akan dialokasikan di Partisi 10.

- **P4 berukuran 32KB.**

Partisi	Sisa Memori
0	$32 - 30 = 2$
2	$256 - 32 = 224$
4	$256 - 32 = 224$
6	$128 - 100 = 28$
8	$512 - 32 = 480$
10	$128 - 50 = 78$
12	$128 - 32 = 96$
14	$256 - 32 = 224$
16	$512 - 32 = 480$

Maka P4 akan dialokasikan di Partisi 12.

NAMA : EDWARD      Mata Kuliah : Operating System  
 NIM : 2201741971      Kode Mata Kuliah : COMP6153  
 KELAS : LB-08      Fakultas / Departemen : School of Computer Science

- **P5 berukuran 300KB.**

Partisi	Sisa Memori
0	$32 - 30 = 2$
2	$256 - 300 = -44$ (TIDAK BISA)
4	$256 - 300 = -44$ (TIDAK BISA)
6	$128 - 100 = 28$
8	$512 - 300 = 212$
10	$128 - 50 = 78$
12	$128 - 32 = 96$
14	$256 - 300 = -44$ (TIDAK BISA)
16	$512 - 300 = 212$

Maka P5 akan dialokasikan di Partisi 8.

- **P6 berukuran 500KB.**

Partisi	Sisa Memori
0	$32 - 30 = 2$
2	$256 - 500 = -244$ (TIDAK BISA)
4	$256 - 500 = -244$ (TIDAK BISA)
6	$128 - 100 = 28$
8	$512 - 300 = 212$
10	$128 - 50 = 78$
12	$128 - 32 = 96$
14	$256 - 500 = -244$ (TIDAK BISA)
16	$512 - 500 = 12$

Maka P6 akan dialokasikan di Partisi 16.

NAMA : EDWARD      Mata Kuliah : Operating System  
NIM : 2201741971      Kode Mata Kuliah : COMP6153  
KELAS : LB-08      Fakultas / Departemen : School of Computer Science

iii) Next Fit

**Jawaban:**

Hampir sama persis dengan First Fit, kecuali Next Fit meneruskan proses pencarian terhadap partisi yang cukup besar untuk sebuah proses mulai dari partisi sebelumnya yang telah sesuai dengan proses sebelumnya. Pendek kata, algoritma ini **tidak memulai pencarian dari awal, melainkan dari POINTER yang ditunjuk. Dan dari tabel cek Partisi Genap saja dikarenakan partisi genap tersedia ruang untuk dialokasikan dimulai dari Partisi 4.**

- **P1 berukuran 100KB.**

Partisi	Sisa Memori
4	$256 - 100 = 156$
6	$128 - 100 = 28$
8	$512 - 100 = 412$
10	$128 - 100 = 28$
12	$128 - 100 = 28$
14	$256 - 100 = 156$
16	$512 - 100 = 412$

**Maka P1 akan dialokasikan di Partisi 4.**

- **P2 berukuran 30KB.**

Partisi	Sisa Memori
4	$256 - 100 = 156$
6	$128 - 30 = 98$
8	$512 - 30 = 482$
10	$128 - 30 = 98$
12	$128 - 30 = 98$
14	$256 - 30 = 226$
16	$512 - 30 = 482$

**Maka P2 akan dialokasikan di Partisi 6.**

NAMA : EDWARD      Mata Kuliah : Operating System  
 NIM : 2201741971      Kode Mata Kuliah : COMP6153  
 KELAS : LB-08      Fakultas / Departemen : School of Computer Science

- **P3 berukuran 50KB.**

Partisi	Sisa Memori
4	$256 - 100 = 156$
6	$128 - 30 = 98$
8	$512 - 50 = 462$
10	$128 - 50 = 78$
12	$128 - 50 = 78$
14	$256 - 50 = 206$
16	$512 - 50 = 462$

Maka P3 akan dialokasikan di Partisi 8.

- **P4 berukuran 32KB.**

Partisi	Sisa Memori
4	$256 - 100 = 156$
6	$128 - 30 = 98$
8	$512 - 50 = 462$
10	$128 - 32 = 96$
12	$128 - 32 = 96$
14	$256 - 32 = 224$
16	$512 - 32 = 480$

Maka P4 akan dialokasikan di Partisi 10.

NAMA : EDWARD      Mata Kuliah : Operating System  
 NIM : 2201741971      Kode Mata Kuliah : COMP6153  
 KELAS : LB-08      Fakultas / Departemen : School of Computer Science

- **P5 berukuran 300KB.**

Partisi	Sisa Memori
4	$256 - 100 = 156$
6	$128 - 30 = 98$
8	$512 - 50 = 462$
10	$128 - 32 = 96$
12	$128 - 300 = -172$ (TIDAK BISA)
14	$256 - 300 = -44$ (TIDAK BISA)
16	$512 - 300 = 212$

Maka P5 akan dialokasikan di Partisi 16.

- **P6 berukuran 500KB.**

Partisi	Sisa Memori
4	$256 - 100 = 156$
6	$128 - 30 = 98$
8	$512 - 50 = 462$
10	$128 - 32 = 96$
12	$128 - 500 = -172$ (TIDAK BISA)
14	$256 - 500 = -244$ (TIDAK BISA)
16	$512 - 300 = 212$

Maka P6 Tidak Bisa Dialokasikan di Partisi Manapun.

**NAMA : EDWARD**      **Mata Kuliah**      **: Operating System**  
**NIM : 2201741971**      **Kode Mata Kuliah**      **: COMP6153**  
**KELAS : LB-08**      **Fakultas / Departemen**      **: School of Computer Science**

#### 4. [20%] Virtual Memory

Anda ingin membandingkan beberapa algoritma page replacement, agar dapat menentukan algoritma mana yang terbaik. Untuk itu Anda melakukan analisis terhadap algoritma: Optimal, FIFO, LRU dan Clock. Anda menggunakan kasus sbb: menggunakan 4 memory frame dengan urutan Page Address Stream yang membutuhkan memory: 3 2 1 4 5 2 3 4 2

1. Hitung jumlah page fault yang terjadi untuk setiap algoritma, dan berdasarkan hasil analisis Anda, algoritma mana yang terbaik dalam kasus ini? Jelaskan.

**Jawaban:**

##### a) Algoritma Optimal

- **First Step (Page Fault = 1)**

3			
---	--	--	--

- **Second Step (Page Fault = 2)**

3	2		
---	---	--	--

- **Third Step (Page Fault = 3)**

3	2	1	
---	---	---	--

- **Fourth Step (Page Fault = 4)**

3	2	1	4
---	---	---	---

- **Fifth Step (Page Fault = 5)**

3	2	5	4
---	---	---	---

- **Sixth Step (Page Fault = 5)**

3	2	5	4
---	---	---	---

- **Seventh Step (Page Fault = 5)**

3	2	5	4
---	---	---	---

- **Eighth Step (Page Fault = 5)**

3	2	5	4
---	---	---	---

- **Ninth Step (Page Fault = 5)**

3	2	5	4
---	---	---	---

- **Tenth Step (Page Fault = 6)**

1	2	5	4
---	---	---	---

**Dengan Algoritma Optimal, Page Fault yang terjadi sebanyak 6.**



NAMA : EDWARD      Mata Kuliah : Operating System  
 NIM : 2201741971      Kode Mata Kuliah : COMP6153  
 KELAS : LB-08      Fakultas / Departemen : School of Computer Science

**b) Algoritma FIFO**

- **First Step (Page Fault = 1)**

3			
---	--	--	--

- **Second Step (Page Fault = 2)**

3	2		
---	---	--	--

- **Third Step (Page Fault = 3)**

3	2	1	
---	---	---	--

- **Fourth Step (Page Fault = 4)**

3	2	1	4
---	---	---	---

- **Fifth Step (Page Fault = 5)**

5	2	1	4
---	---	---	---

- **Sixth Step (Page Fault = 5)**

5	2	1	4
---	---	---	---

- **Seventh Step (Page Fault = 6)**

5	3	1	4
---	---	---	---

- **Eighth Step (Page Fault = 6)**

5	3	1	4
---	---	---	---

- **Ninth Step (Page Fault = 7)**

5	3	2	4
---	---	---	---

- **Tenth Step (Page Fault = 8)**

5	3	2	1
---	---	---	---

**Dengan Algoritma FIFO, Page Fault yang terjadi sebanyak 8.**

**NAMA : EDWARD**      **Mata Kuliah**      **: Operating System**  
**NIM : 2201741971**      **Kode Mata Kuliah**      **: COMP6153**  
**KELAS : LB-08**      **Fakultas / Departemen**      **: School of Computer Science**  
**c) Algoritma LRU**

- **First Step (Page Fault = 1)**

3			
---	--	--	--

- **Second Step (Page Fault = 2)**

3	2		
---	---	--	--

- **Third Step (Page Fault = 3)**

3	2	1	
---	---	---	--

- **Fourth Step (Page Fault = 4)**

3	2	1	4
---	---	---	---

- **Fifth Step (Page Fault = 5)**

5	2	1	4
---	---	---	---

- **Sixth Step (Page Fault = 5)**

5	2	1	4
---	---	---	---

- **Seventh Step (Page Fault = 6)**

5	2	3	4
---	---	---	---

- **Eighth Step (Page Fault = 6)**

5	2	3	4
---	---	---	---

- **Ninth Step (Page Fault = 6)**

5	2	3	4
---	---	---	---

- **Tenth Step (Page Fault = 7)**

1	2	3	4
---	---	---	---

**Dengan Algoritma LRU, Page Fault yang terjadi sebanyak 7.**

**NAMA : EDWARD**      **Mata Kuliah : Operating System**  
**NIM : 2201741971**      **Kode Mata Kuliah : COMP6153**  
**KELAS : LB-08**      **Fakultas / Departemen : School of Computer Science**  
**d) Algoritma Clock**

- **First Step (Page Fault = 1)**

3			
---	--	--	--

- **Second Step (Page Fault = 2)**

3	2		
---	---	--	--

- **Third Step (Page Fault = 3)**

3	2	1	
---	---	---	--

- **Fourth Step (Page Fault = 4) {0,0,0,0}**

3	2	1	4
---	---	---	---

- **Fifth Step (Page Fault = 5) {0,0,0,0}**

5	2	1	4
---	---	---	---

- **Sixth Step (Page Fault = 5) {0,1,0,0}**

5	2	1	4
---	---	---	---

- **Seventh Step (Page Fault = 6) {0,0,0,0}**

5	2	3	4
---	---	---	---

- **Eighth Step (Page Fault = 6) {0,0,0,0}**

5	2	3	4
---	---	---	---

- **Ninth Step (Page Fault = 6) {0,0,0,1}**

5	2	3	4
---	---	---	---

- **Tenth Step (Page Fault = 7) {0,1,0,1}**

5	2	1	4
---	---	---	---

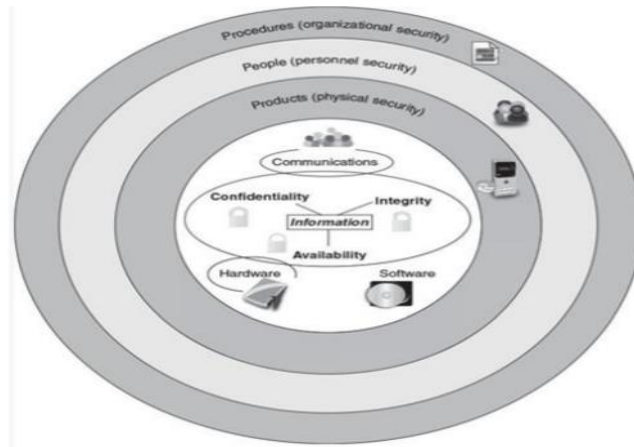
**Dengan Algoritma Clock, Page Fault yang terjadi sebanyak 7.**

**Algoritma yang terbaik dalam kasus ini adalah OPTIMAL, karena Page Fault yang terjadi berjumlah paling sedikit yaitu 6.**

NAMA : EDWARD                      Mata Kuliah : Operating System  
NIM : 2201741971                      Kode Mata Kuliah : COMP6153  
KELAS : LB-08                      Fakultas / Departemen : School of Computer Science

**5. [15%] Security**

Jelaskan dengan bahasa sendiri mengenai hubungan antara asset (hardware, software dan Network), CIA (confidentiality, Integrity dan Availability) dengan 3 layers perlindungan informasi seperti gambar di bawah ini:



**Jawaban (161 Kata) :**

- **Confidentiality**

Upaya pencegahan bagi mereka yang tidak berkepentingan untuk dapat mencapai informasi . Contohnya E-mail maupun data-data perdagangan dari perusahaan.

- **Integrity**

Bentuk pencegahan terhadap kemungkinan penghapusan informasi oleh mereka yang tidak berhak. Contohnya ketika Mengirim pesan ke seseorang, pesan tersebut harus tepat tuju kepada orang yang dituju.

- **Availability**

Upaya pencegahan ditahannya informasi atau sumber daya terkait oleh mereka yang tidak berhak. Contohnya seperti “messaging system” maka pesan itu harus dapat dibaca oleh siapapun yang dialamatkan atau yang diarahkan, sewaktu mereka ingin membacanya.

**NAMA : EDWARD      Mata Kuliah : Operating System**  
**NIM : 2201741971      Kode Mata Kuliah : COMP6153**  
**KELAS : LB-08      Fakultas / Departemen : School of Computer Science**

- **Access Control**

Untuk mengatur 'siapa boleh melakukan apa', 'dari mana boleh ke mana'. Contoh seperti antar jaringan, ACL proxy (pembatasan bandwidth). Contoh lainnya seperti Dosen dan Mahasiswa, dimana dosen dapat memberikan tugas kepada mahasiswa, sedangkan Mahasiswa dapat mengumpulkan tugasnya kepada Dosen.

- **Autentikasi**

Untuk menentukan atau mengonfirmasi bahwa seseorang (atau sesuatu) adalah autentik atau asli. Contoh autentikasi biasanya terjadi pada saat login atau permintaan akses.

- **Accountability**

Untuk keperluan pengecekan sehingga transaksi dapat dipertanggungjawabkan. Implementasi dapat berupa IDS/IPS (firewall), syslog (router).

<b>NAMA : EDWARD</b>	<b>Mata Kuliah</b>	<b>: Operating System</b>
<b>NIM : 2201741971</b>	<b>Kode Mata Kuliah</b>	<b>: COMP6153</b>
<b>KELAS : LB-08</b>	<b>Fakultas / Departemen</b>	<b>: School of Computer Science</b>

**6. [15%] Studi Kasus**

Pada arsitektur Windows, terdapat kernel dan Hardware Abstraction Layer (HAL). Jelaskan yang dimaksud dengan kernel dan HAL. Jelaskan hubungan antara kernel dan HAL.

**Jawaban (158 Kata):**

Kernel berguna sebagai jembatan yang menghubungkan berbagai aplikasi dan data processing yang di laksanakan di bagian perangkat keras. Jadi semua proses yang berangsur harus melewati kernel terlebih dahulu sebelum diarahkan ke bagian yang semestinya. Jika tidak melalui kernel, akan terjadi konflik satu sama lain dan dapat menyebabkan kerusakan yang fatal.

Hardware Abstraction Layer adalah layer yang menghubungkan operating system komputer untuk berkomunikasi dengan peralatan perangkat keras secara general atau abstrak. Jadi, sama seperti Kernel, Hardware Abstraction Layer juga berfungsi sebagai jembatan antara Central Processing Unit dan perangkat keras yang tersedia.

Hubungan antara Hardware Abstraction Layer dan Kernel bisa disimpulkan bahwa ketika suatu proses membutuhkan suatu perangkat keras untuk bekerja maka kernel akan memprosesnya dan mengirimkan pesan kepada Hardware Abstraction Layer terlebih dahulu karena Hardware Abstraction Layer adalah jembatan antara komputer dengan perangkat keras. Setelah perangkat keras yang berhubungan sudah dipastikan, maka Hardware Abstraction Layer akan mengumpan balikkan pesan kepada Kernel untuk mengakolasi memori serta memulai proses program tersebut.