

PR03 SOLUSI

NAMA: _____

NPM: _____

1. Sebuah sistem mempunyai ukuran page 2 KByte dengan jumlah page sebanyak 4096 page dan jumlah frame memori fisik sebanyak 2048 frame.

(Unit ukuran: 1KByte= 1024 Byte = 2^{10} Byte; 1 MByte = 2^{20} Byte)

- a. Berapa jumlah bit yang dibutuhkan untuk Logical Address?

$$\text{LA} = 12 (\text{Total Page}) + 11 (\text{Page Size}) = 23 \text{ Bit}$$

- b. Berapa jumlah bit yang dibutuhkan untuk Physical Address?

$$\text{PA} = 11 (\text{Total Frame}) + 11 (\text{Page Size}) = 22 \text{ Bit}$$

- c. Untuk alamat logik: 2000, Hitung

- a. Nomor Page (dalam desimal)

$$\text{PN} = 2000 / 2048 = 0, \dots = 0$$

- b. Offset (dalam desimal)

$$\text{Offset} = 2000 \% 2048 = 2000$$

- d. Berapa MByte kapasitas memori fisik?

$$\text{Kapasitas MF} = 2048 (\text{Total Frame}) \times 2048 (\text{Page Size}) = 2^{11} \times 2^{11} = 2^{22} = 4 \text{ Mbyte}$$

Cara Lain

$$\text{PA} = 11 (\text{Total Frame}) + 11 (\text{Page Size}) = 22 \text{ Bit}$$

$$\text{Kapasitas MF} = 2^{22} \times 1 \text{ Byte} = 4 \text{ Mbyte}$$

2. Sebuah sistem menggunakan page berukuran 256 bytes dengan 12 bit alamat virtual dan alamat fisik. Dibawah ini adalah page table yang digunakan. Karakter i menandakan *invalid*, artinya page tidak ada pada memori.

Page	Frame
0	i
1	5
2	i
3	1
4	8
5	9
6	7
7	E
8	4

9	D
A	i
B	F
C	i
D	3
E	i
F	i

Frame kosong yang tersedia secara berurutan mulai dari A, B dan C. Jika terjadi *invalid access*, maka page akan ditempatkan pada frame kosong mulai dari frame A. Tentukan alamat fisik dari alamat virtual berikut (dalam hexadesimal):

- a) 8A1 → 4A1
- b) 2BE → ABE
- c) 569 → 969

- d) A4D → B4D
- e) B51 → F51

3. Diketahui sebuah sistem mempunyai

- 16 bit logical address space
- Ukuran page: 1 KByte
- Ukuran 1 Page Table Entry: 2 Byte
- Jumlah proses: 10 Proses

a) Berapa Byte yang dibutuhkan untuk menampung 1 page table?

Alokasi bit untuk page table = 16-10 (bit page offset) = 4 bit

Byte untuk menampung Page Table = $2^4 \times 2 \text{ byte PTE} = 32 \text{ Byte}$

b) Berapa Byte yang dibutuhkan untuk menampung page table dari seluruh proses?

32 Byte x 10 = 320 Byte

4. Sebuah proses secara berurutan mengakses page-page sebagai berikut:

6, 7, 8, 8, 9, 6, 4, 6, 7, 8

Jika jumlah frame kosong adalah 4 frame , hitunglah jumlah page fault jika sistem menggunakan algoritma pergantian page:

a. FIFO

6	6	6	6	4	4	4	4
	7	7	7	7	6	6	6
		8	8	8	8	7	7
			9	9	9	9	8

PF= 8

b. LRU

6	6	6	6	6	6	6
	7	7	7	4	4	4
		8	8	8	7	7
			9	9	9	8

PF= 7

c. OPTIMAL

6	6	6	6	6
	7	7	7	7
		8	8	8
			9	4

PF= 5

5. Sebuah sistem mempunyai spesifikasi sebagai berikut:

- Data Array: `int A[][] = new int A[10][10]`.
 - o Penyusunan array pada memori dimulai dari `A[0][0]`, `A[0][1]` ..., `A[0][9]`, `A[1][0]`, `A[1][1]`, ..., `A[9][9]`
- Satu integer berukuran 4 byte
- Ukuran 1 page = 80 Byte
- Jumlah frame = 1 frame

Berdasarkan informasi diatas, tentukan jumlah page fault masing-masing potongan kode program berikut:

Isi page:

Page 0	A[0][0]	Page 2	A[4][0]	Page 4	A[8][0]
	A[0][1]		A[4][1]		A[8][1]

	A[0][9]		A[4][9]		A[8][9]
	A[1][0]		A[5][0]		A[9][0]
	A[1][1]		A[5][1]		A[9][1]

	A[1][9]		A[5][9]		A[9][9]
Page 1	A[2][0]	Page 3	A[6][0]		
	A[2][1]		A[6][1]		
		
	A[2][9]		A[6][9]		
	A[3][0]		A[7][0]		
	A[3][1]		A[7][1]		
		
	A[3][9]		A[7][9]		

a)

```
for (int j = 0; j < 10 ; j++)
    for (int i=0; i < 10; i++)
        A[i][j]=0;
```

Jumlah frame = 1 frame

Page yang diakses:

0,0,1,1,2,2,3,3,4,4 → interaksi j = 0 = 5 PF

0,0,1,1,2,2,3,3,4,4 → interaksi j = 1 = 5 PF

...

...

0,0,1,1,2,2,3,3,4,4 → interaksi j = 9 = 5 PF

Total Page Fault = 10 x 5 PF = 50 PF

```
b)   for (int i=0; i < 10; i++)  
        for (int j = 0; j < 10 ; j++)  
            A[i][j]=0;
```

Jumlah frame = 1 frame

Page yang diakses:

0,0,0,0,0,0,0,0,0,0 → interaksi i = 0 = 1 PF

0,0,0,0,0,0,0,0,0,0 → interaksi i = 1

1,1,1,1,1,1,1,1,1,1 → interaksi i = 2 = 1 PF

1,1,1,1,1,1,1,1,1,1 → interaksi i = 3

...

4,4,4,4,4,4,4,4,4,4 → interaksi i = 8 = 1 PF

4,4,4,4,4,4,4,4,4,4 → interaksi i = 9

Total Page Fault = 5 x 1 PF = 5 PF

c). Berapa jumlah page yang dibutuhkan untuk menampung data array?

**int A[][]= new int A[10][10] → jumlah data = 10 x 10 = 100 elemen data
bertipekan integer**

Total byte = 100 elemen data x 4 byte integer = 400 byte.

Jumlah page = total byte data/ukuran 1 page = 400/80 = 5 page