

Kelas A.

Tugas 3 Praktikum Analgo

- 1] Untuk  $T(n) = 2 + 4 + 8 + 16 + \dots + 2^n$ , tentukan nilai :  $C$ ,  $f(n)$ ,  $n_0$  dan notasi sehingga  $T(n) = O(f(n))$  jika  $T(n) \leq C$  untuk semua  $n \geq n_0$ .

Jawab. $T(n)$  adalah deret geometri

$$S_n = a \frac{(r^n - 1)}{r - 1} = \frac{2(2^n - 1)}{2 - 1} = 2 \cdot 2^n - 2$$

$$T_n = 2 \cdot 2^n - 2 = 2^{n+1} - 2$$

$$\boxed{T(n) = O(2^n)} \text{ dan } \boxed{f(n) = 2^n}$$

Untuk  $C = 2$  dan untuk semua  $n \geq 0$  dan  $n_0 = 0$  dapat dibuktikan  $T(n) \leq C \cdot f(n)$ .

$$\frac{2^{n+1} - 2}{2^n} \leq C \cdot 2^n ; 2^n$$

$$\frac{2^{n+1}}{2^n} - \frac{2}{2^n} \leq C$$

$$2 - \frac{2}{2^n} \leq C ; n=1$$

$$2 - 1 \leq C$$

$$1 \leq C$$

$$\boxed{n_0 = 1 \quad C \geq 1}$$

- 2] Buktikan bahwa untuk konstanta-konstanta positif  $p$ ,  $q$ , dan  $r$ :  $T(n) = pn^2 + qn + r$  adalah  $O(n^2)$ ,  $\Omega(n^2)$  dan  $\Theta(n^2)$ .

Jawab.

$$\bullet pn^2 + qn + r = O(n^2)$$

$$pn^2 + qn + r \leq n^2 C ; 2$$

$$p + \frac{q}{n} + \frac{r}{n^2} \leq C ; n=1$$

$$p + q + r \leq C$$

$$\boxed{\text{Terbukti } n_0 = 1 \text{ dan } C \geq p + q + r}$$

$$\bullet pn^2 + qn + r = \Omega(n^2)$$

$$pn^2 + qn + r \geq n^2 C ; n^2$$

$$p + \frac{q}{n} + \frac{r}{n^2} \geq C ; n=1$$

$$p + q + r \geq C$$

$$\boxed{\text{Terbukti } n_0 = 1 \text{ dan } C \leq p + q + r}$$

$$\bullet \text{ karena Big } O = \text{Big } \Omega = n^2 \text{ maka Big } \Theta = n^2$$

- 3] Tentukan waktu kompleksitas asimptotik (Big-O, Big- $\Omega$ , Big- $\Theta$ ) dari kode program berikut:

```

O(n) for k ← 1 to n do
O(n)   for i ← 1 to n do
O(n)     for j ← 1 to n do
O(1)       wij ← wij or wik and wjk
           endfor
         endfor
       endfor

```

Jawab.

$$T(n) = O(n) + O(n) + O(n) + O(1) = O(n^3) \rightarrow f(n)$$

$$\bullet \text{ Big-O} = O(f(n)) = O(n^3)$$

$$\bullet \text{ Big-}\Omega = \Omega(f(n)) = \Omega(n^3)$$

$$\bullet \text{ karena Big-O} = \text{Big-}\Omega \text{ maka Big-}\Theta = \Theta(f(n)) = \Theta(n^3)$$

- 4] Tuliskan algoritma untuk menjumlahkan 2 buah matriks yang masing-masing berukuran  $n \times n$ . Berapa kompleksitas waktu asimptotik nya yang dinyatakan dalam Big-O, Big- $\Omega$ , dan Big- $\Theta$ .  
 $\rightarrow T(n)?$  dan



Jawab.

```
for i ← 1 to n do      O(n)
  for j ← 1 to n do    O(n)
    hasil [i,j] ← a[i,j] + b[i,j]
  endfor
endfor.
```

$$T(n) = O(n) + O(n) \\ = O(n^2) \rightarrow f(n)$$

- Big-O =  $O(f(n)) = O(n^2)$
- Big-Ω =  $\Omega(f(n)) = \Omega(n^2)$
- Big-O = Big-Ω maka  
Big-Θ =  $\Theta(n^2)$

5] Tulislah algoritma untuk menyalin (copy) isi sebuah larik lain. Ukuran elemen larik adalah n elemen. Berapa kompleksitas waktu  $T(n)$ ? dan berapa kompleksitas waktu asimptotiknya?

Jawab.

```
for i ← 1 to n do      O(n)
  b[i-1] ← a[i-1]
endfor
```

$$T(n) = O(n) \rightarrow f(n)$$

- Big-O =  $O(n)$
- Big-Ω =  $\Omega(n)$
- Big-Θ =  $\Theta(n)$

6] Diberikan algoritma bubble sort.

- Hitung berapa jumlah operasi perbandingan?
- Berapa kali maks pertukaran?
- Hitung Big-O, Big-Ω, Big-Θ!

Jawab.

a) Pass	Jumlah operasi	$T(n) = (n-1) + (n-2) + \dots + 1$
1	n-1	
2	n-2	
3	n-3	
⋮	⋮	
n	1	
		$= \frac{n(n-1)}{2}$
		$= \frac{(n^2-n)}{2}$

$$b) \text{ Maks pertukaran elemen} = \frac{n(n-1)}{2}$$

c) Kompleksitas waktu asimptotik:

$$\text{Big-O} \rightarrow T(n) = \frac{n^2-n}{2} = O(n^2)$$

$$\text{Big-}\Omega \rightarrow T(n) = n^2+n = \Omega(n^2)$$

$$\text{Big-}\Theta \rightarrow T(n) = n^2+n = \Theta(n^2)$$

7] Untuk menyelesaikan problem X dengan ukuran N tersedia 3 macam algoritma:

- Algoritma A;  $T(n) = O(\log N)$
- Algoritma B;  $T(n) = O(N \log N)$
- Algoritma C;  $T(n) = O(N^2)$

$N=8$ . Mana algoritma yang paling cepat?

Jawab.

- Algoritma A =  $O(\log 8)$
- Algoritma B =  $O(8 \log 8)$
- Algoritma C =  $O(64)$

Yang paling cepat adalah algoritma A, karena semakin kecil angka didalam kurung /  $f(n)$  maka semakin sedikit operasi yang dikerjakan.

$$8] P(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + x(a_{n-1} + a_n x))))$$

Hitung berapa operasi perkalian dan penjumlahan. Tentukan Big-O. Mana yang terbaik, P atau P2?

Jawab.

- Algoritma P → Jumlah = n kali kali = n kali  
 $T(n) = n+n = 2n = n$
- Algoritma P2 →  $T_2(n) = 1+n = O(n)$
- P dan P2 sama-sama baik karena big-O nya sama-sama  $O(n)$