# Notasi Asimtotik (Part II)

Sukmawati Nur Endah

Departemen Informatika UNDIP

# Ingat Kembali : Notasi Asimtotik

- Fungsi pertumbuhan mempunyai notasi ➔ notasi Asimtotik
- Notasi Asimtotik
  - Big O (O besar)
  - Big Ω (Omega Besar)
  - Big Θ (Tetha Besar)

# Big O

- Notasi Big-O digunakan untuk :
  - mengkategorikan algoritma ke dalam fungsi yang menggambarkan batas atas (*upper limit*) dari pertumbuhan sebuah fungsi ketika masukan dari fungsi tersebut bertambah banyak

- Definisi:

  $T(n) = O(f(n))$ (artinya $T(n)$ berorde paling besar $f(n)$) bila terdapat konstanta $C$ dan $n_0$ sedemikian sehingga $T(n) \leq C.\ f(n)$, untuk $n \geq n_0$

# Catatan:

- Dari def : $T(n) = O(f(n))$ bukan sebaliknya
- Simbol "=" menyatakan "adalah" bukan "sama dengan"
- Untuk menunjukkan $T(n) = O(f(n))$, hanya perlu menemukan $C$ dan $n_0$ sedemikian sehingga $T(n) \leq C.\ f(n)$
    - Ingat : $C$ dan $n_0$ tidak unik

# Contoh :

- Tunjukkkan bahwa $T(n) = 2n^2 + 6n + 1 = O(n^2)$!
- Penyelesaian 1:

  Jika $n \geq 1$ maka $n \leq n^2$ dan $1 \leq n^2$ Sehingga

  $2n^2 + 6n + 1 \leq 2n^2 + 6n^2 + n^2 = 9n^2$ untuk semua $n \geq 1$

  Jadi dengan $C = 9$ dan $n_0 = 1$ terlihat bahwa

  $T(n) = 2n^2 + 6n + 1 = O(n^2)$

# Lanj.Contoh

- Penyelesaian 2

  $6n \leq n^2$ untuk $n \geq 6$ Sehingga

  $2n^2 + 6n + 1 \leq n^2 + n^2 + n^2 = 3n^2$ untuk semua $n \geq 6$

  Jadi dengan $C = 3$ dan $n_0 = 6$ terlihat bahwa

  $T(n) = 2n^2 + 6n + 1 = O(n^2)$

# Lanj. Contoh

- Penyelesaian 3

  Untuk C berapa jika $n_0 = 2$?

  Dengan kata lain :

  Berapa nilai C sdh terpenuhi :

  $$2n^2 + 6n + 1 \le Cn^2 \text{ untuk semua } n \ge 2$$

# Latihan

- Tunjukkan bahwa $T(n) = 5 = O(1)$!

- Penyelesaian :

  $5 = O(1)$ karena $5 \leq 6.1$ untuk $n \geq 1$

  $5 = O(1)$ karena $5 \leq 10.1$ untuk $n \geq 1$

# SOAL

- Tunjukkan bahwa $T(n) = 3n+2 = O(n)$!
- Tunjukkan bahwa kompleksitas waktu algoritma selection sort adalah $T(n) = \frac{1}{2} n (n-1) = O(n^2)$!
- Tunjukkan bahwa $T(n) = 5n^2 = O(n^3)$ dan $T(n) = n^3 \neq O(n^2)$!

# Teorema

- Bila $T(n) = a_m n^m + a_{m-1} n^{m-1} + \ldots + a_1 n + a_0$ adalah polinom berderajat m maka $T(n) = O(n^m)$.

- Bukti :

$T(n) \;= a_m n^m + a_{m-1} n^{m-1} + \ldots + a_1 n + a_0$

$$= \sum_{i=0}^{m} a_i n^i$$

$$\leq n^m \sum_{i=0}^{m} a_i \quad = C\, n^m \text{, untuk } n \geq 1$$

Jadi : $T(n) = O(n^m)$.

# Teorema Big O

- Misalkan $T_1(n) = O(f(n))$ dan $T_2(n) = O(g(n))$, maka:

  (a) (i) $T_1(n) + T_2(n) = O(\max(f(n),g(n))$

  (ii) $T_1(n) + T_2(n) = O(f(n)+g(n))$

  (b) $T_1(n)\ T_2(n) = O(f(n)\ O(g(n)) = O(f(n)\ g(n))$

  (c) $O(cf(n)) = O(f(n))$, c adalah konstanta

  (d) $f(n) = O(f(n))$

# Contoh

- Misalkan $T_1(n) = O(n)$, $T_2(n) = O(n^2)$ dan $T_3(n) = O(mn)$, dengan m adalah peubah, maka :
  - $T_1(n) + T_2(n) = O(max(n,n^2) = O(n^2)$
  - $T_2(n) + T_3(n) = O(n^2+mn)$
  - $T_1(n)\ T_2(n) = O(n.n^2) = O(n^3)$

- $O(5n^2) = O(n^2)$
- $n^2= O(n^2)$

# Aturan menentukan kompleksias waktu asimtotik

- Ditentukan dengan mengambil suku yang mendominasi fungsi T dan menghilangkan keofisiennya
  - Misal :
  - Algoritma Cari Elemen Terbesar : $T(n) = n-1 = O(n)$
  - Algoritma Pencarian Linier

    $T_{min}(n) = 1 = O(1)$

    $T_{max}(n) = n = O(n)$

    $T_{average}(n) = ½ (n + 1) = O(n)$

- Menghitung kompleksitas waktu terburuk saja

# Notasi Big $\Omega$

- Mendefinisikan batas bawah (lower bound) dari suatu kompleksitas waktu

- Definisi :

  $T(n) = \Omega(g(n))$ (artinya $T(n)$ berorde paling kecil $g(n)$) bila terdapat konstanta $C$ dan $n_0$ sedemikian sehingga $T(n) \geq C.\ f(n)$, untuk $n \geq n_0$

# Notasi Big $\Theta$

- Definisi

  $T(n) = \Theta(h(n))$ (artinya $T(n)$ berorde sama dengan $h(n)$) jika

  $T(n) = O(h(n))$ dan $T(n) = \Omega(h(n))$

# Contoh

- Tentukan notasi $\Omega$ dan $\Theta$ untuk $T(n) = 2n^2 + 6n + 1$!

- Penyelesaian :

  Karena $2n^2 + 6n + 1 \geq 2n^2$ untuk semua $n \geq 1$ maka

  $T(n) = 2n^2 + 6n + 1 = \Omega(n^2)$

- Karena $2n^2 + 6n + 1 = O(n^2)$ dan $2n^2 + 6n + 1 = \Omega(n^2)$ maka $2n^2 + 6n + 1 = \Theta(n^2)$

# Latihan

- Tentukan notasi O, Ω dan Θ untuk $T(n) = 5n^3 + 6n^2 \log n$

# Teorema

- Bila $T(n) = a_m n^m + a_{m-1} n^{m-1} + \ldots + a_1 n + a_0$ adalah polinom berderajat m maka T(n) adalah berode $n^m$.

- Contoh :

  Tentukan notasi O, Ω dan Θ untuk algoritma selection sort!

- Penyelesaian :

  $T(n) = ½\, n(n-1)$

  $= ½\, n^2 – ½\, n$

  Sesuai teorema maka algoritma ini berorde ....

  Jadi notasinya....

# KUIS

SELAMAT MENGERJAKAN