Nama : Anugerah Prima Bagaskara

NPM : 140810180006

Kelas : B

Tugas Pengganti Praktikum

Pertemuan 3

1. **Penjelasan *Worst Case* diutamakan untuk dihitung**
   a. Merupakan *upper bound* (nilai indeks terbesar) dari *running time* untuk *input* apa saja.
   b. *Worst case* sering terjadi. Contoh pada kasus pencarian
   c. *Average case = worst case,* karena fungsi kuadratik dari n.
2. **Big-O Notation**
   - Menghitung worst case, contoh : $T(n) = 2n^2 + 6n + 1$
     - Untuk n yang besar, pertumbuhan T(n) sebanding dengan n^2
     - Suku selain $2n^2$ bisa diabaikan karena tidak berarti, maka $T(n) = 2n^2$ + suku-suku lainnya
     - Koefisien 2 pada $2n^2$ boleh diabaikan → $T(n) = O(n^2)$ → Kompleksitas Asimtotik
   - Definisi dari big-O notation adalah sebagai berikut :
     - Misal, T(n) = O(f(n)) artinya T(n) berorde paling besar f(n) bila terdapat konstanta C dan n_0 Untuk $n >= n_0$,
     - T(n) <= C.f(n)
     - Jika n dibuat semakin besar, waktu yang dibutuhkan tidak akan melebihi konstanta C dikali f(n). Maka upper bound.
     - Digunakan untuk memperkirakan kompleksitas dengan mengabaikan suku berorde rendah.
   - Big-O Notation Polinomial Derajat n
     - Teorema 1 :
       - Contoh $T(n) = 3n^3 + 6n2 + n + 8 = O(n^3)$ dinyatakan pada :
       - Bila $T(n) = a_m n^m + a_{m-1} n^{m-1} + a_1 n + a_0$ adalah polinom berderajat m maka $T(n) = O(n^m)$

        - Mengambil suku paling tinggi derajatnya yang diartikan laju pertumbuhan lebih cepat dibandingkan yang lainnya.
    - Teorema 2 :
        - Misal $T_1(n)=O(f(n))$ dan $T_2(n) = O(g(n))$:
            - (i)$T_1(n)+T_2(n) = O(max(f(n),g(n))$
            - $T_1(n).T_2(n) = O(f(n))O(g(n)) = O(f(n).g(n))$
            - $O(cf(n)) = O(f(n))$, c adalah konstanta
            - $F(n) = O(f(n))$

3. **Aturan Kompleksitas Waktu Asimptotik**
    - Cara 1 :
        - Jika T(n) sudah dihitung, dapat ditentukan dengan mengambil suku yang mendominasi fungsi T& meninghilangkan koefisiennya(Teorema 1)
        - Contoh : cariMax, $T(n) = n-1 = O(n)$
    - Cara 2 :
        - Pengisian nilai, perbandingan, operasi aritmatik, read,write, dll. cara penghitungan :
        - $T(n)<=C. (g(n))$, dengan syarat nilai C dan N positif

4. **Big-Ω dan Big-Ө Notation**
    - Big-O hanya menyediakan upper bound, sedangkan lower bound, dapat diperoleh dengan **Big-Ω dan Big-Ө**
        - Big-Ω Notation:
            - $T(n) = \Omega(f(n))$, artinya T(n) berorde paling kecil f(n) bila terdapat konstanta C dan n0 sehingga :
            - $T(n) > C . (f(n))$, dengan syarat nilai c dan n positif

        - Big-Θ Notation:
            - $T(n) = \Theta(h(n))$, artinya T(n) berorde sama dengan h(n) Jika $T(n) = O(h(n))$ dan
            - $T(n) = \Omega (g(n))$.
            - $C1 f(n) < T(n) < C2 f(n)$, dengan syarat nilai c dan n positif

Contoh Soal :

1. Tunjukkan bahwa T(n) = 5 = O(1)

   Penyelesaian :

   a. 5 = O(1) karena 5<=6. 1 untuk n>=1.

      (C=6 dan n_0=1)

      5 = O(1) karena 5 <=10 . 1 untuk n>=1

2. Membuat_baris_fibonacci

Kamus:
 a, b, c, i, n : integer

Algoritma:
 a <-- 0
 b <-- 0
 c <-- 0

 input(n)

 for i <-- 1 to n do
  output(c)
  c = a+b
  b = a
  a = c

 endfor

$T_{min}(n) = 4$
$T_{max}(n) = 4n$
$T_{avg}(n) = (4 + 4n)/2$
$= n/2$
$= n$

Big Oh : t(n)≤ cg(n)

Tmin(n) : $4 \leq 4$

Tmax(n) : $4n \leq n.n$ (untuk semua $n \geq 5$)
$4n \leq n^2$
$20 \leq 25$
c = 20, n0 = 5

Tavg(n) : $4+4n \leq n+4n$ (untuk semua $n \geq 7$)
$4+4n \leq 5n$
$32 \leq 35$

c= 32, n0 = 7

Big Omega : $t(n) \geq cg(n)$

Tmin : $4 \geq 4$

Tmax : $4n \geq n.n$ (untuk semua $n \geq 1$)
$4n \geq n^2$
c = 4, n0 = 1

Tavg(n) : $4+4n \geq n+4n$ (untuk semua $n \geq 1$)
$4+4n \geq 5n$
$8 \geq 5$
c = 8, n0 = 5

Big Theta : $c_2g(n) \leq t(n) \leq c_1g(n)$

Tmin : $4 > 0$

Tmax : $C_1g(n) = 5$
$C_2g(n) = 1$
$C_2g(n) \leq 4n \leq C_1g(n)$ (untuk semua $n \geq 4$)
$C_1 = 5$, $C_2 = 1$, n0 = 4

Tavg : $C_1g(n) = 4$
$C_2g(n) = 0$
$C_2g(n) \leq 4+4n \leq C_1g(n)$ (untuk semua $n \geq 5$)
$C_1 = 0$,$C_2 = 4$, n0 = 5

3. **Procedure faktorial (input a1..an : integer); à fak : real;**

```
Kamus :
    i : integer;
    fak : real;

Algoritma :
    Input(n);
    If (n = 0) or (n = 1) then
        faktorial ß 1;
    Else
        fak ß 1;
        for I ß 2 to n do
            fak ß fak * I;
        endfor;
    endif;
    output(fak);

endprocedure;
```

Procedure ini mempunyai $T(n) = n + 5$
Maka
Big O untuk semua $n \geq 1$
$T(n) \leq C (g(n))$
$n + 5 \leq n + 5 + n$
$1 + 5 \leq 1 + 5 + 1$
$6 \leq 7$
$C = 7$, $n = 1$

Big Ω untuk semua $n \geq 1$
$T(n) \geq C (g(n))$
$5 + n \geq 0 (n)$
$5 + 1 \geq 0 (1)$
$6 \geq 0$
$C = 0$, $n = 1$

Big Θ untuk semua $n \geq n(0)$
$C_2 g(n) \leq t(n) \leq C_1 g(n)$

Batas Atas :
$T(n) \leq C (g(n))$
$5 + n \leq 5 + n + n$
$5 + 1 \leq 5 + 1 + 1$
$6 \leq 7$
$C = 7$, $n = 1$

Batas Bawah :
$T(n) \geq C (g(n))$
$5 + n \geq 0 (n)$
$5 + 1 \geq 0 (1)$
$6 \geq 0$
$C = 0$, $n = 1$

Jadi, $\Omega(g(n)) \leq 5 + n \leq O(g(n))$