20 Nov 2007

NOTASI ASIMTOTIK

Notasi asimtotik merupakan himpunan fungsi yang dibatasi oleh suatu fungsi $n \in \mathbf{N}$ yang cukup besar.

Fungsi : $\mathbf{N} \rightarrow \mathbf{R}$ (sering $\mathbf{R}^+$)

Ada tiga notasi :
O (big – Oh) untuk batas atas.
Ω (omega) untuk batas bawah
θ (teta) untuk ekuivalen

Masing-masing mempunyai parameter berupa fungsi.

$1000\ n^2 \leq n^3$; untuk $n \geq 1000$

1. Notasi Big O (big – Oh)

Misalkan :

$$g : \mathbf{N} \rightarrow \mathbf{R}^+$$

$$O(g(n)) = \{f(n)/(\exists c \in \mathbf{R}^+)\ (\exists\ n \in \mathbf{N}) \ni |f(n)| \leq c\ g\ (n),\ n \geq \mathbf{N})$$

g(n) sebagai batas atas untuk semua fungsi dalam O(g(n))

Contoh :

- $1000\ n^2 \in O(n^3)$ karena

  $1000\ n^2 \leq 1 \times n^3$ untuk $n \geq 1000$ $1 = c$, $1000 = \mathbf{N}$

- $1000\ n^2 \in O(n^2)$
  Carilah c dan n
  $1000\ n^2 \in O(n^2)$
  $1000\ n^2 \leq c\ n^2$
  $c = 1000$
  $1000\ n^2 \leq 1000\ n^2$, $n \geq 1$

- Diketahui $5000\ n^2 + 10000\ n + 10^6 \in O(n^2)$
  Carilah c dan n

  $5000\ n^2 + 10000\ n + 10^6 \leq c.n^2$
  ambil nilai sebarang untuk c, misalnya $c=10^7$
  $5000\ n^2 + 10000\ n + 10^6 \leq 10^7\ n^2$, untuk $n \geq 1$

- Apakah $5n + 10 \in O(n^2)$?
  Ya, karena $5n + 10 < 5n^2 + 10n^2 = 15n^2$ untuk $n > 1$
  Jadi untuk $c = 15$, $n_0 = 1$ $|5n + 10| < c \cdot |n^2|$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = L$,

Jika L = 0, maka $f(n) \in O(g(n))$
$g(n) \notin O(f(n))$

Jika $L \neq 0$, maka $f(n) \in O(g(n))$
$g(n) \in O(f(n))$

Jika $L = \pm\infty$, maka $f(n) \notin O(g(n))$
$g(n) \in O(f(n))$

Contoh :

a. $\lim_{n\to\infty} \frac{1000n^2}{n^3} = \lim_{n\to\infty} \frac{1000}{n} = 0$

Berarti
$1000\,n^2 \in O(n^3)$
$n^3 \notin O(1000\,n^2)$

Dalam limit berlaku
$0/a = 0$; $a/0 = \infty$;
$\infty/a = \infty$; $a/\infty = 0$;
$\infty \pm a = \infty$

b. $5n + 10 \ln n \in O(\ln n)$

$$\lim_{n\to\infty} \frac{5n + 10\ln n}{\ln n} = \lim_{n\to\infty} \frac{5n}{\ln n} + \frac{10\ln n}{\ln n}$$
$$= \lim_{n\to\infty} \frac{5n}{\ln n} + 10$$
$$= \lim_{n\to\infty} \frac{5}{\frac{1}{n}} + 10$$

Menggunakan *Teorema D'Hospital*

$$= \lim_{n\to\infty} 5n + 10$$
$$= \infty + 10$$
$$= \infty$$

$5n + 10 \ln n \notin O(\ln n)$
$\text{Ln}\, n \in O(5n + 10 \ln n)$

c. ln n ∈ O (5n + 10 ln n)

$$\lim_{n\to\infty} \frac{\ln n}{5n + 10\ln n} \quad = \lim_{n\to\infty} \frac{\frac{1}{n}}{5 + 10\frac{1}{n}}$$ Menggunakan *Teorema D'Hospital*

$$= \frac{0}{5+0}$$
$$= \frac{0}{5}$$
$$= 0$$

ln n ∈ O (5n + 10 ln n)
5n + 10 ln n ∉ ln n

d. (n +1)! ∈ O (n!)

$$\lim_{n\to\infty} \frac{(n+1)!}{n!} = \lim_{n\to\infty} \frac{(n+1).n!}{n!}$$
$$= \lim_{n\to\infty} n+1$$
$$= \infty + 1$$
$$= \infty$$

(n + 1) ! ∉ O (n!)
n ! ∈ O ((n+1)!)

e. Buktikan $n^2 \in O(2^n)$
Bukti :

$n^2 \in O(2^n)$ jika $\lim_{n\to\infty} \frac{n^2}{2^n} < \infty$

$$\lim_{n\to\infty} \frac{n^2}{2^n} = \lim_{n\to\infty} \frac{2n}{\log 2.\ 2^n}$$ Teo D' Hospital

$$= \lim_{n\to\infty} \frac{2}{\log 2.\log 2.\ 2^n}$$ Teo D' Hospital

$$= \frac{2}{\infty} = 0$$

f. Buktikan $n! \notin O(2^n)$
Bukti :

$n! \notin O(2^n)$ jika $\lim_{n\to\infty} \frac{n!}{2^n} = \pm\infty$

$$\frac{n!}{2^n} = \frac{n}{2} \cdot \frac{n-1}{2} \cdot \frac{n-2}{2} \cdots \frac{3}{2} \cdot \frac{2}{2} \cdot \frac{1}{2} > \frac{n}{2} \cdot 1 \cdots 1 \cdot \frac{1}{2} = \frac{n}{4}$$

$$\lim_{n\to\infty} \frac{n}{4} = \frac{\infty}{4} = \infty$$

Karena $\frac{n!}{2^n} > \frac{n}{4}$ maka $\lim_{n\to\infty} \frac{n!}{2^n} > \lim_{n\to\infty} \frac{n}{4}$

Berarti $\lim_{n\to\infty} \frac{n!}{2^n} = \infty$.

$O(1) \subset O(\log n) \subset O(n) \subset O(n \log n) \subset O(n^2) \subset O(n^3) \subset O(2^n) \subset O(e^n)$

2. Notasi Ω (Omega)

Misalkan :

$$g : \mathbb{N} \to \mathbb{R}^+$$

$\Omega(g(n)) = \{f(n) / (\exists c \in \mathbb{R}^+)(\exists n \in \mathbb{N}) \ni |f(n)| \geq c \cdot g(n), n \geq \mathbb{N})$

Contoh :

a. $n^3 \geq 1000\, n^2$ untuk $n \geq 1000$
$n^3 \in \Omega(1000\, n^2)$

b. $n^3 \geq n^2$, $n \geq 1$
$n^3 \in \Omega(n^2)$

c. $(n + 1)! = (n + 1)\, n! \geq n!$ untuk $n \geq 1$
$(n + 1)! \in \Omega(n!)$

d. $5000\, n^2 + 10000\, n + 10^6 \geq n^2$, untuk $n \geq 1$
$5000\, n^2 + 10000\, n + 10^6 \in \Omega(n^2)$
$5000\, n^2 + 10000\, n + 10^6 \in O(n^2)$
$5000\, n^2 + 10000\, n + 10^6 \in O(n^2) \cap \Omega(n^2)$ $O(n^2) \cap \Omega(n^2) = \theta(n^2)$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = L$,

	Jika L = 0, maka	$f(n) \notin \Omega(g(n))$
		$g(n) \in \Omega(f(n))$
	Jika $L \neq 0$, maka	$f(n) \in \Omega(g(n))$
		$g(n) \in \Omega(f(n))$
	Jika $L = \pm\infty$, maka	$f(n) \in \Omega(g(n))$
		$g(n) \notin \Omega(f(n))$

$50\ n + 10 \ln n \in \Omega(\ln n)$

$n^2 \notin \Omega(n^3)$

3. Notasi θ (Teta)

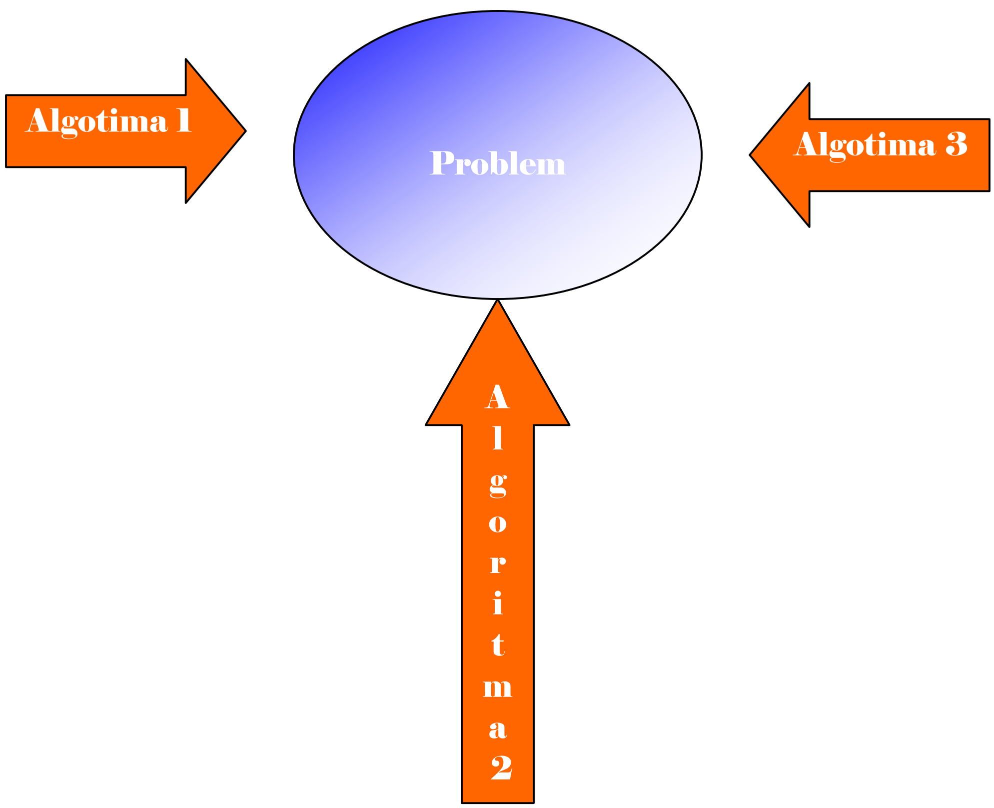
$$f(n) \in \theta(g(n)) \text{ bila dan hanya bila } f(n) \in O(g(n) \cap \Omega(g(n)))$$

f(n) mempunyai order yang sama dengan g(n)

$f(n) \in \theta(g(n)$ bila dan hanya bila $g(n) \in \theta(f(n))$ f(n) berupa fungsi non rekursif

Notasi Asimtotik digunakan untuk menentukan kompleksitas suatu algoritma dengan melihat waktu tempuh algoritma. Waktu tempuh algoritma merupakan fungsi : $\mathbf{N} \rightarrow \mathbf{R}^+$

27 November 2007



Analisis algoritma dengan waktu tempuh meliputi *Space* dan *banyak langkah*. Space sering berubah pada saat runtime seperti :

- ♣ pointer : link list. Integer 8 byte (compile) runtime berubah menjadi …kb.
- ♣ Pointer, memo/ text space nya tidak dapat ditentukn pada compile time.

♣ Stack untuk bentuk rekursif.

Namun ada juga yang bisa ditentukan antara lain :
♣ Primitif (integer, boolean, real, char, byte)
♣ Record yang tidak memuat pointer array dan matriks

Oleh karena space sering berubah tersebut, terkadang space tidak dilibatkan dalam waktu tempuh. Yang berperah dalam waktu adalah banyak langkah.

+ , - , * , /   dianggap mempunyai waktu yang sama

Contoh :
x + y mempunyai waktu yang sama dengan x * y

x + y mempunyai waktu berbeda dengan x + y * z

Faktor-faktor yang menentukan banyak langkah antara lain :

1. Banyak operator dasar yang digunakan
   Contoh :
   y : x + z   banyak langkahnya 1 karena mempunyai 1 operator.

2. Assigment (konstanta c)
3. function Call :
   a. Reserved
   b. User Defined

4. Struktur Program :
   a. Sekuential
   b. Percabangan
   c. Kalang (loop)

```
FUNCTION Sinus (x) : real;
BEGIN
      Sinus : 0
      FOR i : = 0 TO 1000
      IF i mod 2 = 0 THEN d:= 1
      ELSE
            d:= - 1
      jum:=jum + d * exp ((2 * i + 1) * ln (x)) / fakt (2 * i + 1)
      sinus : = jum
END
```

Waktu tempuh = space + banyak langkah

SEKUENTIAL

Misalkan dalam algoritma terdapat blok statement, masing -masing mempunyai banyak langkah :

S_1 banyak langkah P_1
S_2 banyak langkah P_2
S_3 banyak langkah P_3
. .
. .
. .
S_n banyak langkah P_n

Total banyak langkah blok-blok statement tersebut $\sum_{i=1}^{n} P_i$

S_i bisa berupa : assigment, procedure call, percentage, kalang.

Contoh :

x ← x * y	operasi 1	= 1
y ← a * sin (x)	operasi 1, procedure 1	= 2
readln (b)	assigment 1	= 1
writeln (x + y + b)	assigment 1, operasi 2	= 3 +
	Banyak Langkah	= 7

PENCABANGAN

Bentuk IF k THEN S_1
ELSE S_2

k = kondisi dengan banyak langkah c
S_1, S_2 = blok statement dengan banyak langkah P_1, P_2

Kasus terbaik mempunyai banyak langkah
c + min (P_1, P_2)

Kasus terburuk mempunyai banyak langkah
c + max (P_1, P_2)

Yang digunakan dalam menentukan banyak langkah dalam suatu pencabangan adalah kasus terburuk.

Operator dasar logika : AND, OR, NOT dihitung 1 langkah

Contoh :

Not (P AND Q) mempunyai langkah sebanyak 2

Not (x > 0 AND y > 0) mempunyai langkah sebanyak 4

$C\ n^k \in \theta\ (n^k)$ C = kombinasi

$$\lim_{n\to\infty} \frac{Cn^k}{n^k} = C$$

$Cn^k \in O\ (n^k) \cap \Omega\ (n^k)$

$$\left.\begin{array}{l}\text{IF } x > 0 \text{ THEN } x := x - 1 \\ \qquad\qquad\qquad\quad y := x + y\end{array}\right\} \quad c + 2$$

ELSE

$$\left. y := x - y \right\} \quad c + 1$$

c = 1

Banyak langkah kondisi I adalah 2

Banyak langkah kondisi II adalah 1

Kasus terjelek adalah $c + \max(P_1, P_2) = 1 + 2 = 3$

Dengan demikian banyak langkah untuk pencabangan diatas dihitung 3.

4 Des 2007

LOOP

While/ Repeat tidak mudah untuk dianalisis karena banyak langkah tidak pasti. Yang paling mungkin dianalisis adalah For loop.

Bentuk Umum For Loop

FOR *variabel nilai awal* TO *nilai akhir* STEP *S*

S var

Dalam bahasa C, FOR (var = awal, var = akhir, var = +step)

Dalam Matlab, FOR var = awal : step : akhir

Tipe Counter : integer dalam Bahasa Visual Basic

Real dalam Bahasa C dan Matlab

Step : Integer dalam Bahasa Visual Basic

Real dalam Bahasa C dan Matlab

Contoh dalam matlab FOR n = 0,5 : 0,3 : 7,1

N	Step Ke
0,5	1
0,8	2
1,1	3
. . .	. . .
7,1	23

Banyak Langkah untuk Statement FOR

Kasus I:

Counter : integer

Step : 1

Statement S mempunyai banyak langkah yang tidak bergantung nilai counter

FOR *counter : awal* TO *akhir*
S

S dieksekusi sebanyak <u>*akhir – awal +1*</u> kali

Hidden : Counter ≤ Akhir   S dieksekusi sebanyak <u>*akhir – awal + 2*</u> kali
Counter = counter + 1   S dieksekusi sebanyak <u>*akhir – awal + 1*</u> kali

Banyak Langkah = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)

p = banyak langkah statement.

Contoh :

Berapa banyak langkah dari
FOR i = 1 TO n
x : = x + 5
y : = y + x

Penyelesaian :

Banyak langkahnya = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)
= (n – 1 + 2) + (n – 1 + 1) (2 + 1)
= (n + 1) + (n)(3)
= n + 1 + 3n
= 4n + 1

Kasus II : seperti kasus I tetapi mempunyai STEP = s

s dieksekusi sebanyak $\left\lfloor \frac{akhir - awal}{s} + 1 \right\rfloor$[i] atau ((akhir – awal) **div** s + 1)

Contoh :

Berapa banyak langkah dari

```
FOR i:= j TO n STEP 3
      x := x + i
      y := y + j
```

Penyelesaian :

$$\text{Banyak langkahnya} = \left\lfloor \frac{akhir - awal}{s} + 2 \right\rfloor + \left\lfloor \frac{akhir - awal}{s} + 1 \right\rfloor (p + 1)$$

$$= \left(\frac{n-j}{3} + 2\right) + \left(\frac{n-j}{3} + 1\right)(2+1)$$

$$= \left(\frac{n-j}{3} + 2\right) + \left(\frac{n-j}{3} + 1\right)(3)$$

$$= \left(\frac{n-j}{3} + 2\right) + \left(3.\frac{n-j}{3} + 3\right)$$

$$= \frac{n-j}{3} + 2 + 3.\frac{n-j}{3} + 3$$

$$= 4\frac{n-j}{3} + 5$$

$4\frac{n-j}{3} + 5 \in O(n)$ $\qquad$ $P_d(n) \in O(n^d)$ $\qquad$ P = Polinomial

d = Derajat

Contoh :

Berapa banyak langkah dari

```
FOR i = 0,5 TO 7,1 STEP 0,3
      x := x + i
      y := y + j
```

Penyelesaian :

$$\text{Banyak langkahnya} = \left\lfloor \frac{akhir - awal}{s} + 2 \right\rfloor + \left\lfloor \frac{akhir - awal}{s} + 1 \right\rfloor (p + 1)$$

$$= \left(\frac{7{,}1 - 0{,}5}{0{,}3} + 2\right) + \left(\frac{7{,}1 - 0{,}5}{0{,}3} + 1\right)(2+1)$$

[i] ⌊ X ⌋ berarti berlaku pembulatan kebawah terhadap X jika X bukan merupakan bilangan bulat

$$= \left(\frac{6{,}6}{0{,}3}+2\right)+\left(\frac{6{,}6}{0{,}3}+1\right)3$$

= (22 + 2) + (22 + 1) . 3
= 24 + 23. 3
= 24 + 69
= 93

Kasus III : banyak langkah S bergantung nilai Counter

Contoh :

```
FOR i = 1 TO n
        x : = x + y
        FOR j : = i TO n
                y : = i + j
```

Ket :

Outer Loop

S

Inner Loop

Inner Loop
Banyak langkah = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)
= ((n – i) + 2) + ((n – i) + 1) (1 + 1)
= ((n – i) + 2) + ((n – i) + 1) . 2
= ((n – i) + 2) + 2(n – i) + 2
= 3 (n – i) + 4
= 3n – 3i + 4

(P(i)) = Banyak Langkah dalam S = 1 + banyak langkah inner loop
= 1 + 3n – 3i + 4
= 3n – 3i + 5

banyak langkah
x : = x + y adalah 1

Outer Loop

Banyak langkah = (akhir – awal + 2) + (akhir – awal + 1) .1 + $\sum_{i=1}^{n} P(i)$

= ((n – 1) + 2) + ((n – 1) + 1) .1 + $\sum_{i=1}^{n}(3n-3i+5)$

= 2n + 1 + $\sum_{i=1}^{n} 3n$ - $\sum_{i=1}^{n} 3i$ + $\sum_{i=1}^{n} 5$

= 2n + 1 + 3n.n – 3. $\left(\frac{1}{2}n(n+1)\right)$ + 5.n

$$\sum_{i=1}^{n} i = \left(\frac{1}{2}n(n+1)\right)$$

= 2n + 1 + $3n^2$ - $\frac{3}{2}n^2 - \frac{3}{2}n$ +5n

= 4n + 2 + $6n^2$ – $3n^2$ - 3n + 10n

= $3n^2$ + 11 n + 2

$3n^2 + 11n + 1 \in O(n^2)$

```
FOR i : = awal TO akhir STEP s
        S(i)
```

Misal P(i) banyak langkah S(i) maka banyak langkah loop tersebut

$$2\left\lfloor \frac{akhir - awal}{s} \right\rfloor + 3 + \sum_{i=awal,s}^{akhir} P(i)$$

$$\sum_{i=awal,s}^{akhir} P(i) = \text{P.awal} + (\text{P.awal}+s) + (\text{p.awal}+2.s) + \ldots + (\text{p.akhir})$$

Contoh :
Hitung banyak langkah dari

```
FOR i : = 1 TO n
        x: = x + y
        FOR j:= 1 TO i
                y : = i + j
```

Outer Loop

S

Inner Loop

Penyelesaian :
<u>Inner Loop</u>

Alternatif 1
Banyak langkah = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)
= ((i – 1) + 2) + ((i – 1) + 1) (1 + 1)
= (i + 1) + 2i
= 3i + 1

Alternatif 2
Banyak langkah = 2 (akhir – awal) + 3 + p (akhir – awal + 1)
= 2 (i – 1) + 3 + 1. ((i – 1) + 1)
= 2i – 2 + 3 + i
= 3i + 1

Banyak langkah dalam S = 1 + banyak langkah inner loop
= 1 + 3i + 1
= 3i + 2

<u>Outer Loop</u>

$$\text{Banyak Langkah} = 2\,(akhir - awal) + 3 + \sum_{i=awal,s}^{akhir} P(i)$$

$$= 2\,(\text{n} - 1) + 3 + \sum_{i=1}^{n} 3i + 2$$

$$= 2\text{n} - 2 + 3 + \sum_{i=1}^{n} 3i + \sum_{i=1}^{n} 2$$

$$= 2\text{n} + 1 + 3\left(\frac{1}{2}n(n+1)\right) + 2.\,\text{n}$$

$$= 4\text{n} + 1 + \left(\frac{3}{2}n^2 + \frac{3}{2}n\right)$$

$$= 3\text{n}^2 + 11\text{n} + 2$$

$$3\text{n}^2 + 11\text{n} + 2 \in O(\text{n}^2)$$

Tugas !

Diketahui

```
Read (x)
y: = 0
FOR i:= 1 TO n
     x:= x + y
     FOR j:= i + 1 TO n^2
          y:= y + i + j
     write (x,y)
write (x,y)
```

Tentukan T(n) = **….** $\in$ O (**…**)

Penyelesaian :

```
Read (x)
y: = 0
FOR i:= 1 TO n
     x:= x + y                          …..(***)
     FOR j:= i + 1 TO n^2       …..(**)
          y:= y + i + j   …..(*)
     write (x,y)
write (x,y)
```

Banyak langkah (*) = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)

= $(n^2 - (i + 1) + 2) + (n^2 - (i + 1) + 1)\,(2 + 1)$

= $(n^2 - i - 1 + 2) + (n^2 - i - 1 + 1)\,3$

= $n^2 - i + 1 + 3n^2 - 3i$

= $4n^2 - 4i + 1$

Atau dengan alternatif lain :

Banyak langkah (*) $= 2(\text{akhir} - \text{awal}) + 3 + p\ (\text{akhir} - \text{awal} + 1)$ ………….[ii]
$= 2\ (n^2 - (i + 1)) + 3 + 2\ (n^2 - (i + 1) + 1)$
$= 2\ (n^2 - i - 1) + 3 + 2\ (n^2 - i\ )$
$= 2n^2 - 2i - 2 + 3 + 2n^2 - 2i$
$= 4n^2 - 4i + 1$

Banyak langkah (**) $= 2 + \text{Banyak langkah } (*)$
$= 2 + 4n^2 - 4i + 1$
$= 4n^2 - 4i + 3$

Banyak langkah (***) $= 2\ (\text{akhir} - \text{awal}) + 3 + \sum_{i=awal,s}^{akhir} P(i)$

$$= 2\ (n - 1) + 3 + \sum_{i=1}^{n} 4n^2 - 4i + 3$$

$$= 2n - 2 + 3 + \sum_{i=1}^{n} 4n^2 - \sum_{i=1}^{n} 4i + \sum_{i=1}^{n} 3$$

$$= 2n + 1 + 4n^2\ .\ n - 4.\left(\frac{1}{2} n(n+1)\right) + 3.\ n$$

$$= 4n^3 + 5n + 1 - 2n^2 - 2n$$

$$= 4n^3 - 2n^2 + 3n + 1$$

Banyak Langkah Program = T(n) $= 3 + \text{Banyak langkah } (***)$
$= 3 + 4n^3 - 2n^2 + 3n + 1$
$= 4n^3 - 2n^2 + 3n + 4$

$$4n^3 - 2n^2 + 3n + 4 \in O\ (n^3)$$

[ii] (akhir-awal + 2) + (akhir – awal + 1) (p + 1) = 2(akhir – awal) + 3 + p (akhir – awal + 1)

11 Des 2007

RECURSIF CALL

Misalkan fungsi f(n) berbentuk rekursif . Untuk memanggil fungsi tersebut juga melibatkan pemanggilan bentuk rekursifnya. Banyak langkah untuk pemanggilan fungsi tersebut :

1. Banyak langkah bentuk non rekursif.
2. Banyak langkah fungsi tersebut dalam bentuk non rekursif.

Contoh :

```
Fakt (n)
IF n = 0 THEN fakt : = 1                    --> Non Rekursif
ELSE
     Fakt := n * Fakt (n – 1)               --> Rekursif
```

Misalkan
T(n) waktu yang diperlukan untuk memanggil fakt (n)
Jika n = 0, T (0) = 2 (konstanta)
Jika $n \ge 1$, T(n) = T (n – 1) + 2

2 adalah banyak langkah
fakt : = n * fakt (n – 1)

Waktu tempuh untuk pemanggilan fakt (n)

$$T(n) = \begin{cases} 2 & , n = 0 \\ T(n-1) + 2 & , n \ge 1 \end{cases}$$

Persamaan Karakter Homogen

$$\frac{x^n = x^{n-1}}{x = 1} \quad : x^{n-1}$$

Persamaan Karakter Non Homogen

$2 = 1^n.(2.\, n^0) \rightarrow (x-1)^{0+1} = x - 1$

$T(n) = (c_1 + c_2.\, n) .\, 1^n$
$\quad\;\; = c_1 + c_2\, n$

$T(0) = c_1 + 0 = 2$
$\quad\quad\; c_1 = 2$

$T(1) = c_1 + c_2 .1 = T(n\text{-}1) + 2$
$\quad c_1 + c_2 = T(0) + 2$
$\quad 2 + c_2 = 2 + 2$
$\quad\quad c_2 = 2$

$T(n) = c_1 + c_2 n$
$= 2 + 2n$
$= 2n + 2$
$2n + 2 \in O(n)$

Latihan :

1. Tentukan waktu tempuh serta kompleksitas fragmen algoritma berikut :

```
x : = 1;
READLN (y);
FOR k:=1 TO 2 * n DO
BEGIN
      x:= k + y;
      IF x > 0 AND y > 0 THEN
           x:= x * y;
           READLN (y);
      ELSE
           FOR j:= 1 TO k DO y:= x + y;
END;
```

2. Tentukan waktu tempuh serta kompleksitas fragmen algoritma berikut :

```
PROCEDURE Testing (x : Real; n : Integer)
VAR     a : array [1:n] OF Integer
        j  : Integer;

PROCEDURE A
BEGIN
        FOR j:= 1 TO n DO a[j]:= x * j;
END;

BEGIN
     IF n = 1 OR n = 2 THEN x:= n * x
     ELSE
         BEGIN
             A;
             FOR j:= 1 TO n DO
             BEGIN
                  A[j]:= x + a[j];
                  Sum:= Sum + a[j];
             END;
             Testing (Sum, n – 1);
             READLN (y);
             IF Sum > y  THEN Testing (Sum – y, n – 2)
             ELSE
```

```
                Testing (y – Sum, n – 2);
            Testing (y, n – 2);
         End;
END
```

3. Tentukan waktu tempuh serta kompleksitas fragmen algoritma berikut:

```
FUNCTION Bla (n)
VAR     k, l       : Integer;
        a[1..n]    : Real;

BEGIN
     FOR k := 1 TO n DO a(k):= 1;
     IF n = 1 OR n = 2 THEN
        READ (l);
        IF l < 0 THEN
             l := l ^ 2;
             k := n + l;
        ELSE
             k := n * l;
        Bla := k;

     ELSE
         FOR k:= 1 TO n DO
             READ (l);
             a(k) := a(k) + 1;
         l := bla(n – 1);
         FOR k := 1 TO n do l:=l + a(k);
         l := l + Bla (n – 2);
         READ (k);
         IF k < 0 THEN k : = l ^ 2 + Bla (n – 2)
         ELSE
             k := l + 2 * Bla (n – 2)
         Bla := k + n;
END
```

4. Tentukan waktu tempuh minimal dan waktu tempuh maksimal algoritma berikut :

```
x : = 1;
READLN (y);
FOR k := 1 TO n DO
BEGIN
     x := i + Sin (y);
     IF x > 0 AND y > 0 THEN
        x := x * y;
        READLN (y);
        y := x + y;
        END;
```

```
        ELSE
          FOR j := 1 TO k DO
              y := x + y;
              x := y ^ SQRT (t)
          END;
    END;
    FOR i:= 1 TO n ^ 2 DO
      x := x + y + i ;
```

Pembahasan Latihan

Nomor 1

```
x : = 1;
READLN (y);
FOR k:=1 TO 2 * n DO
BEGIN
     x:= k + y;
     IF x > 0 AND y > 0 THEN
          x : = x * y;
          READLN (y);
     ELSE
          FOR j:= 1 TO k DO y : = x + y;
END;
```

Penyelesaian :

Untuk memudahkan pemahaman dan penyelesaian, algoritma diberi kotak berwarna, masing masing menyimbolkan :

Inner Loop	B	A	Outer Loop

Inner Loop

Banyak Langkah = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)
= ((k – 1) + 2) + ((k – 1) + 1).(1 + 1)
= (k + 1) + 2k
= 3k + 1

Banyak Langkah B = c + max (P_1, P_2)
= 3 + max (2 , 3k + 1)

Banyak Langkah A = 1 + Banyak Langkah B
= 1 + 3 + max (2, 3k + 1)
= 4 + max (2, 3k + 1)

Outer Loop

Banyak Langkah $= 2\ (\text{akhir} - \text{awal}) + 3 + \sum_{awal}^{akhir} P(i)$

$= 2\ ((2*n)-1) + 3 + \sum_{k=1}^{2n} (4 + \max(2, 3k+1))$

$= 2\ (2n - 1) + 3 + \sum_{k=1}^{2n} 4 + \sum_{k=1}^{2n} \max(2, 3k+1)$

$= 4n - 2 + 3 + 4.\ 2n + \max\left(\sum_{k=1}^{2n} 2\right), \left(\sum_{k=1}^{2n} 3k+1\right)$

$= 4n + 1 + 8n + \max\ (\ 2\ .2n,\ \sum_{k=1}^{2n} 3k + \sum_{k=1}^{2n} 1)$

$= 12\ n + 1 + \max\ (4n,\ 3\ .\ \left(\frac{1}{2} n(n+1)\right) + 1.2n)$

$= 12n + 1 + \max\ (4n,\ \frac{3}{2}n^2 + \frac{3}{2}n\ + 2n)$

$= 12n + 1 + \max\ (4n,\ \frac{3}{2}n^2 + \frac{7}{2}n\ )$

Banyak langkah Program $= 2 + 12n + 1 + \max\ (4n,\ \frac{3}{2}n^2 + \frac{7}{2}n\ )$

$= 12n + 3 + \max\ (4n,\ \frac{3}{2}n^2 + \frac{7}{2}n\ )$

$12n + 3 + \max\ (4n,\ \frac{3}{2}n^2 + \frac{7}{2}n\ ) \in O\ (n^2)$

Nomor 2

```
PROCEDURE Testing (x : Real; n : Integer)
VAR  a : array [1:n] OF Integer
     j : Integer;

PROCEDURE A
BEGIN
       FOR j:= 1 TO n DO a[j]:= x * j;
END;

BEGIN
     IF n = 1 OR n = 2 THEN x:=n * x                ......*
     ELSE
         BEGIN
             A;                                     ......**
             FOR j:= 1 TO n DO                      ......***
             BEGIN
                  A[j]:= x + a[j];
```

```
                    Sum:= Sum + a[j];
                END;
                Testing (Sum, n – 1);
                READLN (y);
                IF Sum > y  THEN Testing (Sum – y, n – 2)
                ELSE
                    Testing (y – Sum, n – 2);
                Testing (y, n – 2);
             End;
      END
```

Penyelesaian :

Banyak Langkah ** = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)
= (n – 1 + 2) + (n – 1 + 1) (1 + 1)
= (n + 1) + (n)(2)
= 3n + 1

Atau dengan menggunakan cara lain

Banyak Langkah ** = 2 (akhir – awal) + 3 + p (akhir – awal + 1) ….[iii]
= 2 (n – 1) + 3 + 1 (n – 1 + 1)
= 2n – 2 + 3 + n
= 3n + 1

Banyak langkah *** = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)
= (n – 1 + 2) + (n – 1 + 1) (2 + 1)
= n + 1 + 3n
= 4n + 1

Lanjutkan sampai selesai.. he..he…

[iii] (akhir-awal + 2) + (akhir – awal + 1) (p + 1) = 2(akhir – awal) + 3 + p (akhir – awal + 1)

Nomor 3

```
FUNCTION Bla (n)
VAR      k, l         : Integer;
         a[1..n]      : Real;

BEGIN
      FOR k := 1 TO n DO a(k):= 1;    …(*)
      IF n = 1 OR n = 2 THEN
         READ (l);
         IF l < 0 THEN
               l := l ^ 2;
               k := n + l;            …(**)   …(***)
         ELSE
               k := n * l;
         Bla := k;

      ELSE
          FOR k:= 1 TO n DO
               READ (l);                 …(****)
               a(k) := a(k) + 1;
          l := bla(n – 1);               …(*****)
          FOR k := 1 TO n do l:=l + a(k);    …(******)
          l := l + Bla (n – 2);          …(#)                  ($)
          READ (k);
          IF k < 0 THEN k : = l ^ 2 + Bla (n – 2)
          ELSE                                    …(##)
               k := l + 2 * Bla (n – 2)
          Bla := k + n;
END
```

Penyelesaian :

Banyak langkah (*) = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)
= (n – 1 + 2) + (n – 1 + 1) (1 + 1)
= (n + 1) + 2n
= 3n + 1

Banyak langkah (**) = 1 + max (2, 1)
= 1 + 2
= 3

Banyak langkah (***) = 2 + banyak langkah (**)
= 2 + 3
= 5

Banyak langkah (****) = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)

$= (n - 1 + 2) + (n - 1 + 1)\ (2 + 1)$
$= (n + 1) + 3n$
$= 4n + 1$

Banyak langkah (*****) $= T\ (n - 1) + 1$

Banyak langkah (******) $= (akhir - awal + 2) + (akhir - awal + 1)\ (p + 1)$
$= (n - 1 + 2) + (n - 1 + 1)\ (1 + 1)$
$= (n + 1) + 2n$
$= 3n + 1$

Banyak langkah (#) $= T\ (n - 2) + 2$

Banyak langkah (##) $= 1 + \max (\ T(n - 2) + 3, T(n - 2) + 3)$
$= 1 + T(n - 2) + 3$
$= T\ (n - 2) + 4$

Banyak langkah ($) = 2 + Banyak langkah (****) + banyak langkah (*****) + banyak langkah (******) + banyak langkah (#) + banyak langkah (##)
$= 2 + (4n + 1) + (\ T(n - 1) + 1) + (3n + 1) + (T\ (n - 2) + 2) + T(n - 2) + 4$
$= 2 + 4n + 1 + T(n - 1) + 1 + 3n + 1 + T(n - 2) + 2 + T(n - 2) + 4$
$= T(n - 1) + 2\ T(n - 2) + 7n + 11$

Misalkan program yang didalam kotak adalah @ maka banyak langkah @ adalah

$$T(n) = \begin{cases} 8 & n = 1,\ 2 \\ T(n-1) + 2T(n-2) + 7n + 14 & n > 2 \end{cases}$$

Bentuk Karaketeristik Homogen

$\underline{x^n = x^{n-1} + 2\ x^{n-2}}\ : x^{n-2}$

$x^2 = x + 2$
$x^2 - x - 2 = 0$
$(x + 1)\ (x - 2) = 0$
$x = -1\ ,\ x = 2$

Bentuk Karakteristik Non Homogen

$7n + 14 = 1^n\ (7n + 14)$

$(\ x - 1)^{1+1} = (\ x - 1)^2$

$T(n) = c_1\ 2^n + c_2\ (-1)^n + (c_3 + c_4\ n)\ 1^n$

$= c_1 2^n + c_2 (-1)^n + c_3 + c_4 n$

$T(1) = c_1 2^1 + c_2 (-1)^1 + c_3 + c_4 .1 = 8$
$2c_1 - c_2 + c_3 + c_4 = 8$

$T(2) = c_1 2^2 + c_2 (-1)^2 + c_3 + c_4 . 2 = 8$
$4 c_1 + c_2 + c_3 + 2 c_4 = 8$

$T(3) = c_1 2^3 + c_2 (-1)^3 + c_3 + c_4 . 3 = T (n - 1) + 2 T (n - 2) + 7n + 11$
$= 8 c_1 - c_2 + c_3 + 3 c_4 = T(2) + 2 T(1) + 7.3 + 11$
$8 c_1 - c_2 + c_3 + 3 c_4 = 8 + 2 . 8 + 7.3 + 11$
$8 c_1 - c_2 + c_3 + 3 c_4 = 8 + 16 + 21 + 11$
$8 c_1 - c_2 + c_3 + 3 c_4 = 56$

$T(4) = c_1 2^4 + c_2 (-1)^4 + c_3 + c_4 . 4 = T (n - 1) + 2 T (n - 2) + 7n + 11$
$= 16 c_1 + c_2 + c_3 + 4 c_4 = T(3) + 2 T(2) + 7.4 + 11$
$16 c_1 + c_2 + c_3 + 4 c_4 = 56 + 2 . 8 + 7.4 + 11$
$16 c_1 + c_2 + c_3 + 4 c_4 = 56 + 16 + 28 + 11$
$16 c_1 + c_2 + c_3 + 4 c_4 = 111$

$$\begin{bmatrix} 2 & -1 & 1 & 1 & | & 5 \\ 4 & 1 & 1 & 2 & | & 5 \\ 8 & -1 & 1 & 3 & | & 56 \\ 16 & 1 & 1 & 4 & | & 111 \end{bmatrix}$$

$2 c_1 = -135$

Lanjutkan sampai selesai.. he…he…

Nomor 4

```
x : = 1;
READLN (y);
FOR k := 1 TO n DO
BEGIN
     x := i + Sin (y);
     IF x > 0 AND y > 0 THEN
       x := x * y;
       READLN (y);
       y := x + y;
       END;                                           ...(**)    ...(***)    ...(****)
     ELSE
       FOR j := 1 TO k DO
            y := x + y;                      ...(*)
            x := y ^ SQRT (t)
       END;
END;
FOR i:= 1 TO n ^ 2 DO               ...(*****)
  x := x + y + i ;
```

Penyelesaian :

Banyak langkah (*) = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)
= (k – 1 + 2) + (k – 1 + 1) (3 + 1)
= k + 1 + 4k
= 5k + 1

Banyak langkah (**) = 3 + max (3, 5k + 1)

Banyak langkah (***) = 2 + banyak langkah (**)
= 2 + 3 + max (3, 5k + 1)
= 5 + max (3, 5k + 1)

Banyak langkah Outer Loop (****) $= 2\ (\text{akhir} - \text{awal}) + 3 + \sum_{awal}^{akhir} P(i)$

$$= 2\ (n - 1) + 3 + \sum_{k=1}^{n} (5 + \max(3, 5k+1))$$

$$= 2n + 1 + \sum_{k=1}^{n} 5 + \sum_{k=1}^{n} \max(3, 5k+1)$$

$$= 2n + 1 + 5n + \max\left(\sum_{k=1}^{n} 3, \sum_{k=1}^{n} 5k + 1\right)$$

$$= 7n + 1 + \max\ (3n,\ 5\sum_{k=1}^{n} k + \sum_{k=1}^{n} 1)$$

$$= 7n + 1 + \max\ (3n,\ 5.\left(\frac{1}{2} n(n+1)\right) + n)$$

$$= 7n + 1 + \max(3n, \frac{5}{2}n^2 + \frac{7}{2}n)$$

Banyak langkah (*****) = (akhir – awal + 2) + (akhir – awal + 1) (p + 1)

$= (n^2 - 1 + 2) + (n^2 - 1 + 1)(2+1)$

$= n^2 + 1 + 3n^2$

$= 4n^2 + 1$

Banyak langkah Program = 2 + banyak langkah (****) + banyak langkah (*****)

$$= 2 + (7n + 1 + \max(3n, \frac{5}{2}n^2 + \frac{7}{2}n)) + 4n^2 + 1$$

$$= 4n^2 + 7n + 4 + \max(3n, \frac{5}{2}n^2 + \frac{7}{2}n)$$

$$4n^2 + 7n + 4 + \max(3n, \frac{5}{2}n^2 + \frac{7}{2}n) \in O(n^2)$$

REFERENSI

1. Drs. Retantyo Wardoyo, M.Sc, Ph.D dalam Perkuliahan Analisis Algoritma di Magister Ilmu Komputer UGM Semester I Tahun Ajaran 2007/ 2008.
2. Gilles Brassad, Paul Bratley, 1996. Fundamentals Of Algorithmics. Prentice Hall.
3. Penulis.

BIOGRAFI PENULIS

Hedri Wahyudi lahir di Padang, 14 Maret 1981. Menempuh pendidikan Formal dari SD hingga tingkat SMP di daerah 1000 Parit, yaitu di Kecamatan Batang Tuaka Kabupaten Indragiri Hilir (Inhil) Riau. Kemudian tahun 1994 melanjutkan Pendidikan di SMPN 2 Teluk Kuantan. Pendidikan SLTA ditempuh di SMUN 1 Kuantan Tengah. Kemudian tahun 1998 mengikuti PTT-1 Widyaloka Cab. Pekanbaru. Pada tahun 1999 diterima di Program Studi Pendidikan Matematika FKIP Universitas Riau melalui jalur UMPTN. Pernah mengajar di SMAN 8 Pekanbaru dan saat ini kuliah di Program Magister Ilmu Komputer Universitas Gadjah Mada.

Informasi lebih lanjut dapat dilihat di
http://hedri_w.web.ugm.ac.id(underconstructed)
http://hedriwahyudi.blogspot.com,
http://hedriwahyudi.wordpress.com,
http://www.friendster.com/youdee14, atau
email :
youdee_12@yahoo.com,
hedri_w@mail.ugm.ac.id,
hedri.wahyudi@gmail.com