Studi Kasus : Analisis Algoritma

Sukmawati Nur Endah

Departemen Informatika UNDIP

Latihan (PR sebelumnya)

- Buatlah algoritma Fibbonanci!
- Carilah kompleksitas dari Algoritma Fibbonanci!

Solusi : Alg Fibonanci

```
function Fibonacci (input n : integer) → integer;

Algoritma :
if (n=0) then
          Fibonacci ← 0
else if  (n = 1) then
                Fibonacci ← 1
        else
                 Fibonacci ← Fibonacci  (n-1) + Fibonacci  (n-2)
         endif
endif
endfunction
```

Penyelesaian (Cara I)

- Relasi Rekurens :

  $T(n) = 1$, $n = 0$ dan 1

  $T(n) = T(n-1)+T(n-2)+1$, $n > 0$

- Kompleksitas Waktu :

  $T(n) = T(n-1)+T(n-2)+1$ substituti $T(n-1) = T(n-2)+T(n-3)+1$

  $= (T(n-2)+T(n-3)+1)+T(n-2)+1$

  $= 2T(n-2)+T(n-3)+2$ substituti $T(n-2) = T(n-3)+T(n-4)+1$

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

  $= 3T(n-3)+2T(n-4)+4$ substituti $T(n-3) = T(n-4)+T(n-5)+1$

  $= 3(T(n-4)+T(n-5)+1)+ 2T(n-4)+4$

  $= 5T(n-4)+3T(n-5)+7$ substituti $T(n-4) = T(n-5)+T(n-6)+1$

  $= 5(T(n-5)+T(n-6)+1)+3T(n-5)+7$

  $= 8T(n-5)+5T(n-6)+12$

  $= \ldots$

  $= XT(n-i)+YT(n-(i+1))+(X+Y-1)$

Penyelesaian (Lanjutan Cara I)

- Ternyata X dan Y membentuk deret Fibonacci dengan X merupakan suku ke n=i+1 dan Y suku ke i, misal n-i =1 maka:

  T(n) = XT(n-i)+YT(n-(i+1))+(X+Y-1)

  = XT(1)+YT(1-1)+(X+Y-1)

  = XT(1)+YT(0)+(X+Y-1)   T(1)=T(0)=1

  = X+Y+X+Y-1

  = 2(X+Y)-1

- Deret Fibonacci dapat didefinisikan sebagai elemen barisan sbb:

Deret Fibonacci

$$U_n = \frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{2^n\sqrt{5}}$$

$$\begin{aligned}
jika\ n &= 0, U_n = 0 \\
n &= 1, U_n = 1 \\
n &= 2, U_n = 1 \\
n &= 3, U_n = 2 \\
n &= 4, U_n = 3 \\
n &= 5, U_n = 5 \\
n &= 6, U_n = 8 \\
n &= 7, U_n = 13 \\
n &= 8, U_n = 21
\end{aligned}$$

dan seterusnya

Penyelesaian (Lanjutan Cara I)

- Nilai X dan Y adalah sebagai berikut :

$$X = \frac{(1+\sqrt{5})^{i+1} - (1-\sqrt{5})^{i+1}}{2^{i+1}\sqrt{5}} \; dan \; Y = \frac{(1+\sqrt{5})^{i} - (1-\sqrt{5})^{i}}{2^{i}\sqrt{5}}$$

- T(n) $= 2(\frac{(1+\sqrt{5})^{i+1} - (1-\sqrt{5})^{i+1}}{2^{i+1}\sqrt{5}} + \frac{(1+\sqrt{5})^{i} - (1-\sqrt{5})^{i}}{2^{i}\sqrt{5}}) - 1$

Substitusi i = n-1

T(n) $= 2(\frac{(1+\sqrt{5})^{(n-1)+1} - (1-\sqrt{5})^{(n-1)+1}}{2^{(n-1)+1}\sqrt{5}} + \frac{(1+\sqrt{5})^{(n-1)} - (1-\sqrt{5})^{(n-1)}}{2^{(n-1)}\sqrt{5}})$ - 1

$$= 2(\frac{(1+\sqrt{5})^{n} - (1-\sqrt{5})^{n}}{2^{n}\sqrt{5}} + \frac{(1+\sqrt{5})^{(n-1)} - (1-\sqrt{5})^{(n-1)}}{2^{(n-1)}\sqrt{5}}) - 1$$

$$= \frac{2}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{n} - \left(\frac{1-\sqrt{5}}{2}\right)^{n} + \left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\right) - 1$$

T(n) masuk dalam kelas $\left(\frac{1+\sqrt{5}}{2}\right)^{n} \approx 1.62^{n} \approx 2^{n}$

Penyelesaian : Cara II

- T(n) = T(n-1) + T(n-2) +1
- Karena T(n-1) memiliki tingkat rekursif yang paling dalam daripada T(n-2) maka T(n-2) ≈ T(n-1), sehingga

T(n) = 2 T(n-1) +1

$$= 2(2T(n-2)+1)+1 = 2^2 T(n-2)+2+1$$
$$= 2^2(2T(n-3)+1)+2+1 = 2^3 T(n-3)+2^2+2+1$$
$$= 2^3(2T(n-4)+1)+2^2+2+1 = 2^4 T(n-4)+2^3+2^2+2+1$$
$$= 2^4(2T(n-5)+1)+2^3+2^2+2+1 = 2^5 T(n-5)+2^4+2^3+2^2+2+1$$
$$= 2^5(2T(n-6)+1)+2^4+2^3+2^2+2+1 = 2^6 T(n-6)+2^5+2^4+2^3+2^2+2+1$$

.

.

$$= 2^i T(n-i)+2^{(i-1)}+2^{(i-2)}+\ldots\ldots+2^2+2+1$$

$$misal\,(n-i)=1,\quad dan\ i=n-1,\quad maka$$

$$= 2^{(n-1)} T(1)+2^{((n-1)-1)}+2^{((n-2)-2)}+\ldots\ldots+2^2+2+1$$
$$= 2^{(n-1)}(1)+2^{((n-1)-1)}+2^{((n-1)-2)}+\ldots\ldots+2^2+2+1$$
$$= 2^{(n-1)}+2^{(n-2)}+2^{(n-3)}+\ldots\ldots+2^2+2+1$$

$$T(n) = 2^{(n-1)}+2^{(n-2)}+2^{(n-3)}+\ldots\ldots+2^2+2+1$$
$$T(n) = 1+2+\ldots\ldots+2^{(n-3)}+2^{(n-2)}+2^{(n-1)}$$
$$T(n) = 2^0+2^1+\ldots\ldots+2^{(n-3)}+2^{(n-2)}+2^{(n-1)}$$

Penyelesaian : Cara II

- Banyak suku = n
- Karena T(n) membentuk deret geometri maka :

$$a = 1 \ dan \ r = \frac{U_n}{U_{n-1}} = \frac{2}{1} = 2$$

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

$$T(n) \ masuk \ pada \ kelas \ 2^n$$

Latihan (Persoalan Minimum dan Maksimum)

procedureMinMaks2(inputA : TabelInt, i, j : integer,

outputmin, maks : integer)

{ Mencari nilai maksimum dan minimum di dalam tabel A yang berukuran n elemen secara Divide and Conquer.

Masukan: tabel A yang sudah terdefinisi elemen-elemennya

Keluaran: nilai maksimum dan nilai minimum tabel

}

Deklarasi

min1, min2, maks1, maks2 : integer

Persoalan Minimum & Maksimum

```
if i=j then                          { 1 elemen }
     min←Ai
     maks←Ai
    else
     if (i = j-1) then               { 2 elemen         }
       if Ai < Aj then
          maks←Aj
          min←Ai
       else
         maks←Ai
         min←Aj
       endif
```

Persoalan Minimum & Maksimum

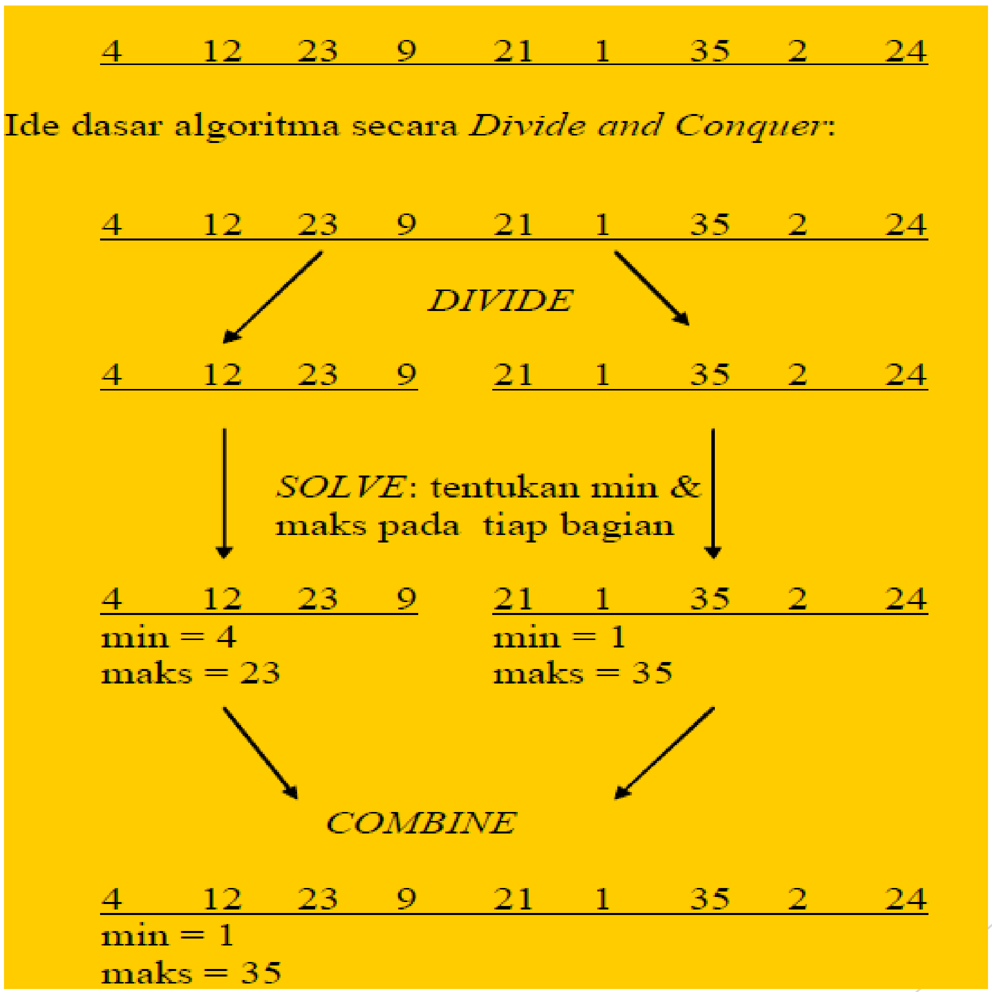
```
else                          { lebih dari 2 elemen }
     k←(i+j) div 2  { bagidua tabel pada posisi k }
     MinMaks2(A, i, k, min1, maks1)
     MinMaks2(A, k+1, j, min2, maks2)
     if min1 < min2 then
       min←min1
     else
       min←min2
     endif

     if maks1<maks2 then
       maks←maks2
     else
       maks←maks2
     endif
```

Latihan (Persoalan Minimum dan Maksimum)

- Buatlah relasi rekurens nya!
- Berapakah Big O nya?

Penyelesaian : Konsep Masalah

- Misalkan Tabel A berisi elemen berikut ini :



Penyelesaian : Konsep Masalah

- Ukuran tabel hasil pembagian dapat dibuat cukup kecil sehingga mencari minimum dan maksimum dapat diselesaikan (SOLVE) secara lebih mudah.

- Dalam hal ini, ukuran kecil yang dipilih adalah 1 elemen atau 2 elemen.

Penyelesaian : Konsep masalah

MinMaks(A, n, min, maks)

Algoritma:

1. Untuk kasus $n = 1$ atau $n = 2$,
   SOLVE: Jika $n = 1$, maka $min = maks = A[n]$
   Jika $n = 2$, maka bandingkan kedua elemen untuk menentukan *min* dan *maks*.

2. Untuk kasus $n > 2$,
   (a) DIVIDE: Bagi dua tabel *A* menjadi dua bagian yang sama, A1 dan A2

   (b) CONQUER:
   MinMaks(A1, n/2, min1, maks1)
   MInMaks(A2, n/2, min2, maks2)

   (c) COMBINE:
   if min1 <min2 then min <- min1 else min <- min2
   if maks1 <maks2 then maks <- maks2 else maks <- maks1

Penyelesaian : Konsep masalah

- Untuk masalah sebelumnya dapat di buat menjadi :

DIVIDE dan CONQUER:

4 12 23 9	21 1 35 2 24

4 12	23 9	21 1	35 2 24

4 12	23 9	21 1	35	2 24

SOLVE dan COMBINE:

4 12	23 9	21 1	35	2 24
min = 4	min = 9	min = 1	min = 35	min = 2
maks = 12	maks = 23	maks = 21	maks =35	maks = 24

4 12 23 9	21 1	35 2 24
min = 4	min = 1	min = 2
maks = 23	maks = 21	maks = 35

4 12 23 9	21 1 35 2 24
min = 4	min = 1
maks = 23	maks = 35

4 12 23 9 21 1 5 2 24
min = 1
maks = 35

Algoritma

```
procedure MinMaks2(input A : TabelInt, i, j : integer,
                   output min, maks : integer)
{ Mencari nilai maksimum dan minimum di dalam tabel A yang berukuran n
elemen secara Divide and Conquer.
Masukan: tabel A yang sudah terdefinisi elemen-elemennya
Keluaran: nilai maksimum dan nilai minimum tabel
}
Deklarasi
      min1, min2, maks1, maks2 : integer

Algoritma:
      if i=j then                    { 1 elemen  }
        min←Ai
        maks←Ai
      else
        if (i = j-1) then            { 2 elemen  }
           if Ai < Aj then
               maks←Aj
            min←Ai
           else
             maks←Ai
             min←Aj
           endif
        else                         { lebih dari 2 elemen }
            k←(i+j) div 2            { bagidua tabel pada posisi k }
            MinMaks2(A, i, k, min1, maks1)
            MinMaks2(A, k+1, j, min2, maks2)
            if min1 < min2 then
               min←min1
            else
               min←min2
            endif

            if maks1<maks2 then
               maks←maks2
            else
               maks←maks2
            endif
```

Penyelesaian

- Relasi Rekurens: (Dilihat dari operasi dasar ➔ perbandingan)

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

- Kompleksitas Waktu

Asumsi: $n = 2^k$, dengan k bilangan bulat positif, maka

$$\begin{aligned} T(n) &= 2\mathrm{T}(n/2) + 2 \\ &= 2(2T(n/4) + 2) + 2 = 4T(\mathrm{n}/4) + 4 + 2 \\ &= 4T(2\mathrm{T}(n/8) + 2) + 4 + 2 = 8T(n/8) + 8 + 4 + 2 \\ &= \ldots \\ &= 2^{k-1}\, T(2) + \sum_{i=1}^{k-1} 2^i \\ &= 2^{k-1} \cdot 1 + 2^k - 2 \\ &= n/2 + n - 2 \\ &= 3n/2 - 2 \\ &= O(n) \end{aligned}$$

SOAL:

Dengan mencari nilai C dan n_0

1. Tunjukkkan bahwa $T(n) = n + 150 = O(n)$!
2. Tunjukkkan bahwa $T(n) = 8n^2 + 5n + 2 = O(n^2)$!
3. Tunjukkkan bahwa $T(n) = (n^2 /4) + 2^n = O(2^n)$!
4. Tunjukkkan bahwa $T(n) = \log n^2 =2 \log n = O(\log n)$!

5. Tentukan notasi O, Ω dan Θ untuk $T(n) = 5n^2 + 8n \log n$

SOAL-SOAL LATIHAN

- Tentukan notasi O, Ω dan Θ untuk T(n) = 1+2+...+n
- Tentukan kompleksitas waktu dari algoritma dibawah ini jika melihat banyaknya operasi a←a+1

```
for i ← 1 to n do
   for j ← 1 to i do
       for k ← j to n do
            a ← a + 1
       endfor
   endfor
endfor
```

Tentukan pula nilai *O*-besar, *Ω*-besar, dan *Θ*-besar dari algoritma diatas (harus penjelasan)

Penyelesaian 1

$1 + 2 + \ldots + n = O(n^2)$ karena

$1 + 2 + \ldots + n \leq n + n + \ldots + n = n^2$ untuk $n \geq 1$.

$1 + 2 + \ldots + n = \Omega(n)$ karena

$1 + 2 + \ldots + n \geq 1 + 1 + \ldots + 1 = n$ untuk $n \geq 1$.

$$
\begin{aligned}
1 + 2 + \ldots + n &\geq \lceil n/2 \rceil + \ldots + \lceil n/2 \rceil + \lceil n/2 \rceil \\
&= \lceil (n + 1)/2 \rceil \lceil n/2 \rceil \\
&\geq (n/2)(n/2) \\
&= n^2/4
\end{aligned}
$$

Kita menyimpulkan bahwa

$$1 + 2 + \ldots + n = \Omega(n^2)$$

Oleh karena itu,

$$1 + 2 + \ldots + n = \Theta(n^2)$$

Penyelesaian 2

Untuk $i = 1$,

    Untuk $j = 1$, jumlah perhitungan = n kali

Untuk $i = 2$,

    Untuk $j = 1$, jumlah perhitungan = n kali

    Untuk $j = 2$, jumlah perhitungan = $n - 1$ kali

...

Untuk $i = n$,

    Untuk $j = 1$, jumlah perhitungan = n kali

    Untuk $j = 2$, jumlah perhitungan = $n - 1$ kali

    ...

    Untuk $j = n$, jumlah perhitungan = 1 kali.

Jadi jumlah perhitungan = $T(n) = n^2 + (n - 1)^2 + (n - 2)^2 + \ldots + 1$

- $T(n) = O(n^3) = \Omega(n^3) = \Theta(n^3)$.

- Salah satu cara penjelasan:

  $T(n) = n^2 + (n - 1)^2 + (n - 2)^2 + \ldots + 1$

  $= n(n + 1)(2n + 1)/6$

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

- Diperoleh $T(n) \le 3n^3$ untuk $n \ge 4$ dan

  $T(n) \ge 2n^3$ untuk $n \ge 1$.