LAPORAN PRAKTIKUM

ANALISIS ALGORITMA



Disusun Oleh :

Risyad Pangestu

140810170003

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2019

1. Dua algortima berikut menggunakan data berupa array a[1 .. n], b[n,n], dan d[n,n].

Pertanyaan : cari masing masing T(n), Kompleksitas O, c, dan N0 !

**// Soal 1.a**

for j=1 to n-1

k=j // n-1 Kali operasi Assigment

for i=j+1 to n

if a[i] < a[k] then // loop (a)

k=i

endif

endfor

tm=a[j] // n-1 Kali operasi Assigment

a[j]=a[k] // n-1 Kali operasi Assigment

a[k]=tm // n-1 Kali operasi Assigment

endfor

loop (a) :

j i Jumlah looping

1 2 3 (n-1)

2 3 2 (n-2)

...

n-1 n 1

T(n) loop a = 2 (n-1/2((n-1)+1))

= 2 (n-1/2 (n))

= n^2 - n

//dikali 2 untuk dua statement

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

= 4n - 4 + n^2 - n

= n^2 + 3n - 4

= O(n^2)

T(n) <= c . f(n)

n^2 + 3n - 4 <= c . n^2

ambil n0 = 1

1 + 3 - 4 <= c . 1

0 <= c

c >= 0

jadi, n0 = 1 dan c = 0

dengan kompleksitas O(n^2)

**//Soal 1.b**

for i=0 to n-1 // 5.

for j=0 to n-1 // 4.

c[c,j]=0 // 3.

for k=0 to n-1 // 2.

cij=d[i,k] and b[k,j] // 1.

c[i,j]=c[i,j] or cij

endfor

endfor

endfor

1. 2

2. 2 \* n = 2n

3. 2n + 1

4. n (2n+1)

5. n (2n^2 + n) = 2n^3 + n^2

jadi, T(n) = 2n^3 + n^2

= O(n^2)

T(n) <= c . f(n)

2n^3 + n^2 <= c . n^2

ambil no = 1

2 + 1 <= c . 1

c >= 3

jadi, n0 = 1 dan c = 3

dengan kompleksitas O(n^3)

2. Algortima berikut menggunakan data a[1..n]

1. algortima 1

Ada = 0; // 1 kali operasi Assigment

Kx=1; // 1 kali operasi Assigment

Input br; // 1 kali operasi Assigment

For (i=1; i<n; i++){ // Looping (a)

If(a[i] == br && (!ada)){

Ada = 1;

kx = 1;

i = n+1;

}

}

Looping (a) :

Kasus terburuk terjadi ketika kondisi If tak terpenuhi, maka kondisi looping

akan sebanyak n kali .

T(n) = n + 1 + 1 + 1

= n + 3

= O(n)

T(n) <= c . f(n)

n + 3 <= c . n

ambil n0 = 1

1 + 3 <= c . 1

c >= 4

jadi n0 = 1 dan c=4

dengan kompleksitas O(n)

(b) Algoritma 2

L=1; // 1 kali operasi Assigment

R=n; // 1 kali operasi Assigment

Ada = 0; // 1 kali operasi Assigment

Input br;

While((L <= R) && (!ada)){ // looping (a)

M = (L+R) div 2;

If (a[m] == br)

Ada = 1;

Else if (br<a[m])

R = m+1;

Else

L=m+1;

}

Looping (a) :

“ M = (L+R) div 2 “

dilakukan sebanyak :

Loop ke - 1 = n/2

Loop ke - 2 = n/2\*2

Loop ke - 3 = n/2\*2\*2

...

Loop ke - k = n/2^k

maka, k = 2log n

M = (L+R) div 2; //2log n

If (a[m] == br) //2log n

Ada = 1; // 1 kali

Else if (br<a[m]) //2log n

R = m+1; atau L=m+1; //2log n

T(n) = 1 + 1 + 1 + 4 (2log n) + 1

= 4 + 4(2log n)

= O(2log n)

T(n) <= c . f(n)

4 + 4(2log n) <= c . (2log n)

4/2log n + 4 <= c

ambil n=2

4 / 2log2 + 4 <= c

8 <= c

c >= 8

jadi, n0=2 dan c=8

dengan kompleksitas O(2log n)

1. Masing masing T(n), Kompleksitas O, c, dan N0 !

jadi n0 = 1 dan c=4

dengan kompleksitas O(n)

1. jadi n0 = 2 dan c=8

dengan kompleksitas O(2log n)

1. Running Time

Komputer A dengan Algoritma A dan kecepatan 10-9 detik/instruksi. Jika n = 10^8 maka

running time = O(10^8) x 10^-9 detik/instruksi = 10^8 x 10^-9 = 0.1 detik

Komputer B dengan Algoritma A dan kecepatan 10-7 detik/instruksi. Jika n = 108 maka

running time = O(2log 10^8) x 10^-7 detik/instruksi = 26.578 x 10^-7 = 2.6578 x 10^-6 detik

Terbukti Algoritma B jauh lebih baik dan dengan simulasi diatas kita bisa menyimpulkan

bahwa perancangan Algoritma yang baik sangat penting dan berpengaruh pada kecepatan suatu program