



CODELABS
BUILD SOCIETY WITH TECHNOLOGY

ANALISIS ALGORITMA

Teorema dan Aturan Perhitungan Notasi
Asimptotik

Disusun Oleh:

Adam Mukharil Bachtiar
Teknik Informatika UNIKOM
adfbipotter@gmail.com



AGENDA PERKULIAHAN

- ➡ Penjelasan Teorema Notasi Asimptotik
- ➡ Aturan Perhitungan Notasi Asimptotik



Teorema Notasi Asimptotik (O)

TEOREMA 1

Isi Teorema:

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

Contoh:

$$\Rightarrow T(n) = 100n + 5 = O(n)$$

$$\Rightarrow T(n) = \frac{1}{2} n (n-1) = O(n^2)$$

TEOREMA 2

Isi Teorema:

Misalkan $T_1(n) = O(f(n))$ dan $T_2(n) = O(g(n))$, maka $T_1(n) + T_2(n) = O(f(n)) + O(g(n)) = O(\max(f(n), g(n)))$

Contoh:

$T_1(n) = O(n)$ dan $T_2(n) = O(n^2)$, maka $T_1(n) + T_2(n) = O(\max(n, n^2)) = O(n^2)$

TEOREMA 3

Isi Teorema:

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

Contoh:

$T_1(n) = O(n)$ dan $T_2(n) = O(n^2)$, maka $T_1(n)T_2(n) = O(n.n^2) = O(n^3)$

TEOREMA 4

Isi Teorema:

Misalkan $T_1(n) = O(f(n))$ dan $T_2(n) = O(g(n))$, maka $O(cf(n)) = O(f(n))$, c adalah konstanta.

Contoh:

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

TEOREMA 5

Isi Teorema:

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

Contoh:

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

Aturan Perhitungan Notasi Asimptotik

T(n) DARI ALGORITMA DIKETAHUI

Contoh T(n):

$$T(n) = (n + 2) \log(n^2 + 1) + 5n^2 = O(n^2)$$

Langkah 1:

$$\begin{aligned} T(n) &= (n + 2) \log(n^2 + 1) + 5n^2 \\ &= f(n) \quad \cdot \quad g(n) \quad + h(n) \end{aligned}$$

Langkah 2:

$$\Rightarrow f(n) = (n + 2) = O(n)$$

$$\Rightarrow g(n) = \log(n^2 + 1) = O(\log n)$$

$$\Rightarrow h(n) = 5n^2 = O(n^2)$$

Langkah 3:

$$\begin{aligned} T(n) &= (n + 2) \log(n^2 + 1) + 5n^2 \\ &= O(n) \cdot O(\log n) + O(n^2) \\ &= O(n \log n) + O(n^2) \\ &= O(\max(n \log n, n^2)) \\ &= O(n^2) \end{aligned}$$

PERHITUNGAN DARI ALGORITMA (I)

Panduan Perhitungan:

Pengisian nilai (*assignment*), perbandingan, operasi aritmetik, *input*, dan *output* membutuhkan waktu $O(1)$.

Contoh:

$$\Rightarrow \text{input(nama)} = O(1)$$

$$\Rightarrow a \leftarrow a + 1 = O(1)$$

PERHITUNGAN DARI ALGORITMA (2)

Panduan Perhitungan:

Pengaksesan elemen larik atau memilih *field* tertentu dari sebuah *record* membutuhkan waktu $O(1)$.

Contoh:

$$\Rightarrow a \leftarrow a + x[i] = O(1)$$

$$\Rightarrow \underline{\text{output}}(\text{mhs.nim}) = O(1)$$

PERHITUNGAN DARI ALGORITMA (3)

Panduan Perhitungan:

if C then S1 else S2; membutuhkan waktu $T_C + \max(T_{S1}, T_{S2})$

Contoh:

| | |
|----------------------------|--------|
| read(x); | $O(1)$ |
| if x mod 2 = 0 then | $O(1)$ |
| begin | |
| x:=x+1; | $O(1)$ |
| writeln(x); | $O(1)$ |
| end | |
| else | |
| writeln(x); | $O(1)$ |

Kompleksitas waktu asimptotik:
= $O(1) + O(1) + \max(O(1)+O(1), O(1))$
= $O(1) + \max(O(1), O(1))$
= $O(1) + O(1)$
= $O(1)$

PERHITUNGAN DARI ALGORITMA (4)

Panduan Perhitungan:

Kompleksitas waktu pengulangan **for** adalah jumlah pengulangan dikali dengan kompleksitas waktu badan (*body*) pengulangan.

Contoh:

```
for i:=1 to n do  
    jumlah:=jumlah + a[i];    O(1)
```

Kompleksitas waktu asimptotik

$$= n \cdot O(1)$$

$$= O(n \cdot 1)$$

$$= O(n)$$

PERHITUNGAN DARI ALGORITMA (5)

Perhitungan Pengulangan Bersarang:

```
for i ← 1 to n do  
    for j ← 1 to n do  
        a[i,j] ← 0                O(1)
```

Pemecahan:

$$n \cdot O(n) = O(n \cdot n) = O(n^2)$$

PERHITUNGAN DARI ALGORITMA (6)

Perhitungan Pengulangan Bersarang Dua Instruksi:

```
for i ← 1 to n do
  for j ← 1 to i do
    begin
      a ← a+1;      O(1)
      b ← b-2        O(1)
    endfor
  endfor
endfor
```

Pemecahan:

Waktu untuk $a := a+1$: $O(1)$

Waktu untuk $b := b-2$: $O(1)$

total waktu untuk badan kalang = $O(1) + O(1) = O(1)$

Pengulangan terluar dieksekusi sebanyak n kali

Pengulangan terdalam dieksekusi sebanyak i kali,
 $i = 1, 2, \dots, n$

Jumlah pengulangan seluruhnya = $1 + 2 + \dots + n$
 $= n(n + 1)/2$

Kompleksitas waktu asimptotik = $n(n + 1)/2 \cdot O(1)$
 $= O(n(n + 1)/2) = O(n^2)$

PERHITUNGAN DARI ALGORITMA (1)

Panduan Perhitungan:

while C **do** S dan **repeat** S **until** C Untuk kedua buah pengulangan, kompleksitas waktunya adalah jumlah pengulangan dikali dengan kompleksitas waktu badan C dan S.

Contoh:

```
i:=2;           O(1)
while i <= n do  O(1)
begin
    jumlah:=jumlah + a[i] O(1)
    i:=i+1           O(1)
endwhile
```

Kompleksitas waktu asimptotiknya adalah

$$\begin{aligned} &= O(1) + (n-1) \{ O(1) + O(1) + O(1) \} \\ &= O(1) + (n-1) O(1) \\ &= O(1) + O(n-1) \\ &= O(1) + O(n) \\ &= O(n) \end{aligned}$$

PERHITUNGAN DARI ALGORITMA (8)

Panduan Perhitungan:

Untuk pengulangan yang tidak dapat ditentukan panjangnya maka kompleksitasnya diambil dari **kompleksitas waktu terburuk ($T_{\max}(n)$)**.

Contoh:

```
ketemu:=false;
while (p <> Nil) and (not ketemu) do
  if p^.kunci = x then
    ketemu:=true
  else
    p:=p^.lalu
{ p = Nil or ketemu }
```

Kompleksitas waktu asimptotiknya adalah
 $= O(n)$

TERIMA KASIH