

ALGORITMA

Istilah algoritma pertama kali diperkenalkan oleh seorang ahli matematika yaitu Abu Ja'far Muhammad Ibnu Musa Al Khawarizmi.

Yang dimaksud dengan algoritma adalah :

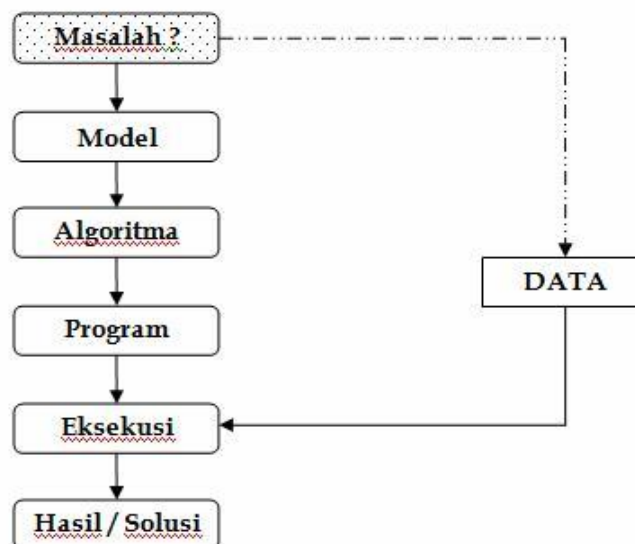
Urutan dari barisan instruksi untuk menyelesaikan suatu masalah. Adapun algoritma dapat dinyatakan dalam bentuk : flow chart, diagram alur, bahasa semu.

Sedangkan secara bahasa, algoritma berarti suatu metode khusus untuk menyelesaikan suatu masalah yang nyata (dari Webster Dictionary).

Dari suatu permasalahan yang akan diselesaikan, bisa terjadi terdapat lebih dari satu algoritma. Dalam memilih algoritma yang terbaik yang dapat digunakan, harus diperhatikan beberapa kriteria. Kriteria tersebut antara lain :

- Efektif dan efisien
- Jumlah langkahnya berhingga
- Berakhir
- Ada output
- Terstruktur

Adapun langkah-langkah yang dilakukan dalam proses penyelesaian masalah dengan bantuan komputer adalah sebagai berikut :



STUDI TENTANG ALGORITMA

Hal-hal yang akan dipelajari mengenai studi algoritma yaitu :

1. Bagaimana Merencanakannya
2. Bagaimana Menyatakannya
3. Bagaimana Validitasnya
4. Bagaimana Menganalisisnya
5. Bagaimana Menguji suatu program

Merencanakan algoritma : Merupakan suatu studi tentang teknik variasi design (model).

Menyatakan algoritma : Menyatakannya dengan singkat, dibuat dalam bahasa pemrograman yang terstruktur, misalnya Pascal, Bahasa C.

Validitas algoritma : Memenuhi kebutuhan yang diinginkan, dan perhitungannya / solusinya selalu benar untuk semua kemungkinan input yang legal.

Menganalisis algoritma : Perbandingan dari waktu perhitungan dan banyaknya storage / memori yang digunakan (efisiensi).

Menguji suatu program : Pengujian suatu program yang dilakukan dalam dua fase, yakni :

- Fase Debugging :
 - proses dari eksekusi program yang mengkoreksi kesalahan dalam bahasa pemrograman (logic & syntax).
- Fase Profiling :
 - program sudah benar
 - melihat/mengukur waktu tempuh & storage

ANALISIS ALGORITMA

Sebagaimana studi tentang algoritma, maka faktor yang sangat diperhitungkan adalah faktor efisiensi, yang meliputi :

a. Waktu tempuh (running time) :

- | | |
|------------------------------|---------------------------------|
| * Banyaknya langkah | * Jenis operasi |
| * Besar dan jenis input data | * Jenis komputer dan kompilator |

b. Jumlah memori yang dipakai

Dalam hal menganalisis algoritma, dikenal istilah kompleksitas. Kompleksitas adalah Sebuah fungsi $F(N)$ yang diberikan untuk waktu tempuh dan / atau kebutuhan storage dengan ukuran N input data. Kompleksitas ini dapat berupa kompleksitas waktu & memori

Beberapa definisi kompleksitas:

1. $f(n) = O(g(n)) \Leftrightarrow \exists$ dua konstanta positif c dan $n_0 \ni f(n) \leq c g(n) \forall n \geq n_0$ 2.

$f(n) = \Omega(g(n)) \Leftrightarrow \exists$ konstanta positif c dan $n_0 \ni f(n) \geq c g(n) \forall n > n_0$

3. $f(n) = \theta(g(n)) \Leftrightarrow \exists$ konstanta positif c_1, c_2 dan $n_0 \ni c_1 g(n) \leq f(n) \leq c_2 g(n) \forall n > n_0$

4. $f(n) \sim o(g(n)) \Leftrightarrow \exists$ sebuah konstanta positif $n_0 \ni \lim_{n \rightarrow \infty} f(n) / g(n) \rightarrow 1, \forall n > n_0$

Dari keempat definisi di atas, yang paling banyak digunakan untuk menganalisis algoritma adalah definisi 1 (Big Oh).

Teorema :

Jika $f(n) = a_m n^m + a_{m-1} n^{m-1} + \dots + a_1 n + a_0$ adalah polinomial tingkat m ,

maka $f(n) = O(n^m)$

Sebagai contoh :

$f(n) = 3n^5 + 4n^4 + 10n^2 + 56$	$= O(n^5)$
$f(n) = 9n^7 + 5n^6 + 36$	$= O(n^7)$
$f(n) = 8n^9$	$= O(n^9)$
$f(n) = n^6 + 19$	$= O(n^6)$
$f(n) = 25$	$= O(n^0) = O(1)$

Berikut ini adalah urutan dari Big Oh - Big Oh :

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n)$$

Berikut ini beberapa contoh analisis terhadap algoritma :

Contoh 1 :

(i) $c \leftarrow a + b$

- (ii) for i ← 1 to n do
 - c ← a + b
 - repeat
- (iii) for i ← 1 to n do
 - for j ← 1 to n do
 - c ← a + b
 - repeat
 - repeat

Analisisnya :

	banyaknya operasi +	f(n)	Big Oh
(i)	1 kali	f(n) = 1	O(1)
(ii)	n kali	f(n) = n	O(n)
(iii)	n ² kali	f(n) = n ²	O(n ²)

Contoh 2 :

Penjumlahan 2 buah matriks berorde (m X n) dan elemennya real

1. Set A[i,j], B[i,j], C[i,j] real
2. Untuk i ← 1 s/d m kerjakan
 - 3. untuk j ← 1 s/d n kerjakan
 - 4. C[i,j] ← A[i,j] + B[i,j]
 - 5. akhir j
6. akhir i

Analisisnya :

jumlah operasi + = **mn** kali
 jumlah memori = 3 mn x 4 = **12 mn** (asumsi : 1 elemen memerlukan 4 satuan memori/byte)
 total = **13 mn**

KEADAAN DARI KOMPLEKSITAS WAKTU ALGORITMA

Keadaan dari kompleksitas waktu algoritma meliputi :

- a. WORST Case → nilai maksimum dari $f(n) \forall$ input yang mungkin
- b. BEST Case → nilai minimum dari $f(n) \forall$ input yang mungkin
- c. AVERAGE Case → nilai ekspektasi dari $f(n)$

Contoh 3 :

Menentukan lokasi suatu elemen pada array data secara linier

1. Set $k := 1$; $loc := 0$
2. Repeat langkah 3 dan 4 While $loc := 0$ dan $k \leq n$
3. If Item := Data(k) then set $loc := k$
4. Set $k := k + 1$
5. If $loc := 0$ then
 Write *Elemen tidak ada pada array data*
 Else Write **loc adalah lokasi dari elemen**
6. Exit

Bila elemen (item) yang dicari merupakan elemen terakhir dari array tersebut atau tidak terdapat dalam array :

→ WORST CASE

⇒ $F(n) = O(n)$

Bila elemen yang dicari merupakan elemen pertama dari array tersebut :

→ BEST CASE

⇒ $F(n) = O(1)$

Bila elemen yang dicari berada diantara elemen pertama dan elemen terakhir dari array tersebut :

→ AVERAGE CASE

Banyaknya elemen dalam array tersebut adalah n , maka probabilitas masing-masing elemen adalah $1/n$

$$\begin{aligned}
 \Rightarrow F(n) &= 1 \cdot 1/n + 2 \cdot 1/n + 3 \cdot 1/n + \dots + n \cdot 1/n \\
 &= (1 + 2 + 3 + \dots + n) \cdot 1/n \\
 &= (n + 1) \cdot n/2 \cdot 1/n \\
 &= (n + 1)/2 \\
 &= 1/2 n + 1/2 \\
 &= O(n)
 \end{aligned}$$