
2 KONSEP DASAR ALGORITMA

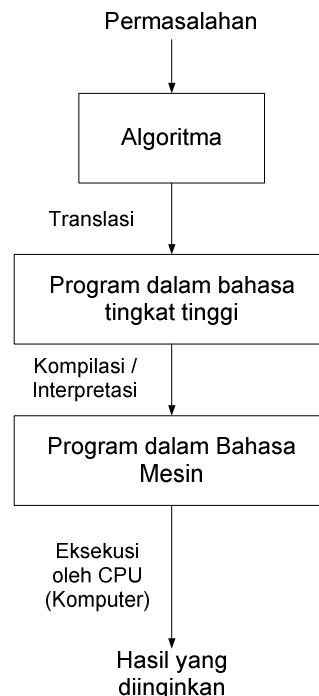
2.1 Pengertian Algoritma

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan tidak tergantung pada bahasa pemrograman tertentu.

Kata *Logis* merupakan kata kunci dalam Algoritma. Langkah-langkah dalam Algoritma harus logis (masuk akal dan mengikuti suatu urutan tertentu, tidak boleh melompat-lompat) serta harus dapat ditentukan bernilai salah atau benar.

Perbedaan Algoritma dan Program

Komputer hanyalah salah satu pemroses. Agar dapat dilaksanakan oleh komputer, algoritma harus ditulis dalam notasi bahasa pemrograman sehingga dinamakan program. Jadi program adalah perwujudan atau implementasi teknis Algoritma yang ditulis dalam bahasa pemrograman tertentu sehingga dapat dilaksanakan oleh komputer.



Tahapan Pelaksanaan algoritma oleh komputer

Dalam bidang komputer, algoritma sangat diperlukan dalam menyelesaikan berbagai masalah pemrograman, terutama dalam komputasi numeris. Tanpa algoritma yang dirancang baik maka proses pemrograman akan menjadi salah, rusak, atau lambat dan tidak efisien.

2.2 Algoritma untuk memecahkan permasalahan sehari-hari

Contoh 1 : Jika seorang ingin memasak atau membuat kue, baik itu melihat resep ataupun tidak pasti akan melakukan suatu langkah-langkah tertentu sehingga masakannya atau kuenya jadi dan rasanya enak.

Contoh 2 : Algoritma TUKAR ISI BEJANA

Diberikan dua buah bejana A dan B, bejana A berisi larutan berwarna merah, bejana B berisi larutan berwarna biru. Pertukarkan isi kedua bejana itu sedemikian sehingga bejana A berisi larutan berwarna biru dan bejana B berisi larutan berwarna merah.

Deskripsi Algoritma 1:

Aksi 1 : Tuangkan larutan dari bejana A ke dalam bejana B

Aksi 2 : Tuangkan larutan dari bejana B ke dalam bejana A.

- Algoritma TUKAR ISI BEJANA di atas tidak menghasilkan pertukaran yang benar. Langkah di atas tidak logis, hasil pertukaran yang terjadi adalah percampuran kedua larutan tersebut.
- Untuk mempertukarkan isi duah bejana, diperlukan sebuah bejana tambahan sebagai tempat penampungan sementara, misalnya bejana C. Maka algoritma untuk menghasilkan pertukaran yang benar adalah sebagai berikut :

Deskripsi Algoritma 2:

Aksi 1 : Tuangkan larutan dari bejana A ke dalam bejana C.

Aksi 2 : Tuangkan larutan dari bejana B ke dalam bejana A.

Aksi 3 : Tuangkan larutan dari bejana C ke dalam bejana B.

Contoh 3 : Ibu Tati mengupas kentang untuk makan malam

Sub masalah : 1. Apakah kentangnya harus dibeli dulu atau sudah di dapur?

2. Apakah pisau sudah siap?

3. Berapa jumlah kentang yang dikupas?

Maka kita harus membatasi dengan jelas keadaan awal dan keadaan akhirnya. Keadaan awal dan keadaan akhir algoritma dapat dijadikan acuan bagi pemrogram dalam merancang sebuah algoritma

Initial State(T0) : Kentang sudah ada di kantong plastik, yang ditaruh di lemari di dapur dimana Ibu Tati akan mengupasnya, pisau ada di rak.

Final State(T1) : 100 Kentang dalam keadaan terkupas siap untuk dimasak dan kantong kentangnya harus dikembalikan ke lemari lagi jika masih ada kentangnya

Deskripsi Algoritma 1

Aksi 1 : Ibu Tati mengambil kantong kentang dari lemari

Aksi 2 : Ibu Tati mengambil pisau dari rak

Aksi 3 : Ibu Tati mengupas kentang

Aksi 4 : Ibu Tati mengembalikan kantong kentang ke dalam lemari

- Deskripsi Algoritma di atas masih belum memenuhi Final State dimana kentang yang sudah dikupas ada 100 buah dan kantong kentang harus dikembalikan ke lemari jika masih ada kentangnya. Pada algoritma tersebut kentang yang dikupas hanya 1 dan Aksi 4 akan tetap dilaksanakan walaupun kantong kentang sudah kosong
- Supaya kentang yang sudah terkupas ada 100 maka perlu dilakukan proses **PENGULANGAN** pengupasan kentang sebanyak 100 kali. Dan supaya Ibu Tati hanya mengembalikan kantong kentang ke lemari hanya jika masih ada isinya, maka perlu ada **PEMILIHAN** berdasarkan kondisi isi kantong kentang. Maka algoritma untuk mencapai Final State yang benar adalah sebagai berikut :

Deskripsi Algoritma 2

Aksi 1 : Ibu Tati mengambil kantong kentang dari lemari dan

Aksi 2 : Ibu Tati mengambil pisau dari rak

Aksi 3 : Selama kentang terkupas < 100 maka

- Kupas 1 kentang

Aksi 4 : Lihat isi kantong

- Kantong Kosong → buang
- Kantong Tidak kosong → Kembalikan kantong ke lemari

Ciri penting algoritma:

- Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas.
- Setiap langkah harus didefinisikan dengan tepat dan tidak berarti-dua (Ambiguitas).
- Algoritma memiliki nol atau lebih masukan (input).
- Algoritma memiliki nol atau lebih keluaran (output).
- Algoritma harus efektif (setiap langkah harus sederhana sehingga dapat dikerjakan dalam waktu yang efisien).

2.3 Struktur Dasar Algoritma

Langkah-langkah penyelesaian masalah bisa berupa :

a. Runtunan (*sequence*)

Sebuah runtunan terdiri dari satu atau lebih instruksi. Tiap instruksi dikerjakan berurutan sesuai aturan penulisannya. Urutan instruksi menentukan keadaan akhir algoritma, jika urutannya diubah maka hasil akhirnya mungkin akan berubah. Urutan instruksi menunjukkan cara berfikir penyusun algoritma dalam menyelesaikan masalah

Runtunan Instruksi : Instruksi 1
 Instruksi 2
 Instruksi 3

Contoh : **Algoritma Tukar isi Bejana**

Runtunan instruksi :

1. Tuangkan larutan dari bejana A ke dalam bejana C
2. Tuangkan larutan dari bejana B ke dalam bejana A
3. Tuangkan larutan dari bejana C ke dalam bejana B

Hasil akhir :

Bejana A berisi larutan dari bejana B, bejana B berisi larutan dari bejana A

Jika runtunan instruksi diubah maka hasilnya berubah

b. Pemilihan (*selection*)

Adakalanya sebuah instruksi dikerjakan jika sebuah kondisi tertentu terpenuhi

Struktur umum :

```
If  kondisi          atau          If  kondisi  then  
then                                     Aksi 1  
      Aksi                               Else  
                                           Aksi 2
```

Contoh :

```
If Amir memperoleh juara kelas then  
    Ayah akan membelikannya hadiah  
If Jalan Dago macet then  
    Ambil alternative Jalan Dipati Ukur  
If Kantong Kentang kosong then  
    Buang  
Else  
    Kembalikan kantong kentang ke lemari  
Endif
```

c. Pengulangan (*repetition*)

Komputer tidak pernah bosan dan lelah jika diminta untuk mengerjakan instruksi secara berulang-ulang.

Contoh :

- Menulis kalimat "Saya harus lebih giat belajar" sebanyak 1000 kali
Ulangi :
 - Tulis kalimat " Saya harus lebih giat belajar"Sampai jumlah_kalimat = 1000
- Mengupas 100 buah kentang
Selama kentang terkupas < 100 maka
 - Kupas 1 kentang