

Konsep Algoritma

Asal kata Algoritma berasal dari nama Abu Ja'far Mohammed Ibn Musa al-Khwarizmi, ilmuwan Persia yang menulis kitab al jabr w'al-muqabbala (rules of restoration and reduction) sekitar tahun 825 M

Definisi Algoritma

Algoritma adalah urutan langkah logis tertentu untuk memecahkan suatu masalah. Yang ditekankan adalah urutan langkah logis, yang berarti algoritma harus mengikuti satu urutan tertentu, tidak boleh melompat-lompat.

Alur pemikiran dalam menyelesaikan suatu pekerjaan yang dituangkan secara tertulis. Yang ditekankan pertama adalah alur pikiran, sehingga algoritma seseorang dapat juga berbeda dari algoritma orang lain. Sedangkan penekanan kedua adalah tertulis, yang artinya dapat berupa kalimat, gambar, atau label tertentu.

Contoh :

Jika seseorang ingin mengirim surat kepada kenalamnya di tempat lain, langkah yang harus dilakukan adalah :

- Menulis surat
- Surat dimasukkan ke dalam amplop tertutup
- Amplop ditempel peranko secukupnya
- Pergi ke Kantor Pos terdekat untuk mengirimkannya

Contoh :

Algoritma menghitung luas persegi panjang:

- Masukkan panjang (P)
- Masukkan lebar (L)
- $Ls = P * L$
- Tulis Ls

Kriteria Algoritma Menurut Donald E. Knuth

1. **Input** : algoritma dapat memiliki nol atau lebih inputan dari luar.
2. **Output** : algoritma harus memiliki minimal satu buah output keluaran.
3. **Definiteness (pasti)**: algoritma memiliki instruksi-instruksi yang jelas dan tidak ambigu.
4. **Finiteness (ada batas)**: algoritma harus memiliki titik berhenti (stopping role).
5. **Effectiveness (tepat dan efisien)**: algoritma sebisa mungkin harus dapat dilaksanakan dan efektif.

3

Aturan Penulisan Teks Algoritma

1. **KEPALA ALGORITMA (HEADER)**
bagian yang terdiri atas nama algoritma dan penjelasan (spesifikasi) tentang algoritma tersebut.
2. **DEKLARASI/INISIALISASI**
bagian yang mendefinisikan semua nama yang digunakan didalam algoritma.
3. **DESKRIPSI**
bagian yang berisi uraian langkah-langkah penyelesaian masalah

Tahapan analisis Algoritma

1. Bagaimana merencanakan suatu algoritma
2. Bagaimana menyajikan suatu algoritma (bentuk algoritma)
 - Menggunakan narasi (*narrative description*) dalam bahasa natural (Bahasa Indonesia, Bahasa Inggris, dan bahasa manusia lainnya)
 - Tapi sering menggunakan (*ambiguities*)
 - Menggunakan *flow chart* (diagram alir)
 - Bagus secara visual akan tetapi cepat kalut algoritmanya Panjang
 - Menggunakan pseudo-code
 - Sudah lebih dekat ke bahasa pemrograman, namun sulit dimengerti oleh orang yang tidak mengerti pemrograman

4

Dengan narasi merupakan alat yang cukup efisien untuk menggambarkan suatu algoritma. Mengembangkan suatu algoritma yang akan dikomunikasikan kepada pemakai sistem.

Contoh :

Hitung Luas Segitiga
 Deklarasi/Instalisasi

Alas, tinggi, luas

Deskripsi

1. Masukkan Nilai alas
2. Masukkan Nilai tinggi
3. Hitung Luas segitiga sama dengan tinggi dikalikan alas dibagi dengan dua
4. Cetak hasil perhitungan luas

Contoh Bentuk Narasi (2)

ALGORITMA. MENUJMLAHKAN DUA BILANGAN BULAT

1. baca dua buah bilangan bulat (misalnya a dan b).
2. Jumlahkan dua bilangan tersebut dan simpan hasilnya ke peubah c ($c \leftarrow a + b$).
3. cetak nilai c.

ALGORITMA. MENUJMLAHKAN DUA BILANGAN BULAT

1. baca a dan b
2. $c \leftarrow a + b$.
3. cetak nilai c.

ALGORITMA. MENGHITUNG LUAS PERSEGI PANJANG:

1. Masukkan panjang (P)
2. Masukkan lebar (L)
3. $Ls = P * L$
4. Tampilkan Ls

Dengan bahasa semu (pseudocode), kode yang mirip dengan pemrograman sebenarnya. Pseudocode berasal dari kata Pseudou yang berarti imitasi, mirip atau menyerupai dengan kode bahasa pemrograman. Pseudocode ditulis berbasis bahasa pemrograman yang akan digunakan, misalnya Basic, Pascal, C++ dan lain-lain sehingga lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan kepada programmer.

Contoh : (Pseudocode-Pascal)

PROGRAM Menghitung Luas Segitiga

VAR \leftarrow alas,tinggi,luas : real

READ(alas)

READ(tinggi)

Luas (tinggi * alas) / 2

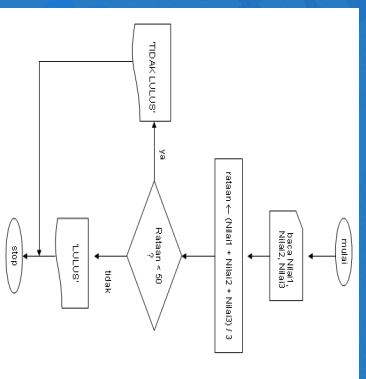
WRITE(luas)

Contoh Bentuk Pseudo Code

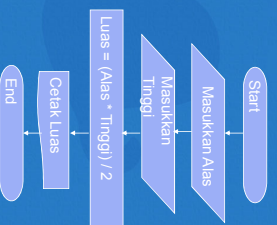
MENENTUKAN STATUS KELULUSAN DARI TIGA NILAI UJIAN

1. Read Nilai1, Nilai2, Nilai3.
2. rataaan \leftarrow (Nilai1 + Nilai2 + Nilai3) / 3.
3. If rataan < 50,
 Then Print 'TIDAK LULUS'
 Else Print 'LULUS'

Contoh Bentuk Flowchart (1)



Contoh Bentuk Flowchart (2)



3. Bagaimana validasi suatu algoritma
4. Bagaimana menganalisa suatu algoritma
5. Bagaimana menguji program dari suatu algoritma

Tahap Proses Uji program :

1. Fase Debugging
2. fase Running

Analisis Suatu Algoritma

Untuk melihat faktor efisiensi & efektifitas dari algoritma dapat dilakukan terhadap suatu algoritma dengan melihat pada :

1. Waktu tempuh (Running Time) dari suatu algoritma.
 - Dipengaruhi antara lain oleh :
 - Banyaknya langkah
 - Besar dan jenis input data
 - Jenis Operasi
 - Komputer dan Kompiator
2. Jumlah memori yang digunakan

Struktur Dasar Algoritma

1. Runtunan (sequence)

Sebuah runtunan terdiri dari satu atau lebih instruksi, yang berarti bahwa :

- Tiap instruksi dikerjakan satu kali
- Tiap instruksi dilaksanakan tepat satu kali
- Urutan instruksi yang dilaksanakan pemroses sama dengan urutan aksi
- Akhir dari instruksi terakhir merupakan akhir algoritma



2. Pemilihan (Selection)

Sebuah instruksi dikerjakan jika kondisi tertentu dipenuhi.

Menganalisis kasus algoritma artinya mendefinisikan :

- Kondisi, yaitu yang bernilai benar atau salah
- Aksi, yang akan dilaksanakan jika kondisi yang sesuai dengan aksi tersebut bernilai benar

Contoh :

1. Masukkan nilai x
2. Jika nilai x kurang dari 0, maka kerjakan baris 3, jika tidak kerjakan baris 4
3. Tulis negatif
4. Tulis positif

3. Pengulangan (Repetition)

Struktur pengulangan terdiri dari dua bagian :

- a. Kondisi pengulangan
- b. Badan (body) pengulangan

Pengulangan biasanya disertai dengan bagian :

- a. Inisialisasi
- b. Terminasi

Contoh :

1. Inisialisasi variable yang digunakan
2. Tentukan nilai awal hitungan
3. Bila hitungan belum mencapai lebih besar dari 10 maka ulang maka ulang blok instruksi berikut
 - Cetak kata "MERDEKA"
 - Hitungan ditambah satu
4. Selesai