

Pengenalan Algoritma

Pendahuluan

Komputer pada dasarnya adalah mesin yang tidak bisa apa-apa. Kita harus memberikan perintah untuk dapat berbicara (berkomunikasi) dengan komputer, dengan cara memberikan serangkaian instruksi kepada komputer agar komputer dapat memecahkan masalah. Langkah-langkah yang kita lakukan dalam memberikan instruksi untuk memecahkan masalah kita kita namakan pemrograman komputer. Untuk menyusun sebuah program yang besar dan kompleks, pemrogram membutuhkan tahapan penyusunan yang sistematis dan terpadu, yaitu:

1. Definisi Masalah
2. Analisis Kebutuhan
3. Penyusunan Algoritma
4. Pengkodean/Pemrograman
5. Testing dan Debugging
6. Pemeliharaan
7. Dokumentasi

Algoritma

Algoritma adalah kunci dari bidang ilmu komputer, dan pada dasarnya setiap hari kita melakukan aktivitas algoritma. Kata algoritma berasal dari sebutan Algorizm (Abu Abdullah Muhammad Ibn Musa Al Khwarizmi, ahli matematika Uzbekistan)

Definisi Algoritma

- Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika
- Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.
- Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.
- Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah.

➔ Langkah-langkah dalam algoritma *harus logis* dan harus dapat ditentukan *bernilai salah atau benar*.

Pembuatan algoritma harus selalu dikaitkan dengan:

- a. Kebenaran algoritma
- b. Kompleksitas (lama dan jumlah waktu proses dan penggunaan memori)

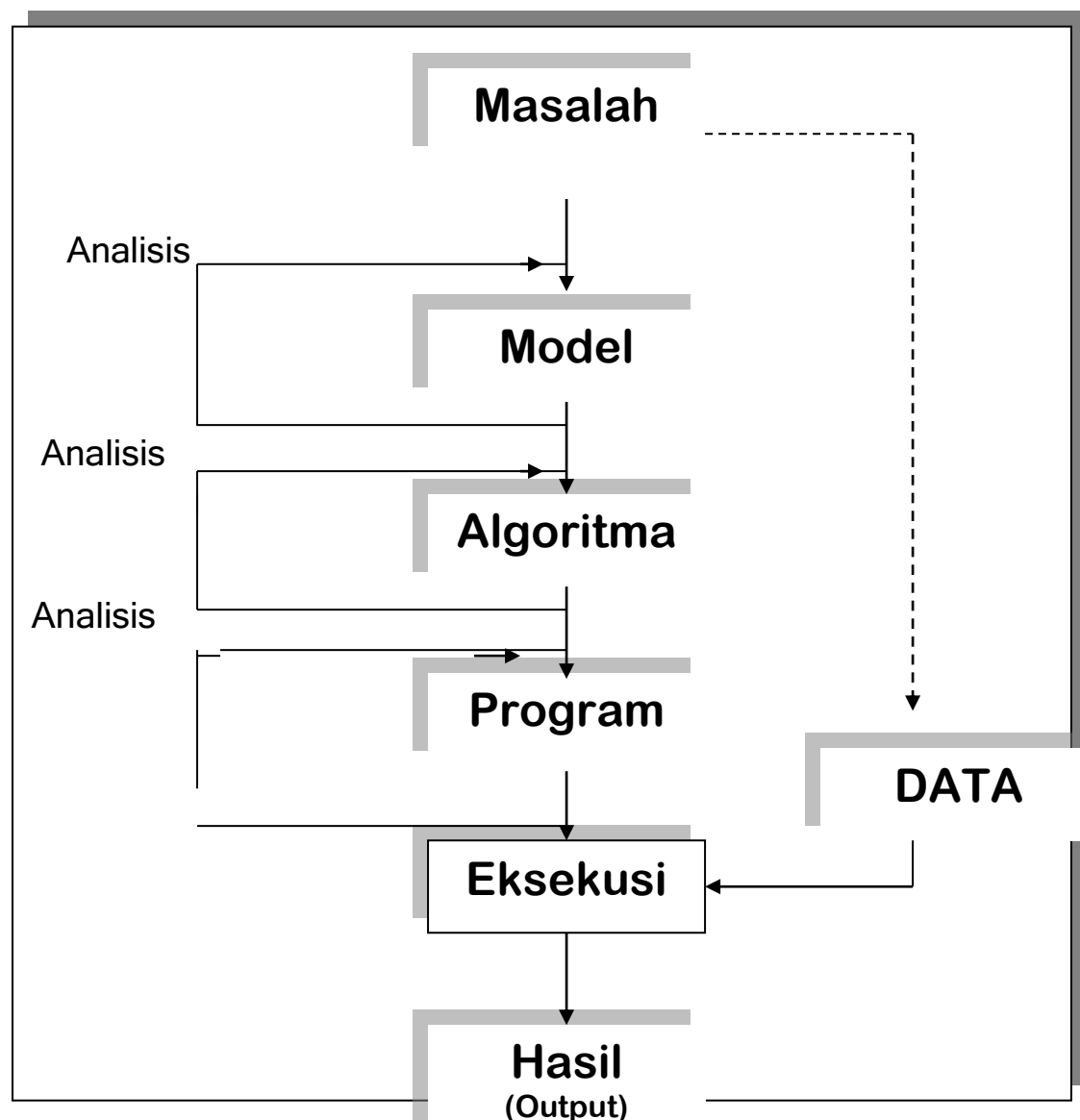
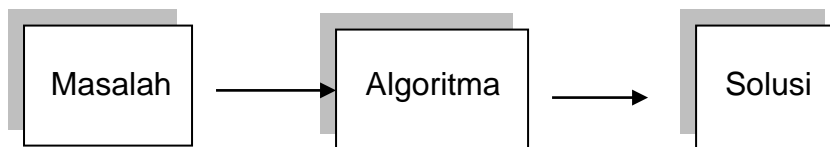
Kriteria Algoritma yang baik:

1. Tepat, benar, sederhana, standar dan efektif
2. Logis, terstruktur dan sistematis
3. Semua operasi terdefinisi
4. Semua proses harus berakhir setelah sejumlah langkah dilakukan

5. Ditulis dengan bahasa yang standar dengan format pemrograman agar mudah untuk diimplementasikan dan tidak menimbulkan arti ganda.

“Suatu algoritma harus menghasilkan output yang tepat guna (efektif) dalam waktu yang relatif singkat dan penggunaan memori yang relatif sedikit (efisien) dengan langkah yang berhingga dan prosesnya berakhir baik dalam keadaan diperoleh suatu solusi ataupun tidak adanya solusi.”

Proses Penyelesaian Masalah



Teknik Pemecahan Masalah

1. Teknik Top-Down
2. Teknik Bottom-Up

➔ Tahap Pemecahan Masalah:

Proses dari masalah hingga menjadi suatu algoritma

➔ Tahap Implementasi

Proses dari algoritma hingga menjadi suatu solusi. Solusi yang dimaksud adalah suatu program yang merupakan implementasi dari algoritma yang disusun.

➔ Contoh algoritma menulis surat, maka kita perlu melakukan beberapa langkah yaitu:

1. Mempersiapkan kertas dan amplop
2. Mempersiapkan alat tulis (pena, pensil)
3. Mulai menulis
4. Memasukkan kertas ke dalam amplop
5. Pergi ke kantor pos

➔ Contoh algoritma untuk menghitung luas lingkaran:

1. Masukan r (=jari-jari lingkaran)
2. Tentukan ϕ (=3.14)
3. $L = \phi * r * r$
4. Tulis L

NO.	Proses	Algoritma	Contoh Langkah dalam algoritma
1.	Membuat Kue	Resep kue	Masukan telur ke dalam wajan, kocok sampai mengembang
2.	Membuat pakaian	Pola pakaian	Gunting kain dari pinggir kiri bawah kearah kanan sejauh 5 cm
3.	Praktikum reaksi kimia	Panduan praktikum	Campurkan 10 ml air ke dalam 15 ml garam
4.	Merakit mobil	Panduan merakit	Sambungkan komponen A dengan komponen B
5.	Kegiatan sehari-hari	Jadwal harian	Pukul 15.00: tidur siang Pukul 16.00: membuat PR
6.	Memainkan musik	Papanot balok	Not balok
7.	Mengisi voicer HP	Panduan pengisian	Tekan 888 Masukan kode voucer

Penyajian Algoritma:

1. Teknik tulisan;
 - English Structure
 - Pseudocode
 2. Gambar;
 - Metode structure chart,
 - Hierarchy plus input-process-output
 - Flowchart
 - Nassi Schneiderman chart
-

English Structure:

- Menggunakan bahasa manusia (Inggris)
- Menggambarkan suatu algoritma yang akan dikomunikasikan kepada pemakai sistem

Pseudocode

- Pseudo= imitasi atau mirip / menyerupai
- Code=program
- Kode yang mirip dengan kode pemrograman yang sebenarnya
- Menggambarkan Algoritma yang akan dikomunikasikan kepada programmer
- Lebih rinci dari English Structure (mis: dalam menyatakan tipe data yang digunakan)

Structure Chart (bagan terstruktur)

- Digunakan untuk mendefinisikan dan mengilustrasikan organisasi dari system secara berjenjang
- Berbentuk modul dan submodul
- Menunjukkan hubungan elemen data dan elemen control serta hubungan antar modul

Aturan Penulisan Teks Algoritma

Setiap algoritma akan selalu terdiri dari :

- Judul (header)
- Deklarasi (kamus)
- Deskripsi Algoritma

🚦 JUDUL ALGORITMA ➔ Algoritma NAMA ALGORITMA

{Penjelasan tentang algoritma, berisi uraian singkat cara kerja program, kondisi awal dan akhir dari program} ➔ spesifikasi algoritma

Catatan, dalam menulis nama-nama dalam algoritma harus mempunyai makna yang mencerminkan proses, sifat atau identitas lainnya yang melekat dengan suatu proses, tipe, konstanta, variabel, sub-program dan lain-lainnya. Nama-nama yang bermakna disebut mnemonic.

🚦 DEKLARASI

(Semua nama yang dipakai, meliputi nama file, nama variable, nama konstanta, nama prosedur serta nama fungsi)

🚦 DESKRIPSI

(Semua langkah/aksi algoritma)

Contoh:

1). Kepala algoritma: Algoritma Luas_Lingkaran { Menghitung luas lingkaran dengan ukuran jari-jari tertentu .Algoritma menerima masukan jejari lingkaran, menghitung luasnya, dan menyajikan hasilnya ke piranti keluaran }

2) Deklarasi algoritma:

```
DEKLARASI { nama konstanta }  
  
const PHI = 3.14; { Nilai phi = 22/7 }  
  
{ nama peubah } var R : real; { input jejari lingkaran bilangan riil }  
I_Lingkaran : real; { luas lingkaran bilangan riil }  
  
{ nama sub program } procedure TUKAR (input/output A:integer,  
input/output B:integer)  
  
{Mempertukarkan nilai A dan B.Parameter A dan B sudah  
terdefinisi nilainya.Setelah pertukaran, A berisi nilai B dan B berisi  
nilai A }
```

3) Deskripsi algoritma:

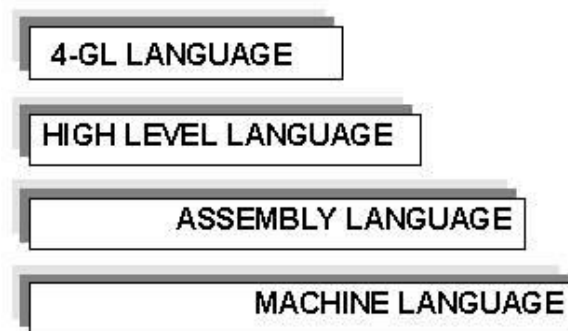
```
{ Baca data jejari lingkaran R.Jika R <= 0 tulis pesan data salah,  
selain itu hitung luas ingkaran. Tampilkan luas lingkaran. }  
baca(R);  
jika R <= 0 then tulis ("Data salah !") selain itu I_Lingkaran = PHI  
x R x R; tulis(I_Lingkaran);
```


Bahasa Pemrograman

Komputer mengerjakan transformasi data berdasarkan kumpulan perintah - program - yang telah dibuat oleh pemrogram. Kumpulan perintah ini harus dimengerti oleh computer, berstruktur tertentu (syntax) dan bermakna. Bahasa pemrograman merupakan notasi untuk memberikan secara tepat program computer.

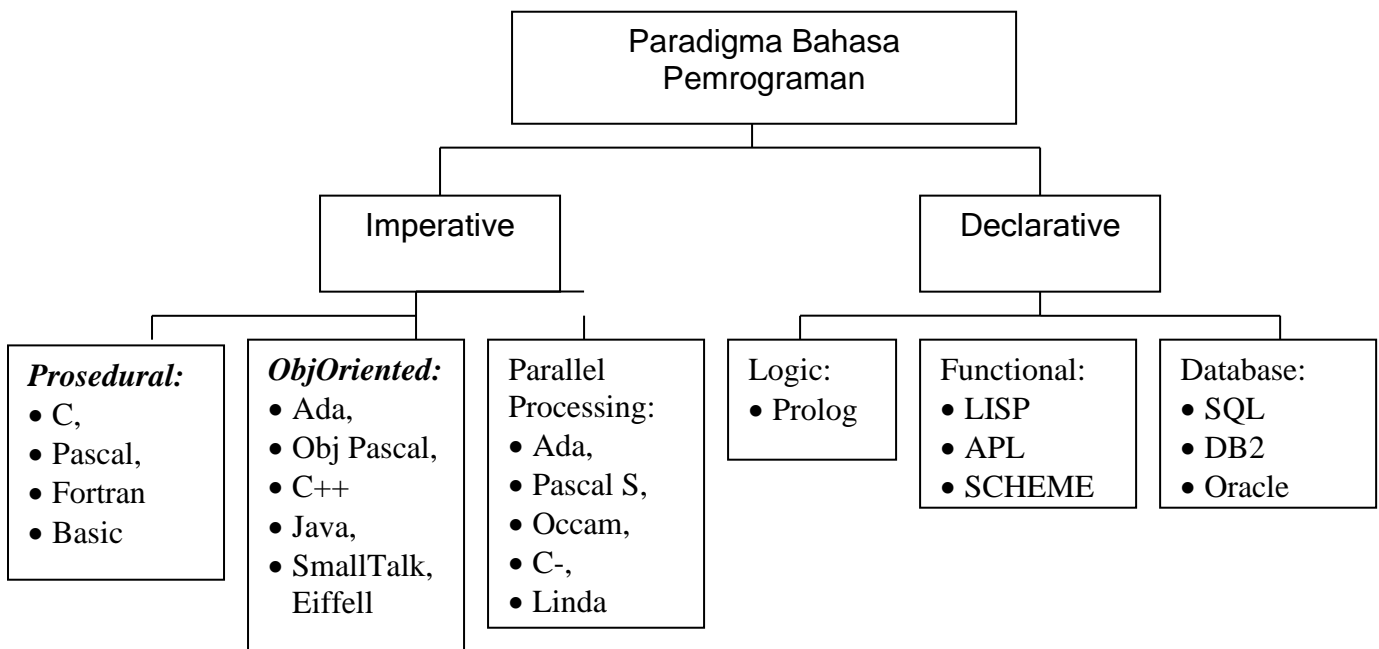
Generasi comput pemrograman:

- Generasi I: machine language
- Generasi II: assembly language : Asssembler
- Generasi III: high-level programming language: C, PASCAL, dsb.
- Generasi IV: 4 GL (fourth-generation language): SQL



Gambar 2. Tingkatan comput computer

Jenis Bahasa Pemrograman



Bahasa Pemrograman untuk tujuan tertentu.

Jenis Program	Bahasa Terbaik	Bahasa Terburuk
Data terstruktur	ADA, C /C++, PASCAL	Assembler, BASIC
Proyek cepat	BASIC	PASCAL, ADA, Assembler
Eksekusi cepat	Assembler, C	BASIC, Intrepreter Language
Kalkulasi matematika	FORTTRAN	PASCAL
Menggunakan memori dinamis	PASCAL, C	BASIC
Lingkungan bermemori terbatas	BASIC, Assembler, C	FORTTRAN
Program real-time	ADA, Assembler, C	BASIC, FORTTRAN
Manipulasi string	BASIC, PASCAL	C
Program mudah dikelola	PASCAL, ADA	C, FORTTRAN