

Nama : Zaidi mufthisor
NIM : 2100018105
Kelas : B

Algoritma, Flowchart and Pseudocode

1. Algoritma

Algoritma adalah serangkaian langkah yang diperintahkan untuk memecahkan masalah. Guna algoritma adalah untuk membantu seseorang dalam menyelesaikan suatu masalah berdasarkan pada pola pikirnya masing-masing.

Ciri – ciri algoritma:

- Ada input.
- Ada proses.
- Ada output.
- Memiliki instruksi yang jelas dan tidak ambigu.
- Harus mempunyai stopping role.

Sifat algoritma:

- Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman.
- Tidak tergantung pada suatu bahasa pemrograman.
- Notasi-notasi yang dapat digunakan untuk seluruh bahasa manapun.
- Algoritma adalah jembatan untuk mempermudah pemahaman awal kerja suatu proses.

2. Flowchart

Bentuk gambar/diagram yang mempunyai alirannya satu atau dua arah secara sekuensial.

Kegunaan

- Untuk mendesain program
- Untuk merepresentasikan program

Secara garis besar, unsur-unsur pemrograman adalah Input, Proses, dan Output. Semua bahasa pemrograman, komponen sebagai berikut:

- Input (scanf)
- Percabangan (if, switch)
- Perulangan (while, for, for each, loop)
- Output (printf)

Input -> Proses -> Output
pasti mempunyai komponen-

Notasi Flowchart

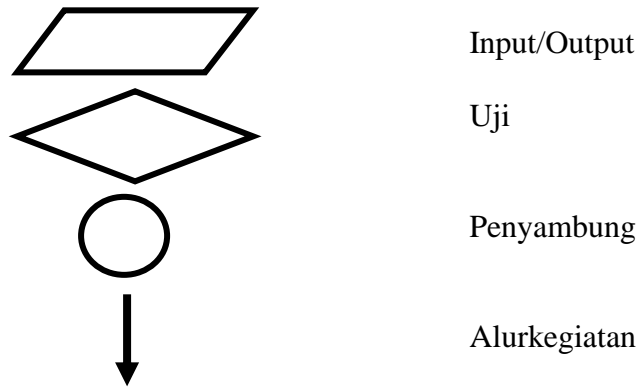
Simbol



Semantik

Mulai/Akhir

Proses



3. Pseudocode

- Pseudocode adalah kode atau tanda yang menyerupai (pseudo) atau merupakan penjelasan cara menyelesaikan suatu masalah.
- Pseudocode sering digunakan oleh seseorang untuk menuliskan algoritma dari suatu permasalahan
- Pseudocode adalah menyelesaikan suatu permasalahan (hampir sama dengan algoritma), hanya saja bentuknya sedikit berbeda dari algoritma.
- Pseudocode menggunakan bahasa yang hampir menyerupai bahasa pemrograman. Selain itu biasanya pseudo-code menggunakan bahasa yang mudah dipahami secara universal dan juga lebih ringkas daripada algoritma.

Kode tindakan dalam pseudocode

- Inisialisasi variabel
- Penugasan nilai ke variabel
- Operasi aritmatika
- Operasi relasional

Proses Pemecahan Masalah Langkah 1 - Analisis masalahnya Garis besar masalah dan persyaratannya Langkah-langkah desain (algoritma) untuk memecahkan masalah Langkah 2 - Terapkan algoritma Menerapkan algoritma dalam kode Verifikasi bahwa algoritme berfungsi Langkah 3 - Pemeliharaan Gunakan dan modifikasi program jika domain masalah berubah Contoh 1: Persegi Panjang Masalah: Buatlah algoritma untuk mencari keliling dan luas persegi panjang. Informasi: Keliling dan luas persegi panjang diberikan oleh rumus berikut: keliling = $2 * (\text{panjang} + \text{lebar})$ luas = $\text{panjang} * \text{lebar}$ Persyaratan: Masukan: panjang dan lebar persegi panjang Output: keliling dan luas persegi panjang Proses: keliling = ???, luas = ??? Algoritma: Hitunglah panjang persegi panjang Dapatkan lebar persegi panjang Cari keliling menggunakan persamaan berikut: keliling = $2 * (\text{panjang} + \text{lebar})$ Carilah luasnya dengan menggunakan persamaan berikut: luas = $\text{panjang} * \text{lebar}$ Menampilkan hasil keliling dan luas Contoh 2: Hitung Biaya Parkir Mobil Tempat parkir mobil dikenakan biaya sebagai berikut: Jam pertama berharga RM2.00. Jam-jam berikutnya berharga RM1.00 per jam. Tulis algoritme untuk menghitung biaya berdasarkan waktu masuk dan keluar kendaraan. Penyelesaian masalah Pemrograman adalah proses pemecahan masalah Teknik pemecahan masalah Analisis masalahnya Garis besar persyaratan masalah Langkah-langkah desain (algoritma) untuk memecahkan masalah Algoritma: Proses pemecahan masalah langkah demi langkah Solusi dicapai dalam waktu terbatas

Program style and form

1) PENGGUNAAN WHITESPACE

Masukkan karakter spasi putih (seperti kosong, tab, dan baris baru) jika perlu untuk meningkatkan keterbacaan kode sumber Anda.

Contoh:

```
int matriks[][3] = { 1, 0, 0, 0, 1, 0, 0, 0, 1 };
```

```
int matriks[][3] = { 1, 0, 0, 0, 1, 0, 0, 0, 1 };
```

Karakter spasi diabaikan oleh kompiler selama kompilasi. Ingatlah untuk memisahkan kata-kata dan pengidentifikasi yang dicadangkan dari satu sama lain dan simbol lainnya.

2) KOMA DAN SEMIKOLONS

- Koma memisahkan item dalam daftar.
Contoh: `int a, b, c;`
- Semua pernyataan C++ diakhiri dengan titik koma.
Contoh: `luas = panjang * lebar;`
- Titik koma juga disebut terminator pernyataan.

3) DOKUMENTASI

- Program lebih mudah dibaca dan dipelihara jika didokumentasikan dengan baik.
- Komentar dapat digunakan untuk mendokumentasikan kode
 - Komentar satu baris dimulai dengan `//` di mana saja di baris
 - Beberapa komentar baris diapit di antara `/*` dan `*/`

4) BENTUK DAN GAYA

Pertimbangkan dua cara untuk mendeklarasikan variabel:

- Metode 1
`int feet, inci;`
`double x, y;`

- Metode 2
int a, b;
dobel x,y;

Keduanya benar, namun, yang kedua sulit dibaca

Syntax and logical error

1) KESALAHAN SINTAKS

Kesalahan sintaks adalah kesalahan dalam kode sumber yang terkait dengan sintaks bahasa. Kesalahan sintaks terdeteksi oleh kompiler. File yang dapat dieksekusi akan dihasilkan oleh kompiler hanya jika kode sumber yang dikompilasi tidak memiliki kesalahan sintaksis. Kesalahan sintaks dilaporkan oleh kompiler dalam bentuk pesan kesalahan.

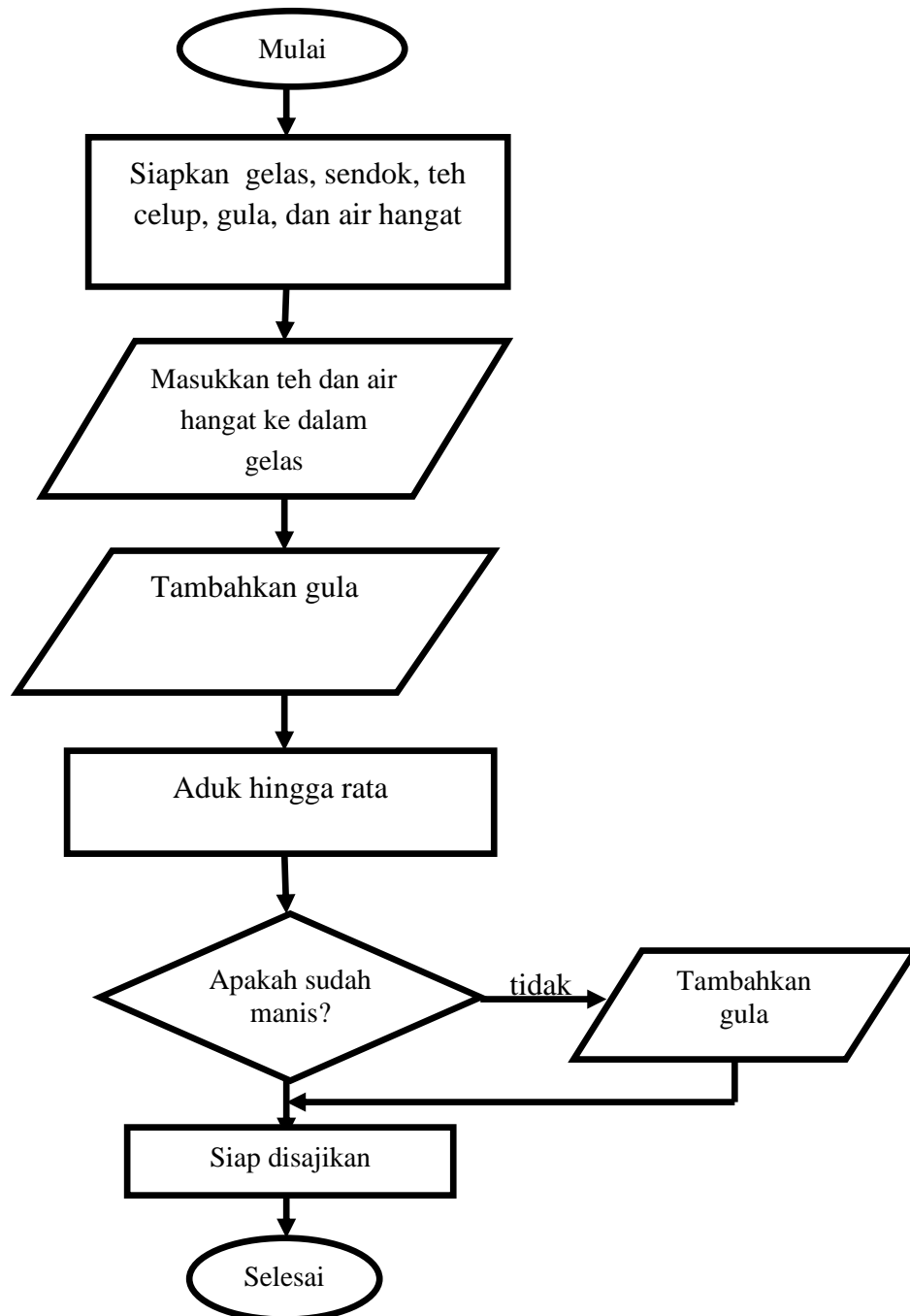
2) KESALAHAN LOGIS

Kesalahan logika adalah kesalahan yang berhubungan dengan logika program. Biasanya, kesalahan logis tidak terdeteksi oleh kompiler. Kesalahan logis biasanya terdeteksi selama runtime program. Misalnya, sebuah program yang menghasilkan hasil yang tidak diharapkan merupakan indikasi bahwa ia memiliki kesalahan logis. Penting untuk diingat bahwa jika kompilator tidak menghasilkan pesan kesalahan, itu tidak berarti bahwa program Anda bebas dari kesalahan logis. Kemungkinan untuk menghapus semua kesalahan sintaks dalam suatu program dan masih belum menjalankannya. Bahkan jika itu berjalan, itu mungkin masih tidak melakukan apa yang Anda maksudkan. Sebagai contoh :

$2 + 3 * 5$ dan $(2 + 3) * 5$

Algoritma membuat teh manis

1. Mulai
2. Siapkan gelas, sendok, teh celup, gula, dan air hangat
3. Masukkan teh dan air hangat ke dalam gelas
4. Tambahkan gula
5. Aduk hingga rata
6. Jika kurang manis tambahkan gula
7. Apabila rasanya telah pas teh manis siap disajikan
8. Selesai



Pemrograman membuat teh manis

Kasus bukan algoritma :

Jika ingin makan makanan pedas, maka jangan tambahkan cabai pada makanannya.