

Alih Kontrol dengan Flowchart

Pada contoh-contoh pertemuan 1, flowchart (diagram alur) mengalir lurus dari atas ke bawah.

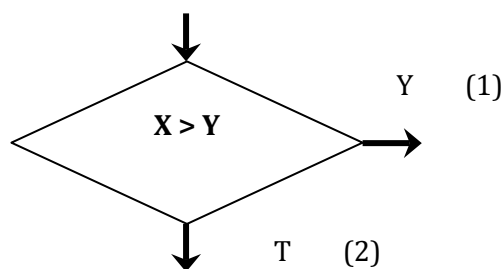
Flowchart demikian biasanya untuk masalah-masalah sederhana.

Untuk masalah yang rumit, pada flowchart banyak terjadi alih kontrol berupa *percabangan (branching)* dan *pemutaran kembali (looping)*.

1. Percabangan (branching)

Percabangan terjadi apabila kita dihadapkan pada kondisi dengan 2 pilihan : BENAR atau SALAH.

Dalam flowchart digunakan simbol *decision (keputusan)*.

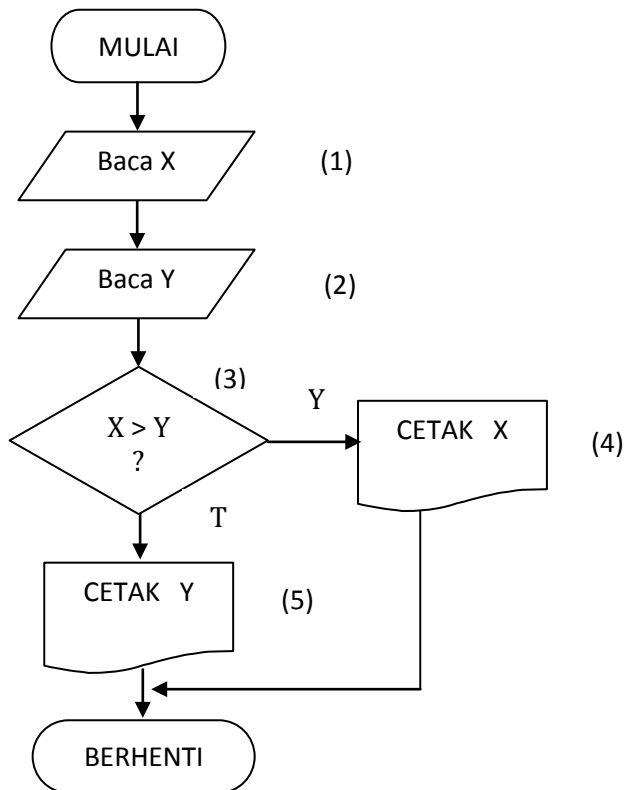


Apabila $X > Y$, kondisi benar, arus mengikuti alur (1) sedangkan bila kondisi salah ($X < Y$) maka arus mengikuti alur (2).

Contoh :

1. Diketahui 2 buah bilangan X dan Y yang di baca dari keyboard.

Akan dibuat flowchart untuk mencetak bilangan yang terbesar diantara ke-2 bilangan tersebut.

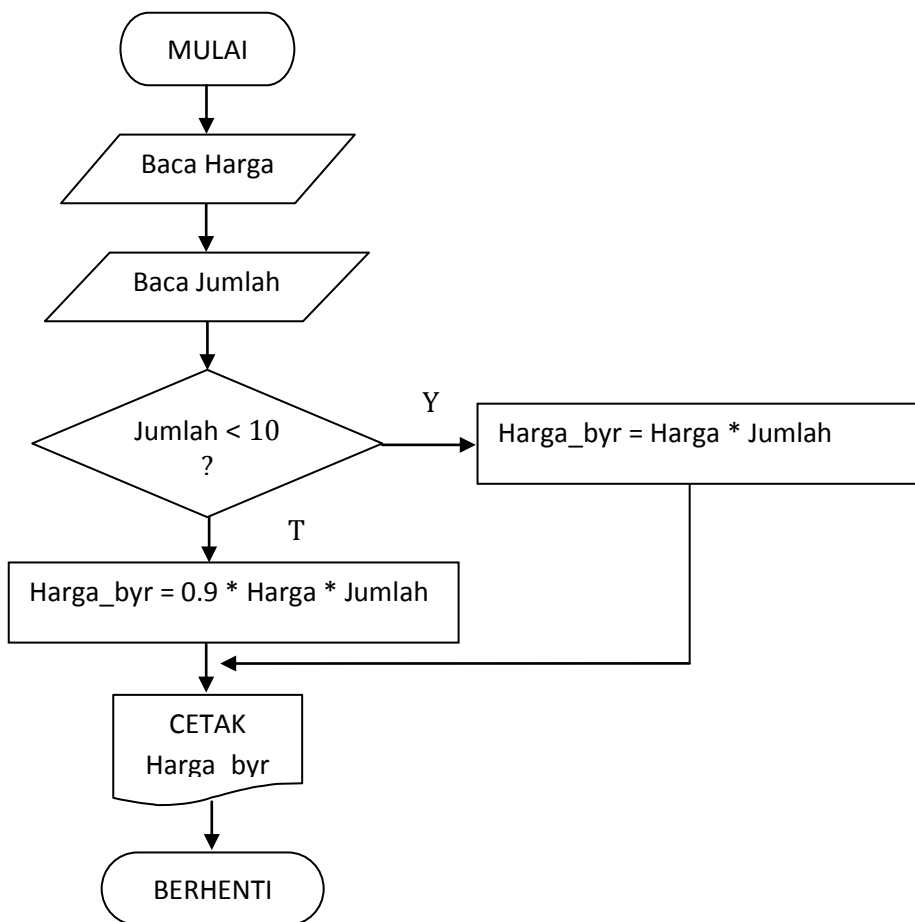


Keterangan :

- (1) Masukan = 11 dibaca, dan diberikan kepada variable X
Harga variable X sekarang = 11
- (2) Masukan = 21 dibaca, dan diberikan kepada variable Y
Harga variable Y sekarang = 21
- (3) Karena harga X (=11) tidak lebih besar dari harga Y (=21), maka dilaksanakan (5) yaitu mencetak Y.
Jadi yang tercetak adalah 21.

2. Kalau kita membeli buku dalam jumlah tertentu, penerbit akan memberikan diskon dengan ketentuan sbb :

- **Pembelian kurang dari 10 buku tidak diberikan diskon.**
- **Pembelian diatas 10 buku diberikan diskon 10%.**
- **Harga buku dan jumlah buku dibaca.**
- **Cetak harga pembayaran buku**

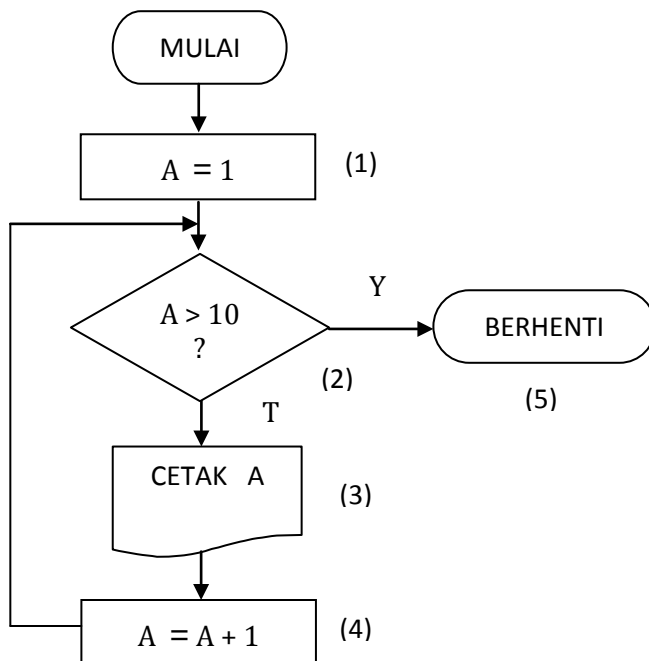


2. Pemutaran kembali (looping)

Pemutaran kembali terjadi ketika mengalihkan arus diagram alur kembali ke atas, sehingga beberapa alur berulang kembali beberapa kali.

Contoh :

Akan dibuat flowchart untuk mencetak deret bilangan
1, 2, 3, 10



Keterangan :

- (1) Variabel A diberi harga 1
- (2) Karena harga A = 1 tidak lebih besar 10, kondisi salah, arus menuju (3)
- (3) Mencetak harga A (tercetak 1)
- (4) Harga A bertambah 1 menjadi 2
Kembali menuju ke (2), dan seterusnya sampai harga A = 11 atau menuju (5), berhenti.

Structured English

- **Structure english → salah satu cara penyajian algoritma dengan cara tulisan.**
- **Alat yang cukup efisien (sederhana) untuk menggambarkan suatu algoritma.**
- **Basis dari structure english → bahasa Inggris (dapat dalam bahasa Indonesia).**
- **Dasar penggambaran algoritma → bahasa manusia**
- **Structure English → menggambarkan suatu algoritma yang akan dikomunikasikan kepada pemakai system (user).**

Gaya Penulisan Structured English

1. *Common Style*

(Menggunakan huruf besar di awal dan selanjutnya huruf kecil)

2. *Capitalized Common Style*

(menggunakan huruf besar semua)

3. *Outline Common Style*

(dengan menggunakan nomor urut)

4. *Gaya lain*

(tiap kata kunci ditulis dengan huruf besar semua)

Contoh 1 :

Gaya Common Style (huruf besar diawal) :

Menghitung luas persegi panjang

- Baca data panjang dan lebar
- Hitung luas sama dengan Panjang dikalikan Lebar
- Cetak luas persegi panjang

Contoh 2 :

Outline Common Style (menggunakan nomor urut) :

Menentukan kelulusan

1. Baca nilai test
2. Jika nilai test ≥ 60 maka cetak "LULUS", jika lebih kecil dari 60 maka cetak "GAGAL"

Pseudocode

- **Pseudocode → salah satu cara penyajian algoritma dengan cara tulisan.**
- **Pseudo artinya imitasi/mirip/menyerupai), sedangkan code artinya program.**
- **Pseudocode → Kode yang mirip dengan kode pemrograman yang sebenarnya.**
- **Berbasis bahasa pemrograman seperti : BASIC, PASCAL, atau C++.**
- **Lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan kepada programmer.**
- **Lebih rinci dari structure English, karena pada Pseudocode menyatakan tipe data yang digunakan.**

Aturan Penulisan Pseudocode

Algoritma Pseudocode terdiri dari 3 bagian, yaitu :

- **Judul**
- **Deklarasi**
- **Deskripsi**

Judul

Untuk mendefinisikan nama algoritma, dengan menentukan apakah algoritma tersebut adalah program, prosedur, atau fungsi.

Deklarasi

Deklarasi adalah bagian teks algoritma sebagai tempat untuk mendefinisikan :

- Nama konstanta
- Nama variabel
- Nama type
- Nama fungsi
- Nama prosedur

Semua nama tersebut, baru dapat dipakai di dalam algoritma/program jika telah didefinisikan terlebih dahulu didalam deklarasi.

Deskripsi

Algoritma adalah bagian inti dari suatu algoritma. Komponen teks algoritma dalam pemrograman procedural dapat berupa :

- Instruksi dasar seperti input/output, assignment
- Perhitungan rumus
- Instruksi yang beruntun
- Percabangan (kondisi)
- Perulangan

Perbedaan penulisan pernyataan algoritma

Contoh 1:

Structured English	Pseudocode BASIC	Pseudocode PASCAL
<ul style="list-style-type: none"> - Masukkan panjang - Masukkan lebar - Luas adalah panjang dikalikan lebar - Cetak luas 	Input panjang Input lebar Luas = panjang * lebar Print Luas	Read panjang Read lebar Luas = panjang * lebar Write Luas

Contoh 2:

Structured English	Pseudocode BASIC	Pseudocode PASCAL
<ul style="list-style-type: none"> - Masukkan X - Masukkan Y - Jika nilai X lebih besar dari Y, maka cetak X, jika nilai X lebih kecil dari Y, maka cetak Y 	Input X Input Y If X > Y then Print X else Print Y	Read X Read Y If X > Y then Write X else Write Y

--	--	--

Aturan penulisan Algoritma Pseudocode

Pseudocode

Contoh 1:

Algoritma Luas persegi panjang → **Judul**
// menghitung luas persegi panjang → **Keterangan Algoritma**

Deklarasi
 panjang, lebar, luas as integer → **Deklarasi**

Deskripsi → **Deskripsi**
 Input panjang
 Input lebar
 luas = panjang * lebar
 Print luas

Contoh 2:

Algoritma Max → **Judul**
// Menentukan bilangan terbesar → **Keterangan Algoritma**

Deklarasi
 X, Y as integer → **Deklarasi**

Deskripsi
 Input X
 Input Y
 If X > Y then
 Print X
 else
 Print Y

Pseudocode

Contoh 1:

Algoritma Hitung diskon
// menghitung jumlah harga

→ **Judul**
→ **Keterangan Algoritma**

Deklarasi
 hrg, jml, jml_hrg : real;

→ **Deklarasi**

Deskripsi
 Read (hrg);
 Read (jml);
 jml_hrg = hrg * jml;
 Write (jml_hrg);

→ **Deskripsi**

Contoh 2:

Algoritma Max
// Menentukan bilangan terbesar

→ **Judul**
→ **Keterangan Algoritma**

Deklarasi
 X, Y : real;

→ **Deklarasi**

Deskripsi
 Read X
 Read Y
 If X > Y then
 Write X
 else
 Write Y

→ **Deskripsi**

Ciri-ciri Algoritma

Menurut Donald E kneth dalam bukunya yang berjudul “The Art Computer Programming”, algoritma harus memiliki ciri - ciri penting, yaitu:

- a. **Input**, algoritma harus memiliki data input.
- b. **Output**, algoritma harus menghasilkan keluaran yaitu besaran yang memiliki hubungan dengan masukan setelah melalui proses.
- c. **Finiteness**, algoritma harus berhenti atau memiliki akhir.
- d. **Definiteness**, setiap langkah harus terdefinisi dengan baik dan tepat.
- e. **Effectiveness**, setiap langkah harus berdaya guna, artinya susunan algoritma harus sependek mungkin, sehingga dapat dikerjakan dalam sejumlah waktu yang masuk akal.

Soal latihan :

- 1. Buatlah algoritma pseudocode untuk mencari luas segitiga.
Rumus luas segitiga = $\text{alas} \times \text{tinggi} / 2$.
Alas dan tinggi dibaca, luas segitiga dicetak.**

- 1. Buatlah algoritma pseudocode untuk menentukan nilai kelulusan. Bila nilai yang dibaca sama dengan atau lebih besar dari 60 maka cetak "Lulus", apabila nilai dibawah 60 cetak "Tidak Lulus".**

