

Pseudocode, Flowchart & Pemilihan

Team Teaching Dasar Pemrograman
2024

Tujuan Pembelajaran

- Mahasiswa memahami dan mampu menjelaskan tentang Pseudo-code
- Mahasiswa menjabarkan dan mampu menjelaskan tentang flowchart dan symbol-simbolnya
- Mahasiswa memahami dan mampu menggunakan pseudocode dan flowchart untuk menyelesaikan studi kasus
- Mahasiswa mampu menjelaskan tentang konsep pemilihan sederhana, serta mampu menuliskan algoritma pemilihan untuk menyelesaikan studi kasus dengan menggunakan flowchart

PSEUDOCODE

Pseudocode

- Pseudocode adalah deskripsi tingkat tinggi dari algoritma pemrograman komputer dengan **menggunakan notasi** yang **mirip** dengan **kode pemrograman**, tetapi tidak terikat oleh aturan baku dalam suatu bahasa pemrograman tertentu.
- Pseudocode digunakan untuk menjelaskan langkah-langkah penyelesaian suatu masalah secara jelas dan mudah dipahami oleh manusia, bukan oleh mesin.
- Ditulis dalam bahasa Inggris atau Indonesia sederhana
- Sintaks digunakan untuk menjelaskan **perintah** (misalnya: “jika”, “ulangi”, “sampai”, “if”, “repeat”, “until”)

Menulis Pseudocode

Sintaks dikelompokkan dalam 6 operasi dasar berikut ini :

1. Menerima data (Input)
2. Menampilkan data (Output)
3. Menghitung data (Compute)
4. Menyimpan data (Store)
5. Membandingkan data (Compare)
6. Melakukan pengulangan (Loop)

1. Menerima data (Input)

- Sewaktu komputer menerima data atau *input*, maka perintah yang biasa digunakan adalah “**Read**”, “**Get**”, atau “**Baca**”
- Contoh:
 - Read Bilangan
 - Get kode_pajak
 - Baca namaMahasiswa

2. Menampilkan data (Output)

- Sewaktu komputer menampilkan data ataupun *output*, maka perintah yang biasa digunakan adalah **“Print”**, **“Write”**, **“Put”**, **“Output”**, **“Display”** atau **“Cetak”**
- Contoh:
 - Print “Politeknik Negeri Malang”
 - Cetak “Metode Perancangan Program”
 - Output Total

3. Menghitung data (Compute)

- Untuk melakukan operasi aritmetika digunakan :
 - + Untuk penjumlahan (add)
 - Untuk pengurangan (subtract)
 - * Untuk perkalian (multiply)
 - / Untuk pembagian (divide)
 - () Untuk kurung
- Statement **“Compute”**, **“Calculate”** ataupun **“Hitung”** juga dapat digunakan.
- Contoh:
$$\text{Compute } X = (1+2) * 3$$
$$\text{Total} = \text{Total} + 10$$

4. Menyimpan data (Store)

- Ada tiga cara untuk menyimpan data :
 - Memberikan **nilai awal**, menggunakan *statement* “**Initialize**” atau “**Set**”
 - Memberikan **nilai sebagai hasil dari suatu proses**, maka tanda “=” digunakan
 - Untuk **menyimpan suatu nilai** maka *statement* “**Save**” atau “**Store**” digunakan
- Contoh:
 - Set Umur to 17
 - Harga = 1000
 - Store 50 to Discount

5. Membandingkan data (Compare)

- Salah satu operasi terpenting yang dapat dilakukan komputer adalah membandingkan dan memilih salah satu alternatif solusi.
- Perintah yang digunakan : **“IF”, “THEN” dan “ELSE” (atau “JIKA”, “MAKA”, “SELAIN ITU”)**
- Contoh
IF Pilih=1 THEN
 Discount = 0.1 * harga
ELSE
 Discount = 0.2 * harga
ENDIF

6. Melakukan pengulangan (Loop)

- Jika ada beberapa perintah yang harus diulang, maka dapat digunakan perintah **“DO WHILE”** dan **“END DO”**.

- Contoh

Bil = 0

DO WHILE bil < 10

 cetak bil

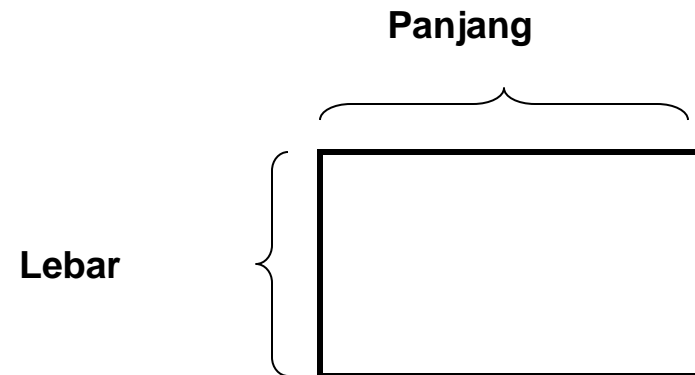
 bil = bil +1

END DO

Contoh

Buatlah algoritma menggunakan pseudocode untuk menghitung luas persegi panjang, dengan rumus :

$$\text{Luas} = \text{Panjang} * \text{Lebar}$$



Jawaban - Pseudocode

Algoritma: `hitung_luas_persegi`

{dibaca Panjang dan lebar dari piranti masukan. Hitunglah luas persegi Panjang tersebut}

Deklarasi :

`panjang, lebar, luas: int`

Deskripsi :

1. `print "Input & Hitung Luas Persegi Panjang"`
2. `print "Masukkan Panjang = "`
3. `read panjang;`
4. `print "Masukkan Lebar = "`
5. `read lebar`
6. `luas = panjang * lebar`
7. `print "Luas Persegi Panjang = "`
8. `print luas`

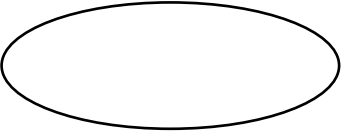
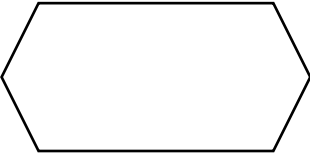


FLOWCHART




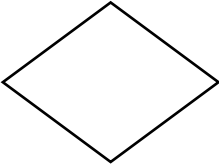
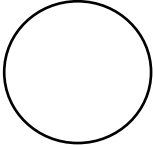
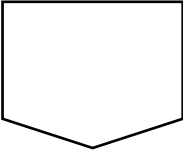

Flowchart

- Flowchart merupakan sebuah bagan dengan **simbol-simbol** tertentu yang digunakan untuk **menjelaskan urutan proses** dan hubungan antar proses lainnya pada **sebuah program**.
- Merupakan cara penyajian dari suatu algoritma.
- Ada 2 macam Flowchart :
 1. System Flowchart → urutan proses dalam system dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.
 2. Program Flowchart → urutan instruksi yang digambarkan dengan symbol tertentu untuk memecahkan masalah dalam suatu program.

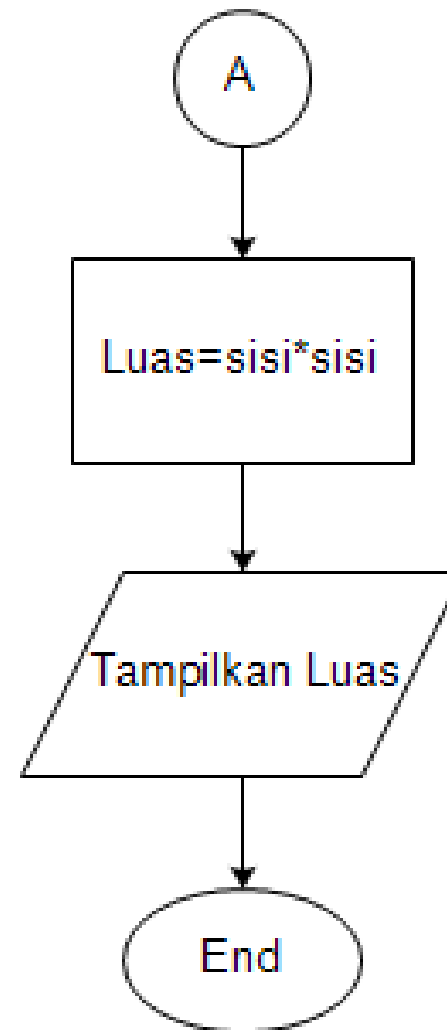
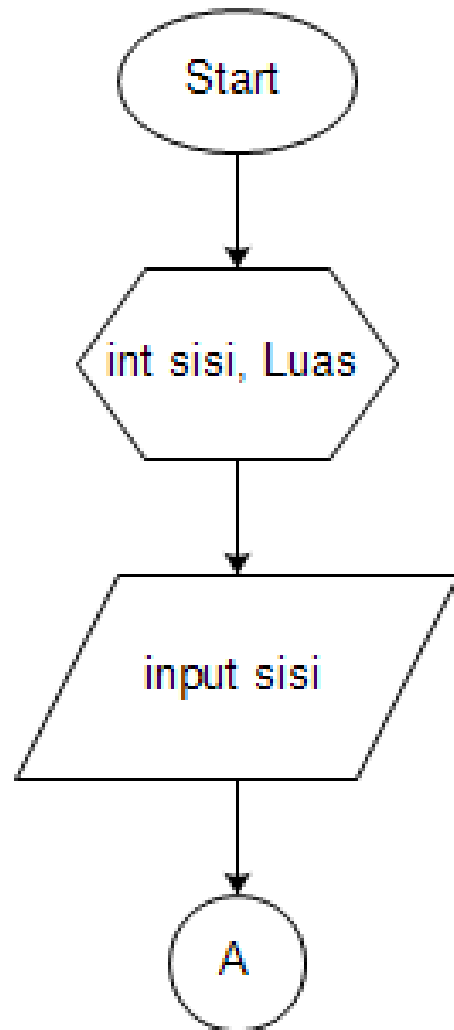
Symbol-symbol Flowchart

Symbol	Nama	Deskripsi
	Terminator	Simbol untuk permulaan (start) dan akhir (end) dalam sebuah proses
	Preparation	Simbol untuk mempersiapkan penyimpanan yang akan digunakan
	Input-output	Simbol yang menyatakan proses input dan output
	Flow Line (Garis Alir)	Arah aliran program

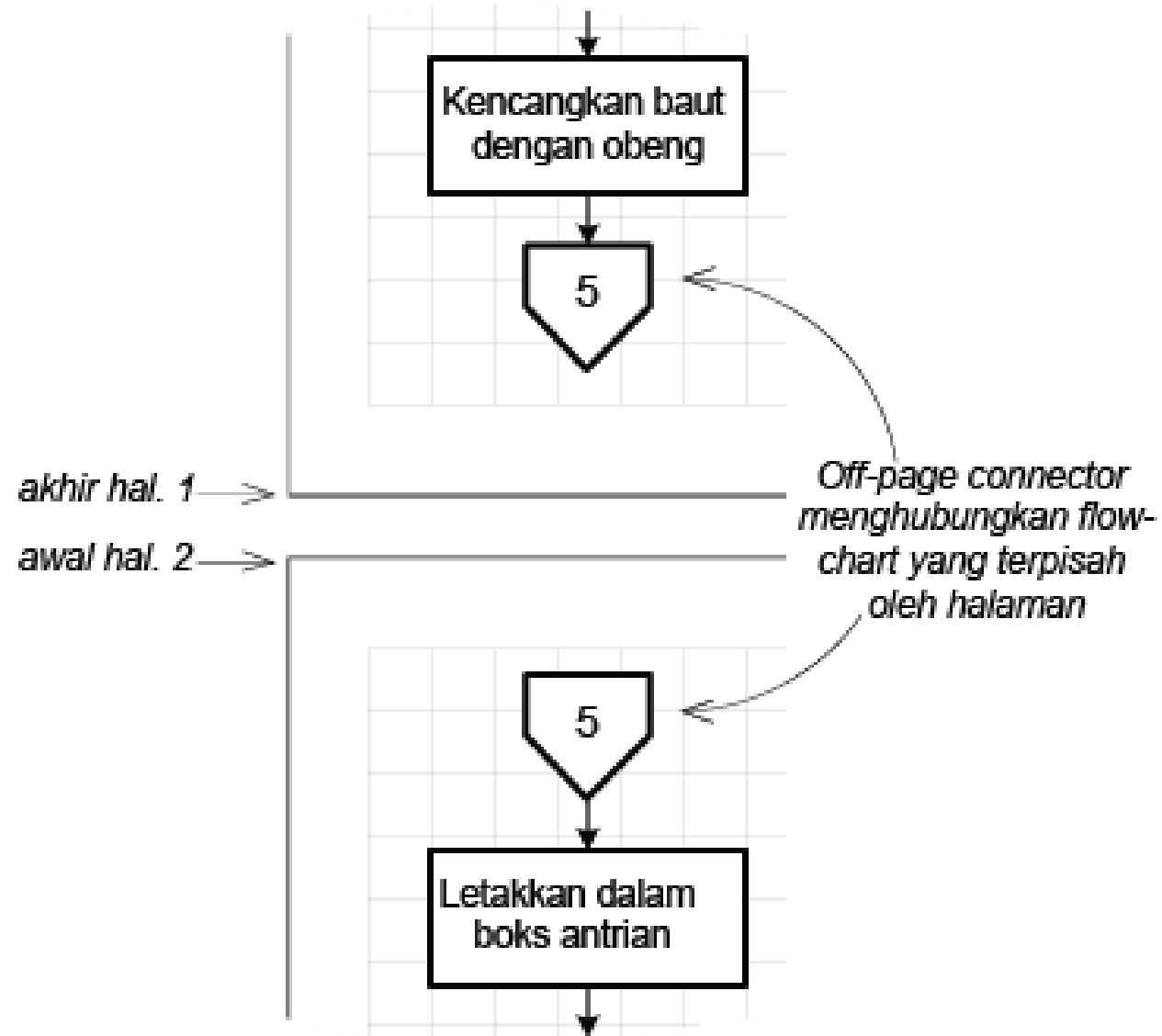
Symbol-symbol Flowchart

Symbol	Nama	Deskripsi
	Processing simbol	Simbol yang menunjukkan pemrosesan oleh komputer
	Percabangan	Simbol yang menunjukan terdapat dua pilihan output
	On Page Connector	Penghubung bagian-bagian flowchart yang ada pada satu halaman
	Off Page Connector	Penghubung bagian-bagian flowchart yang berada pada halaman yang berbeda
	Predefined Process	Simbol untuk pelaksanaan suatu bagian (sub-program)/fungsi/prosedur

Contoh Penggunaan On Page Connector



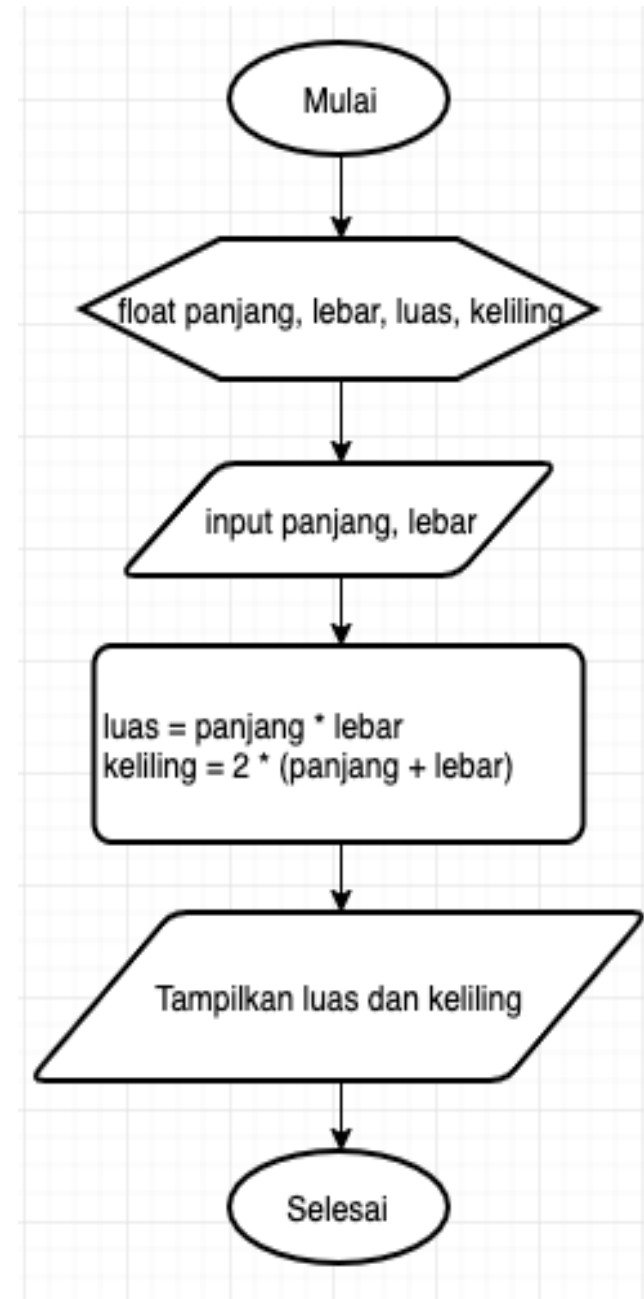
Contoh Penggunaan Off Page Connector



Contoh Flowchart

Contoh Studi Kasus:

Pak Adi mempunyai sebuah kebun berbentuk persegi panjang, buatlah flowchart untuk menghitung luas dan keliling kebun pak Adi!



Contoh Studi Kasus-Pertemuan 3

Pak Adi mempunyai sebuah kebun berbentuk persegi Panjang. Pak Adi ingin membuat pagar kayu untuk mengelilingi kebun tersebut.

Sebelum membuat program untuk membantu pak adi menghitung keliling kebunnya, maka bantulah pak adi untuk mengidentifikasi variable dan tipe data beserta algoritmanya!

Contoh Studi Kasus

1. Menentukan Algoritma

Input: panjang, lebar

Output: keliling

Proses:

1. input panjang, lebar
2. $keliling = 2 \times (\text{panjang} + \text{lebar})$
3. Output keliling

2. Mengidentifikasi variable dan jenis tipe data berdasarkan algoritma

Variabel	Tipe data
panjang	int
lebar	int
keliling	int

PSEUDOCODE

Algoritma : `hitung_keliling_persegi`
{dibaca Panjang dan lebar dari piranti masukan. Hitunglah keliling persegi}!

Deklarasi :

`panjang, lebar, keliling: int`

Deskripsi :

1. `read panjang`
2. `read lebar`
3. `keliling = 2* (panjang + lebar)`
4. `print keliling`

Contoh Studi Kasus

1. Menentukan Algoritma

Input: panjang, lebar

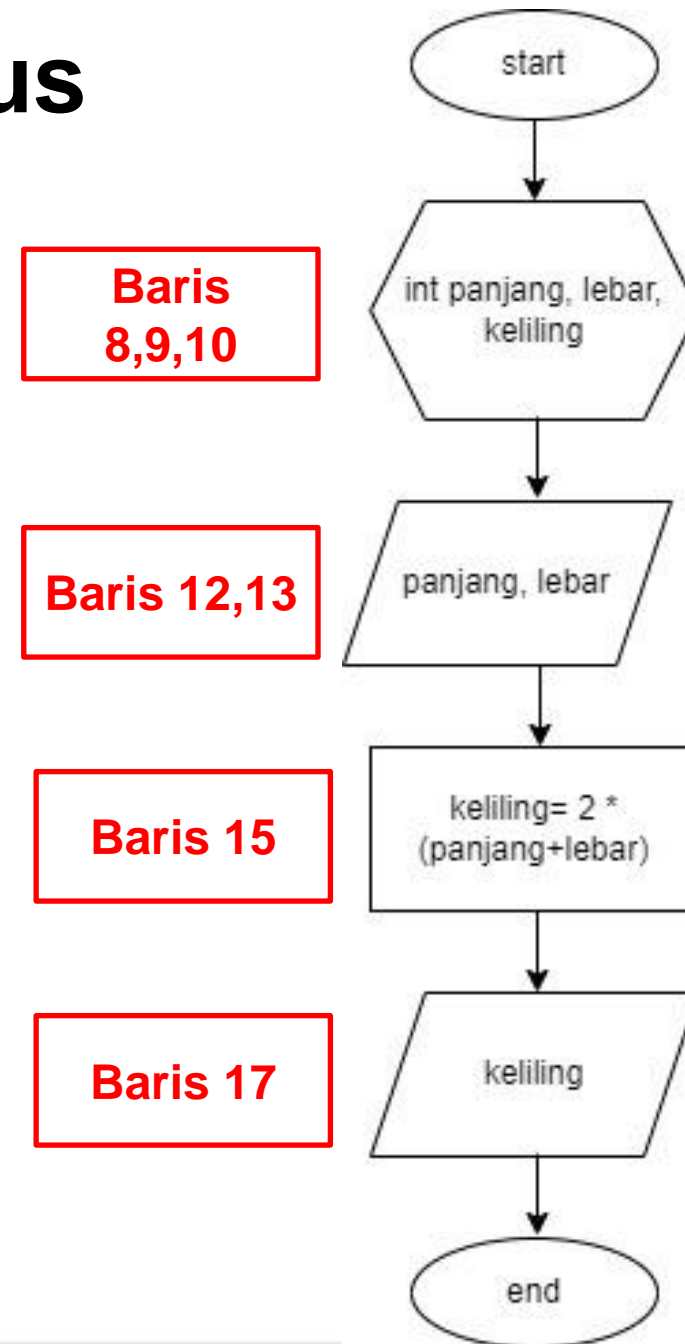
Output: keliling

Proses:

1. input panjang, lebar
2. $keliling = 2 \times (\text{panjang} + \text{lebar})$
3. Output keliling

2. Mengidentifikasi variable dan jenis tipe data berdasarkan algoritma

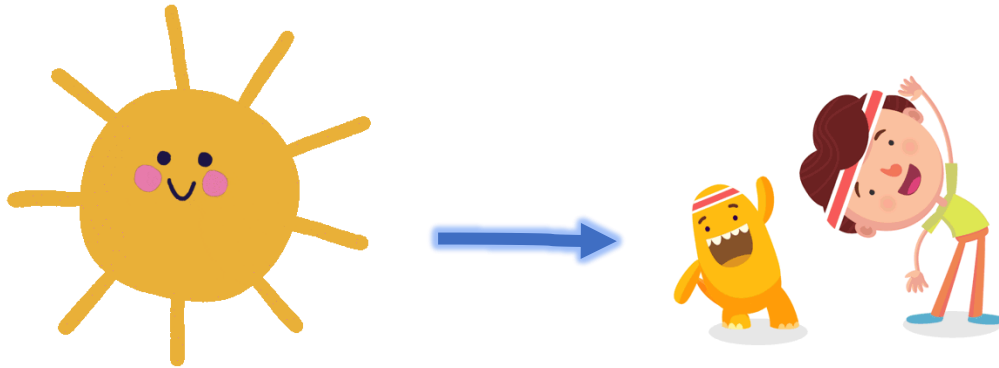
Variabel	Tipe data
panjang	int
lebar	int
keliling	int



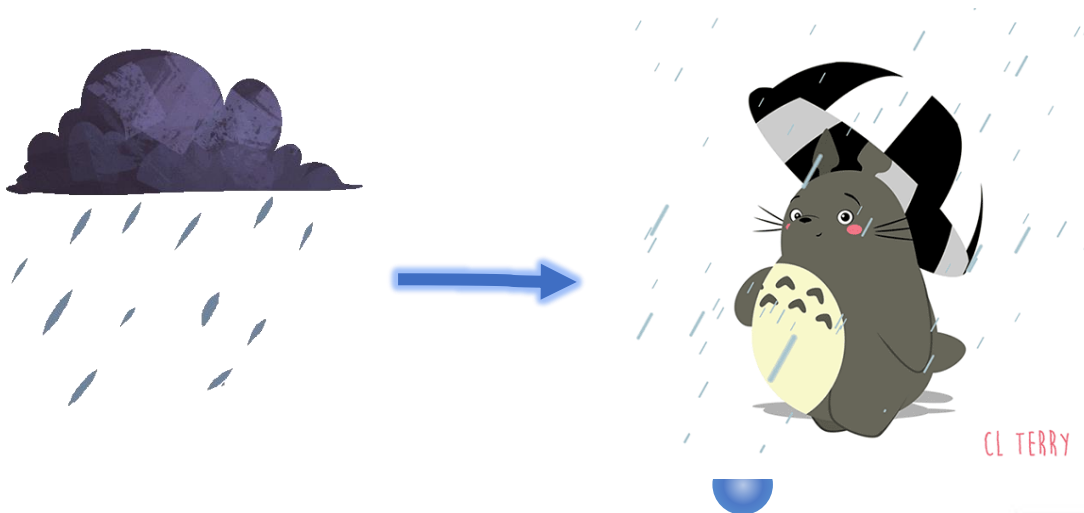
PEMILIHAN

PEMILIHAN

Aktivitas pemilihan dalam kehidupan sehari - hari



Jika Cerah
Saya akan olah raga di luar rumah

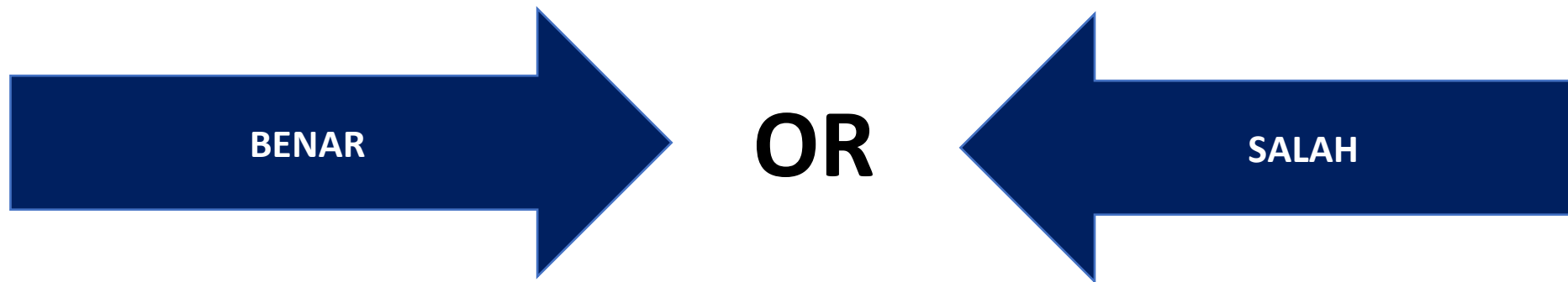


Jika Hujan
Saya akan membawa payung

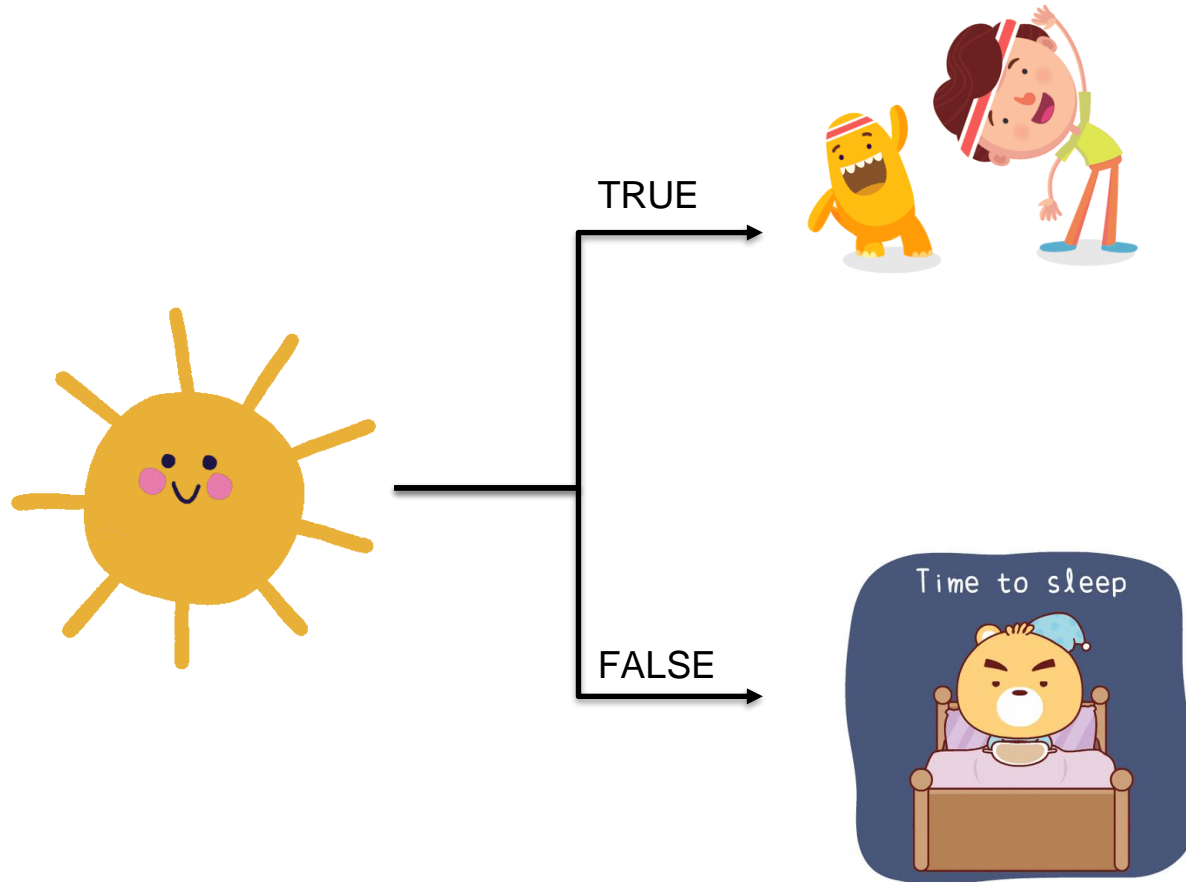
Definisi

- Pemilihan (selection) adalah instruksi untuk yang dipakai untuk menentukan satu pilihan berdasarkan suatu kondisi

Kondisi: suatu pernyataan atau ekspresi (pernyataan logika)



PEMILIHAN → KONDISI



PEMILIHAN → KONDISI

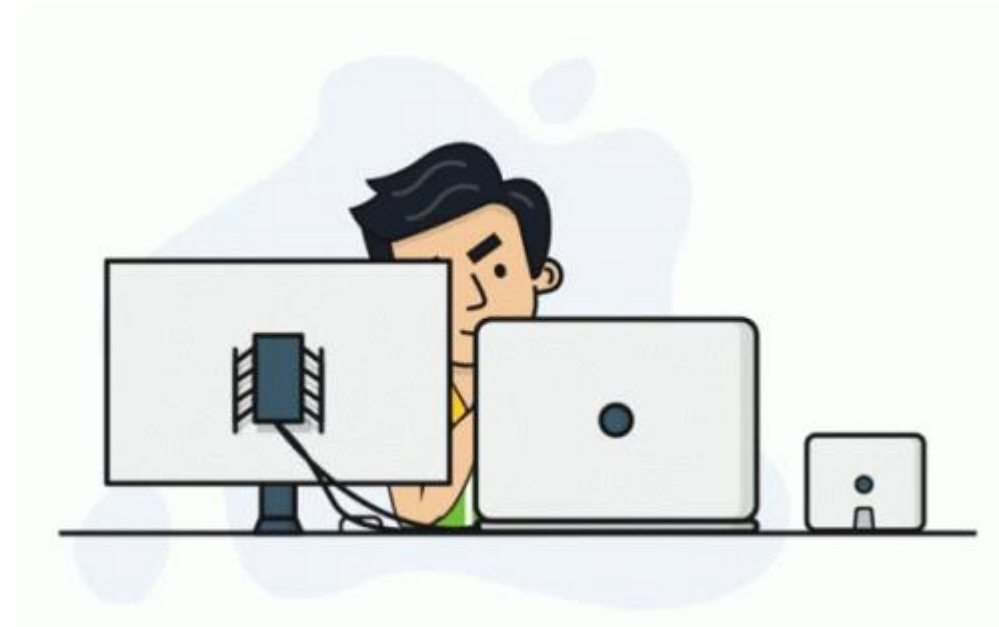
Seperti manusia, program komputer tidak hanya dieksekusi secara sekuensial, tetapi juga berdasarkan kondisi-kondisi tertentu



username mmh

password *****

Sukses login / gagal

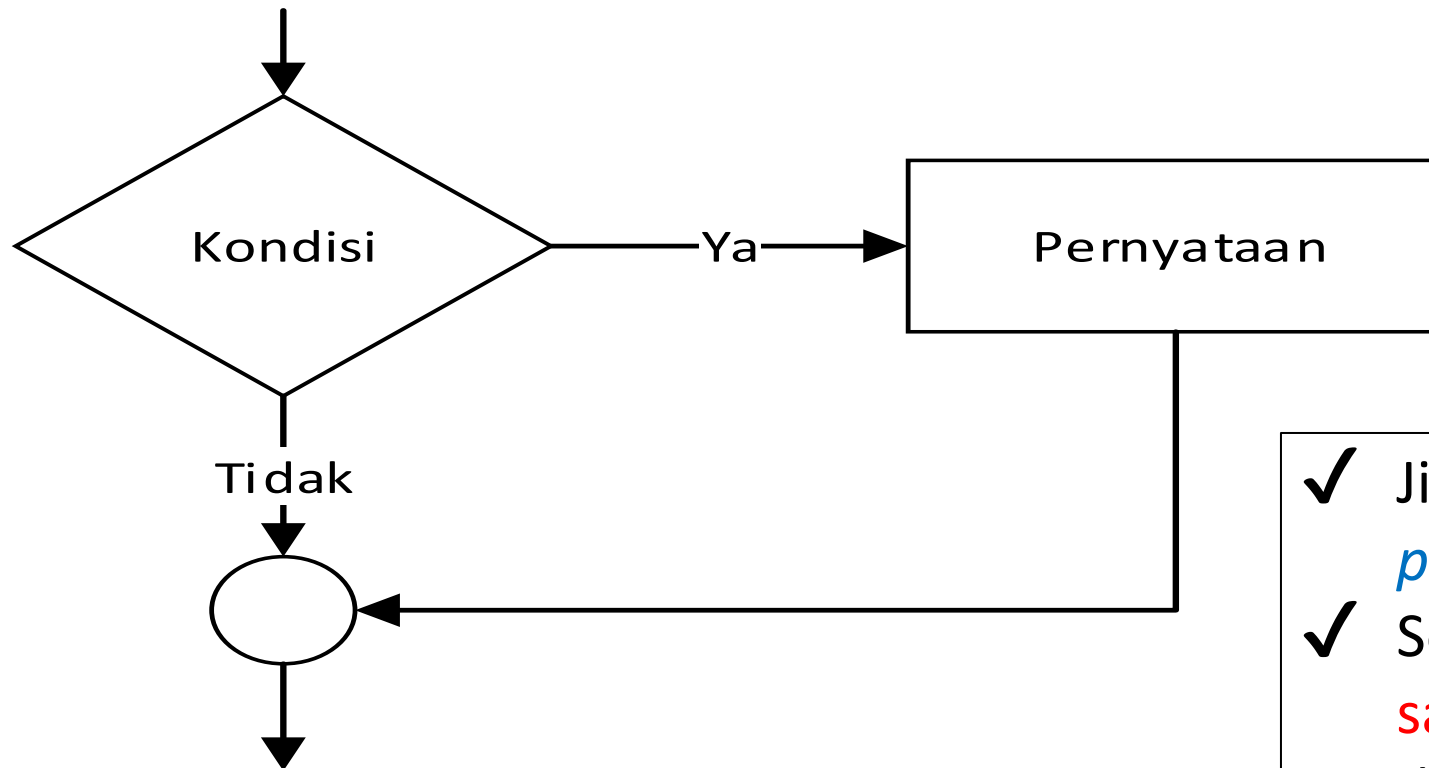


BENTUK SINTAKS PEMILIHAN di JAVA

- 
- 1. IF**
 - 2. IF...ELSE**
 - 3. IF...ELSE IF...ELSE...**
 - 4. SWITCH...CASE**

Struktur Pemilihan IF....

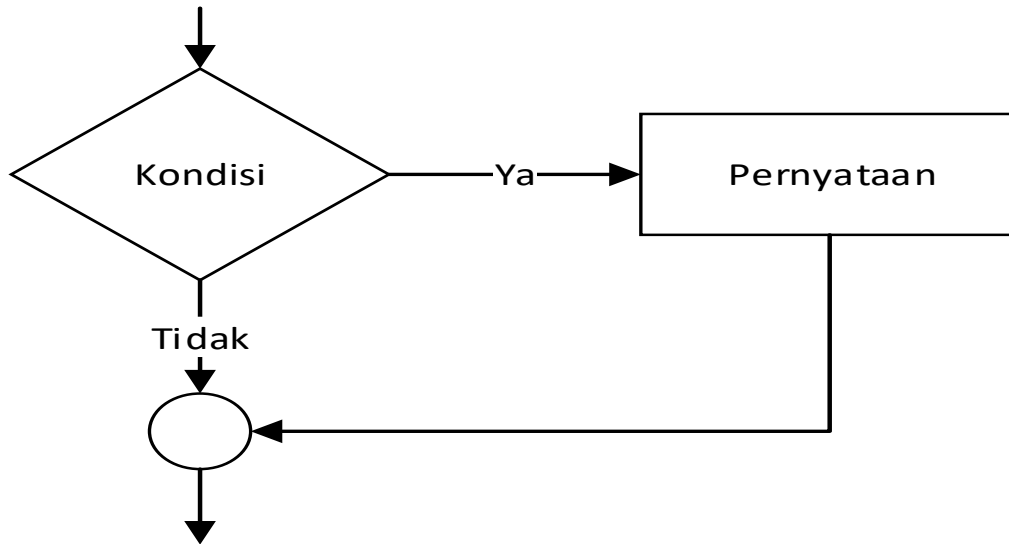
Flowchart If



- ✓ Jika *kondisi* bernilai *benar*, maka *pernyataan* akan dieksekusi.
- ✓ Sebaliknya, jika *kondisi* bernilai *salah*, maka *pernyataan* tidak akan dieksekusi.

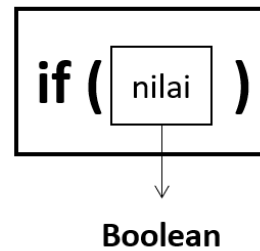
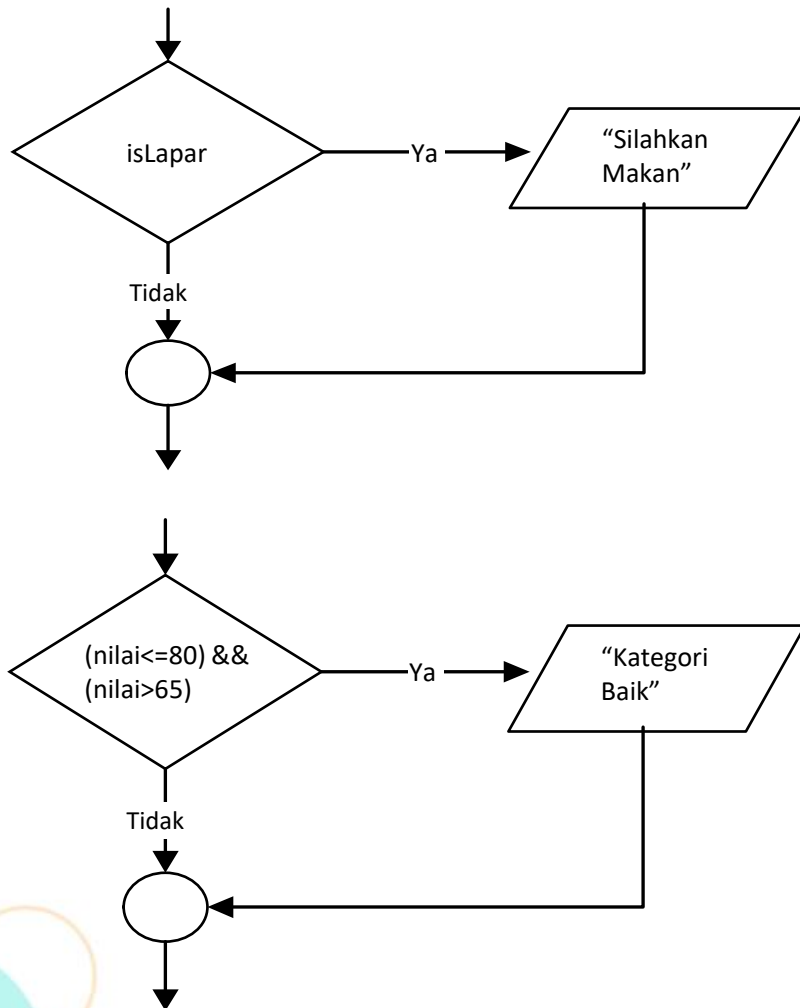
Sintaks Pemilihan IF

- Bentuk Umum:

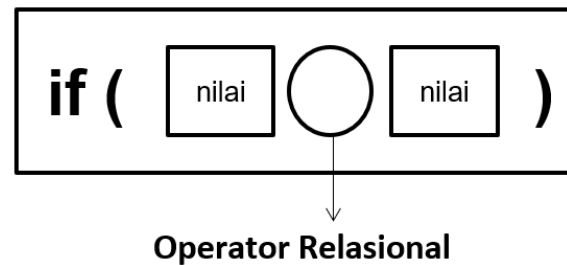


```
if (Kondisi)
{
    Pernyataan;
}
```


Sintaks Pemilihan IF



```
if (isLapar) {  
    Print("Silakan Makan")  
}
```



```
if (nilai <= 80) && (nilai > 65) {  
    Print("Kategori Baik")  
}
```



Contoh Kode Program

```
import java.util.Scanner;

public class PemilihanDenganIf {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Masukkan suhu: ");
        int suhu = sc.nextInt();

        if (suhu < 16) {
            System.out.println(x: "Silakan memakai jaket");
        }
    }
}
```

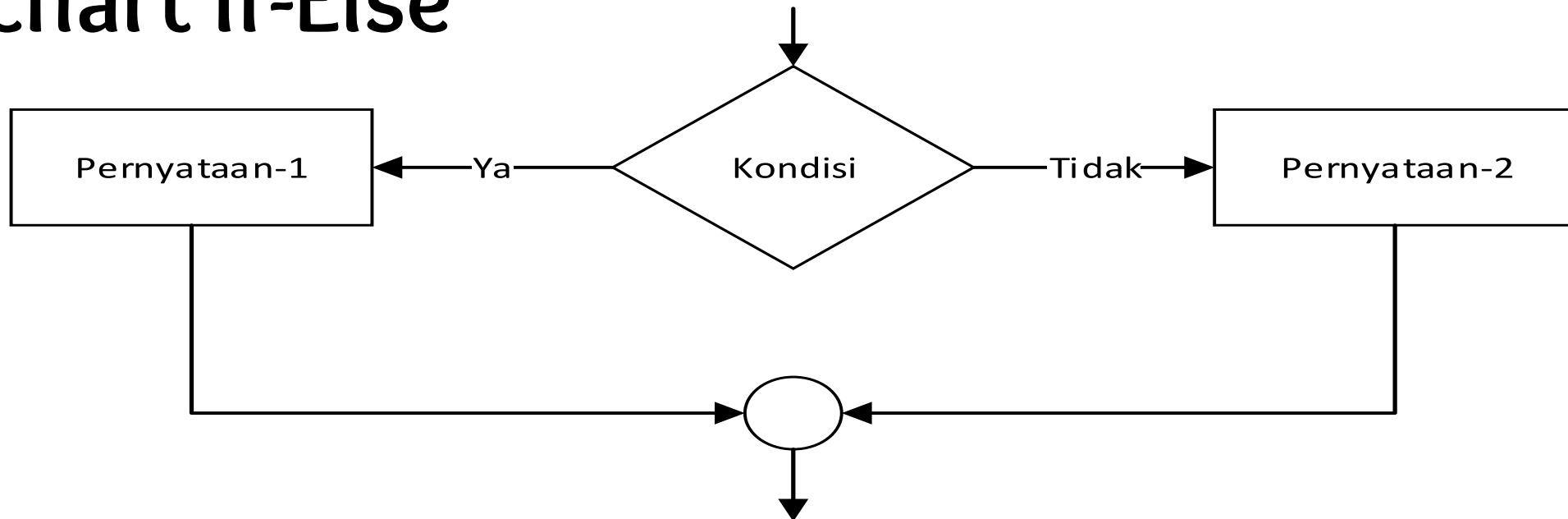
Masukkan suhu: 15

"Silakan memakai jaket"

Masukkan suhu: 17

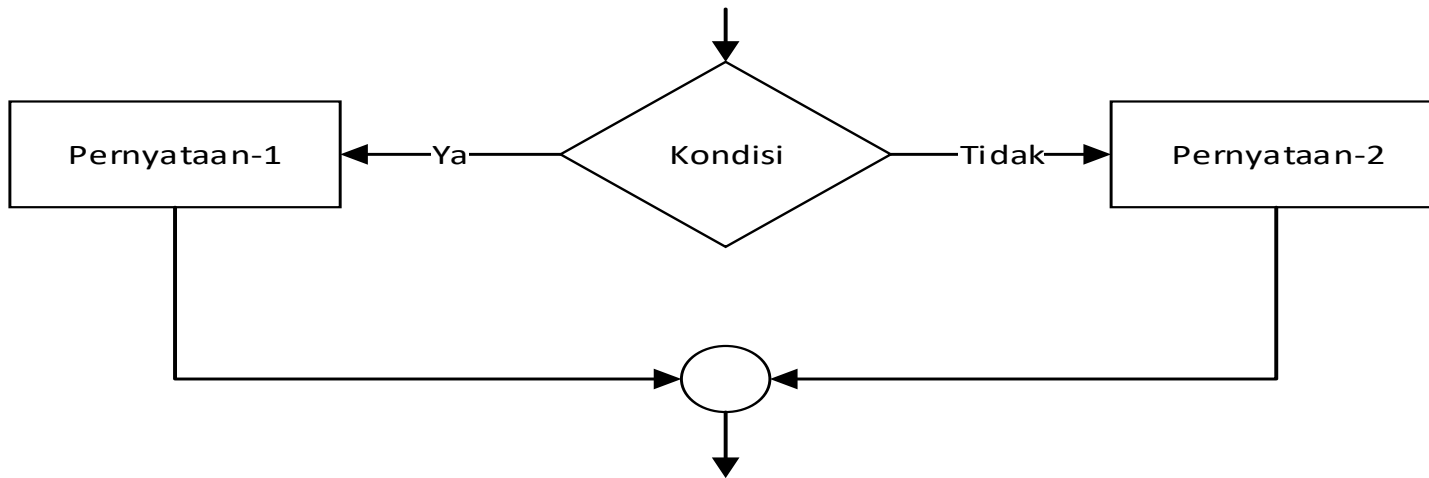
Struktur Pemilihan IF....ELSE

Flowchart If-Else



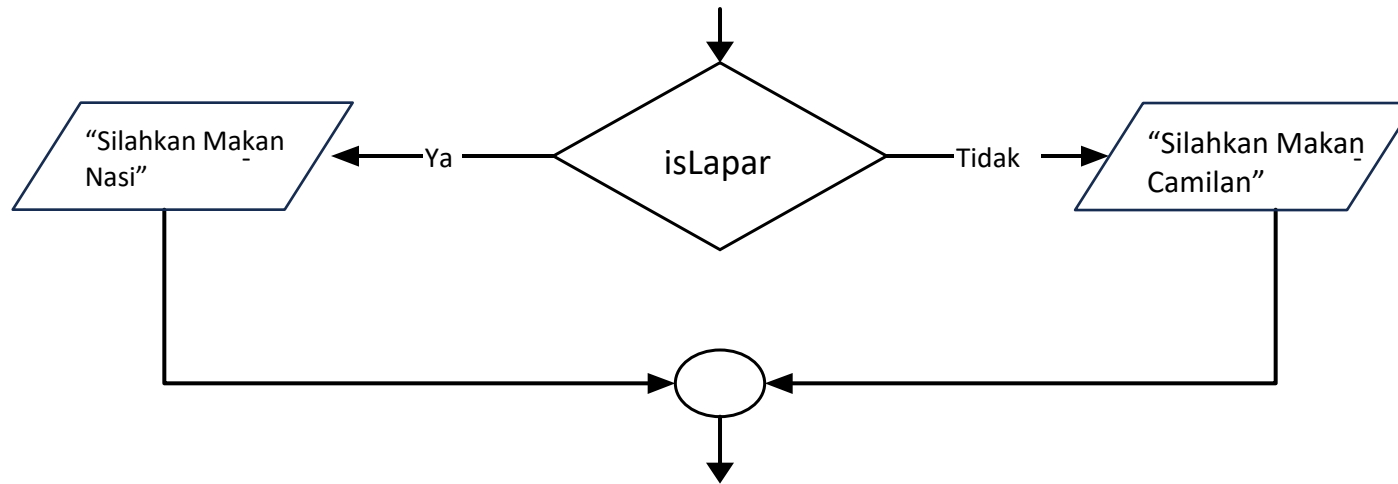
Struktur pemilihan **IF-ELSE** memiliki **2 pernyataan**.
Jika kondisi bernilai TRUE atau terpenuhi, maka **Pernyataan-1** akan dieksekusi.
Namun, jika kondisi bernilai FALSE, maka **Pernyataan-2** yang akan dieksekusi.

Struktur Pemilihan IF...ELSE



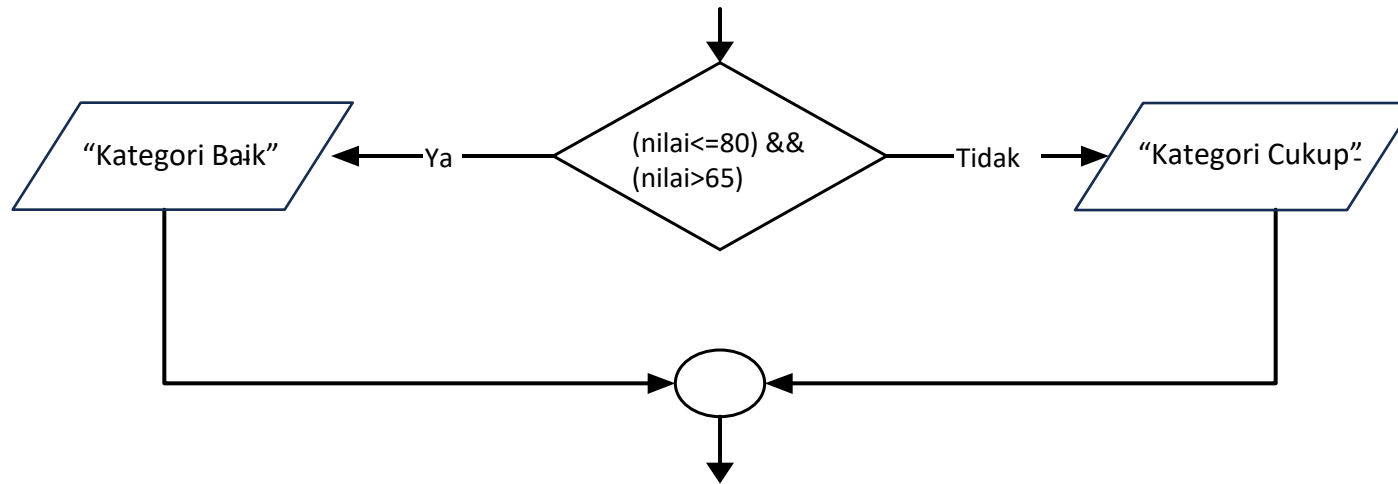
```
if (Kondisi)
{
    Pernyataan-1;
}
else
{
    Pernyataan-2;
}
```

Struktur Pemilihan IF...ELSE



```
if (isLapar) {  
    Print("Silakan Makan Nasi")  
}  
else{  
    Print("Silahkan Makan  
    Camilan")  
}
```

Struktur Pemilihan IF...ELSE



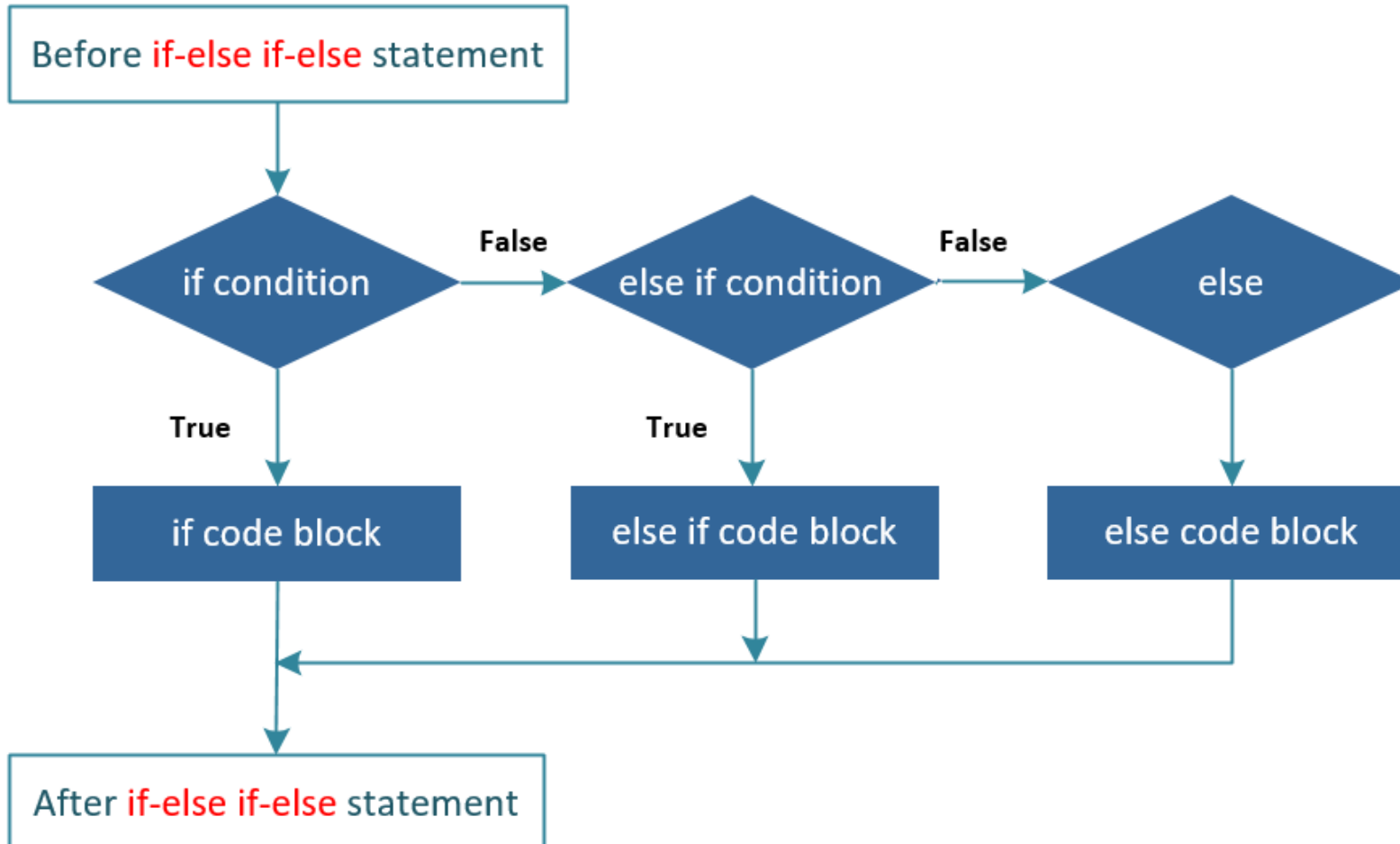
```
if (nilai <= 80) && (nilai > 65) {  
    Print("Kategori Baik")  
}  
else {  
    Print("Kategori Cukup")  
}
```

Struktur Pemilihan

IF....ELSE

IF...ELSE...

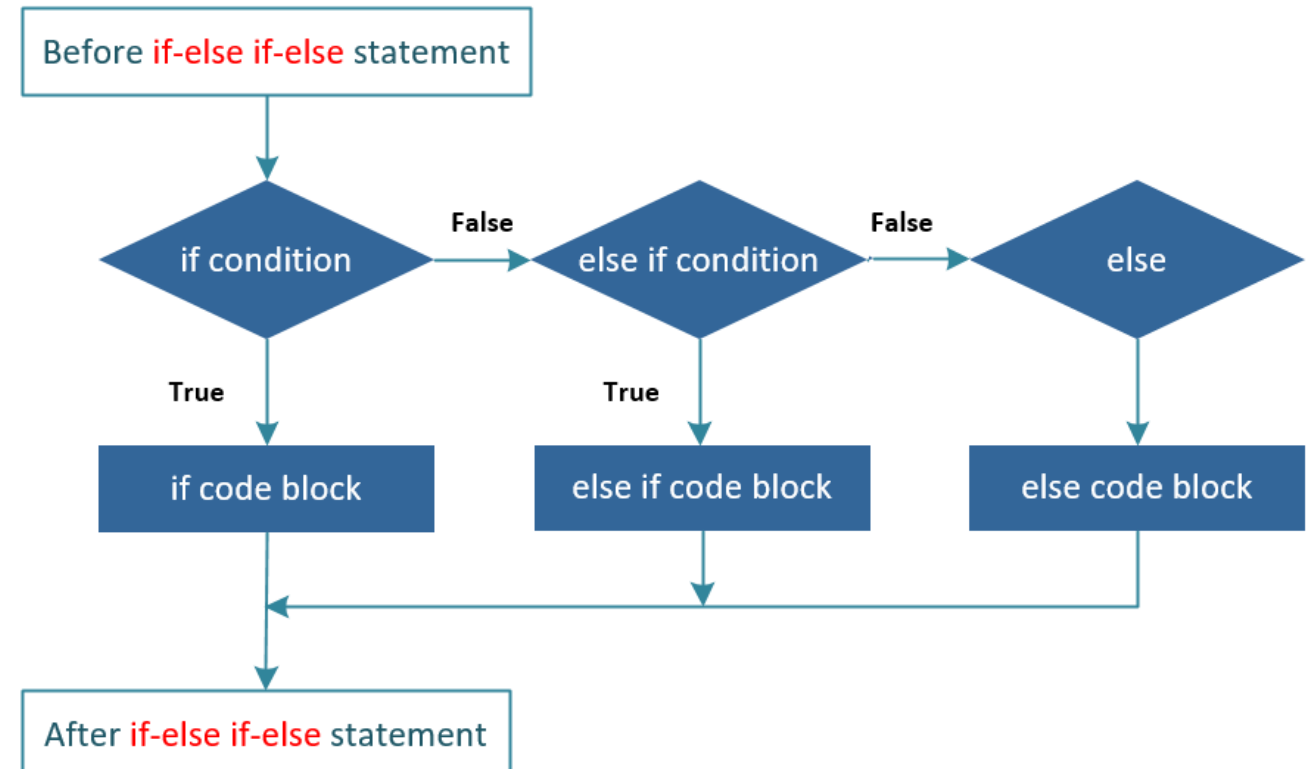
Flowchart



- **pernyataan 1** akan dieksekusi apabila "**kondisi 1**" bernilai **BENAR**.
- Jika "**kondisi 1**" bernilai **SALAH**, maka "**kondisi 2**" akan dicek. Jika "**kondisi 2**" **BENAR** maka pernyataan-2 dieksekusi
- Demikian seterusnya.
- Jika **tidak ada satupun kondisi yang terpenuhi**, maka pernyataan-X akan dieksekusi.

Pemilihan If...else if...else

```
if (kondisi-1)
{
    pernyataan-1;
}
else if (kondisi-2)
{
    pernyataan-2;
}
else if (kondisi-3)
{
    pernyataan-3;
}
....
....
else
{
    pernyataan-x;
}
```



Struktur Pemilihan SWITCH-CASE

Pemilihan SWITCH-CASE

- Bentuk Umum:

```
switch (Kondisi)
{
    case Konstanta-1:
        Pernyataan-1;
        break;
    case Konstanta-2:
        Pernyataan-2;
        break;
    ...
    ...
    case Konstanta-n:
        Pernyataan-n;
        break;
    default:
        Pernyataan;
}
```

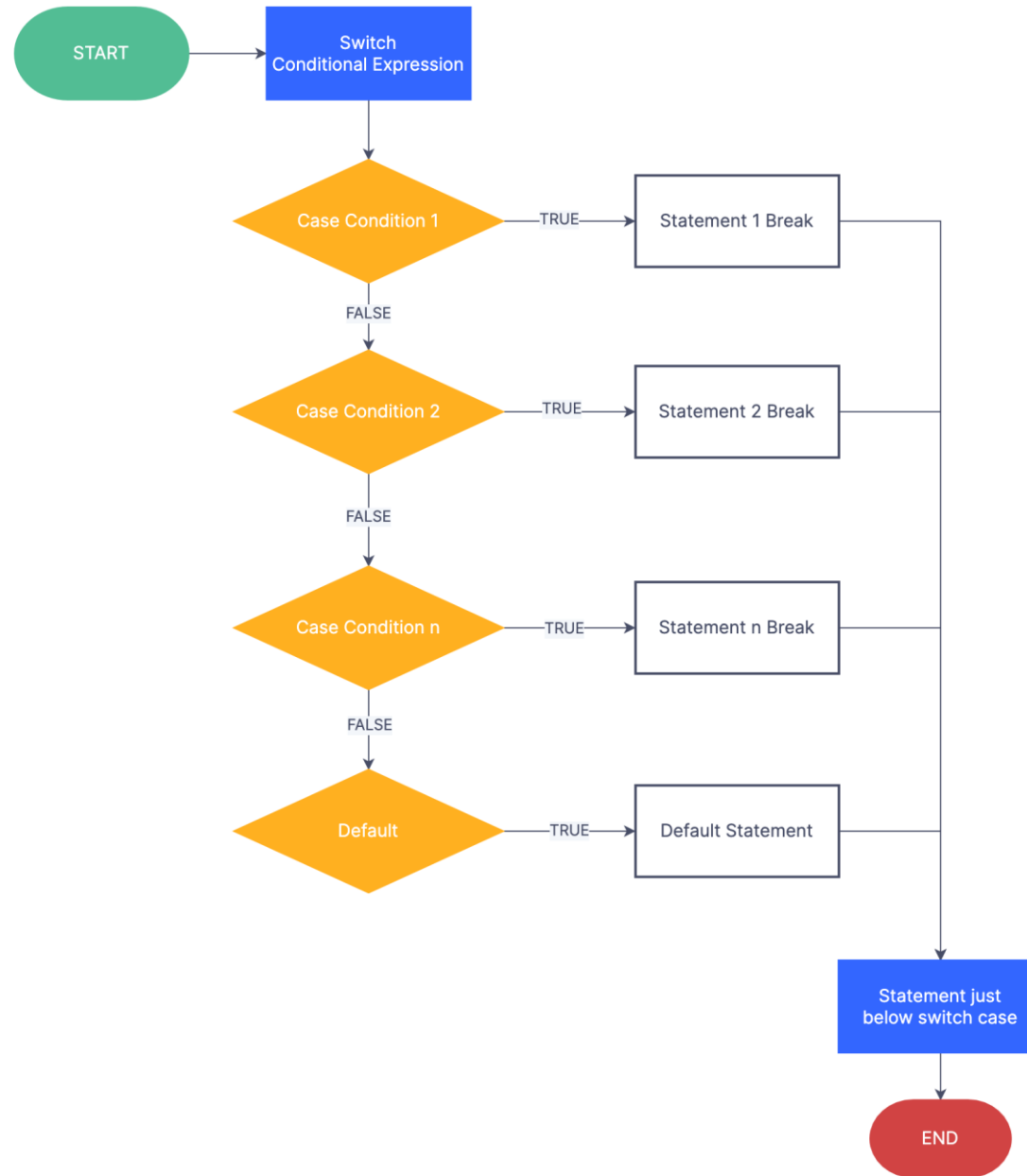
Pemilihan SWITCH-CASE

- Digunakan ketika mempunyai banyak kemungkinan tindakan yang harus dilakukan pada kondisi yang berbeda-beda.
- Sintaks pemilihan ini akan menjalankan salah satu dari beberapa pernyataan “case” sesuai dengan nilai kondisi yang ada di dalam “switch”.
- Selanjutnya proses akan dilanjutkan sampai ditemukan pernyataan “break”. Namun, jika tidak ada nilai pada case yang sesuai dengan nilai kondisi, maka proses akan dilanjutkan ke pernyataan yang ada di dalam “default”.



Pemilihan Switch-Case

Switch Case Flowchart





Contoh

```
System.out.print(s:"Masukkan angka: ");
int angka = sc.nextInt();

switch (angka) {
    case 1:
        System.out.println(x:"Hari Senin");
        break;
    case 2:
        System.out.println(x:"Hari Selasa");
        break;
    case 3:
        System.out.println(x:"Hari Rabu");
        break;
    case 4:
        System.out.println(x:"Hari Kamis");
        break;
    case 5:
        System.out.println(x:"Hari Jum'at");
        break;
    case 6:
        System.out.println(x:"Hari Sabtu");
        break;
    case 7:
        System.out.println(x:"Hari Minggu");
        break;
    default:
        System.out.println(x:"Maaf, angka yang Anda masukkan salah");
        break;
}
```



```
System.out.print(s:"Masukkan jenis buah: ");  
String buah = sc.nextLine();  
  
switch (buah) {  
    case "Jeruk":  
        System.out.println(x:"Harga per kg Rp 15.000");  
        break;  
    case "Nanas":  
        System.out.println(x:"Harga per kg Rp 10.000");  
        break;  
    case "Anggur":  
    case "Apel":  
        System.out.println(x:"Harga per kg Rp 30.000");  
        break;  
    default:  
        System.out.println(x:"Maaf, harga belum terdaftar");  
        break;  
}
```

Masukkan jenis buah: Anggur
Harga per kg Rp 30.000

Ternary Operators



Operator Ternary

- Digunakan sebagai bentuk lain `if ... else ...`
- Hanya bisa jika statement tunggal baik jika kondisi benar maupun salah
- Sintaks:

(kondisi) ? (jika kondisi benar) : (jika kondisi salah)

Contoh

```
public static void main(String[] args) {  
    double angka=5.5;  
  
    String hasil;  
  
    if (angka>0.0){  
        hasil="Bilangan positif";  
    }  
    else{  
        hasil="Bilangan negatif";  
    }  
    System.out.println(angka+ " adalah " + hasil);  
}
```

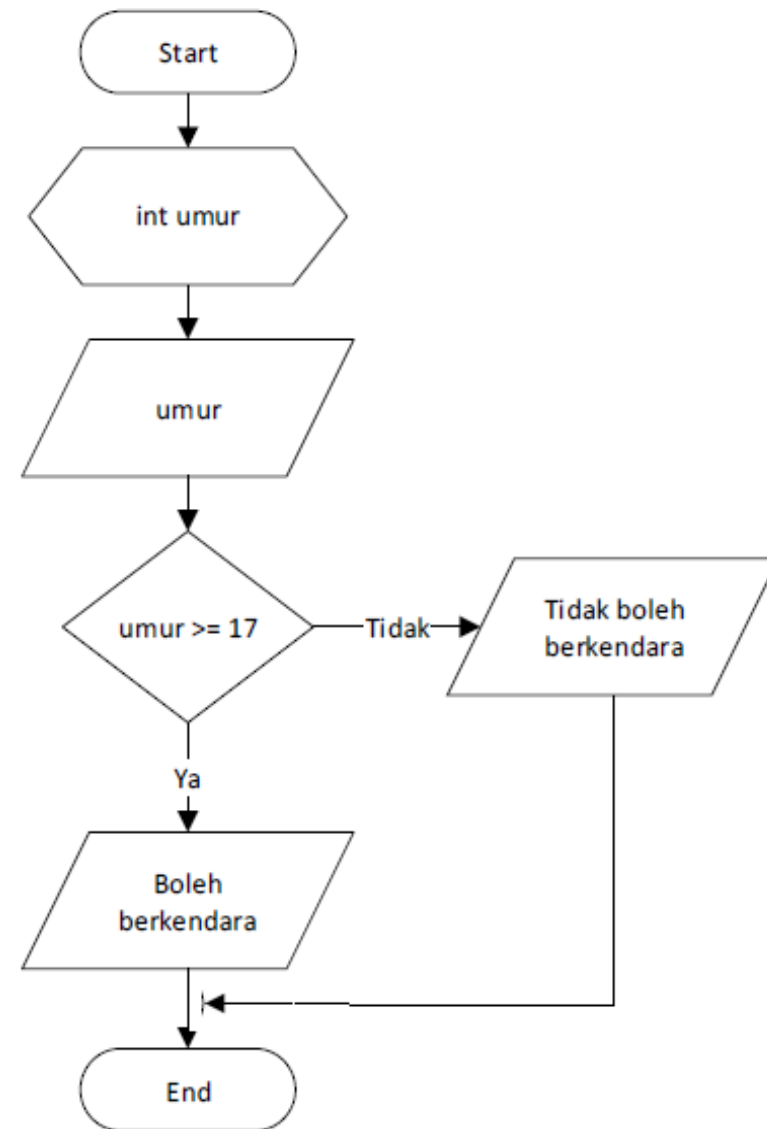
```
public static void main(String[] args) {  
  
    Double angka = 5.5;  
    String hasil;  
    hasil = (angka > 0.0) ? "Bilangan positif" : "Bilangan Negatif";  
    System.out.println(angka + " adalah " + hasil);  
}
```

Contoh Studi Kasus

LATIHAN STUDI KASUS

Di dalam aturan tata tertib berkendara kendaraan bermotor, terdapat aturan bahwa orang yang boleh mengendarai kendaraan bermotor adalah orang yang berumur minimal 17 tahun

- ✓ Tentukan kondisi : Usia > 17
- ✓ Tentukan apa yang akan dilakukan jika kondisi terpenuhi → Boleh berkendara
- ✓ Tentukan apa yang akan dilakukan jika kondisi tidak terpenuhi → Tidak boleh berkendara



```
if (usia >= 17) {  
    System.out.print("Boleh Berkendara")  
}  
else {  
    System.out.print("Tidak Boleh Berkendara")  
}
```

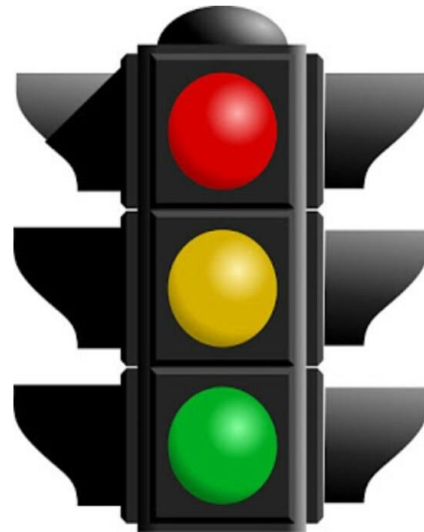
Ada Pertanyaan?

Tugas



Latihan Individu 1

Salah satu cerminan sikap warga negara yang baik adalah menaati peraturan tata tertib di jalan raya, termasuk rambu-rambu lalu lintas. Kamu adalah pengendara sepeda motor yang sedang melintas di jalan raya dan bertemu lampu lalu lintas. Buat flowchart untuk menentukan apa yang harus kamu lakukan untuk setiap kondisi lampu lalu lintas!





Latihan Individu 2

Sebuah sistem harus memastikan bahwa pengguna yang masuk ke sistem sudah terautentikasi, sehingga sistem tersebut membutuhkan fitur login. Input dari proses ini adalah username dan password. Jika username dan password sesuai dengan yang disimpan oleh sistem, maka pengguna tersebut dapat masuk ke dalam sistem. Jika user dan password tidak sesuai maka muncul peringatan “user dan password salah”. Buatlah sebuah flowchart untuk kasus tersebut

Latihan Individu 3

Buatlah sistem parker yang meminta input dari pengguna untuk menentukan jenis kendaraan yang akan diparkir. Jika pengguna memilih "mobil", program harus menghitung biaya parkir selama satu jam sebesar Rp5.000, dan jika memilih "motor", biaya parkir selama satu jam sebesar Rp3.000. Setelah menghitung biaya, tampilkan pesan yang menunjukkan jenis kendaraan dan total biaya yang harus dibayar.



Latihan Individu 4

Buatlah program sederhana yang meminta pengguna untuk memasukkan usia mereka dan menentukan kategori usia berdasarkan input tersebut. Kategorikan usia sebagai "Anak" (0-12 tahun), "Remaja" (13-19 tahun), "Dewasa" (20-64 tahun), atau "Lansia" (65 tahun ke atas). Program harus menampilkan kategori yang sesuai berdasarkan usia yang dimasukkan. Selain itu, sertakan validasi untuk memastikan bahwa input yang diberikan adalah angka positif dan berikan pesan kesalahan jika input tidak valid.

Selamat Mengerjakan

Kerja keras dan jujur merupakan modal untuk dapat menyelesaikan tugas