

Pemilihan

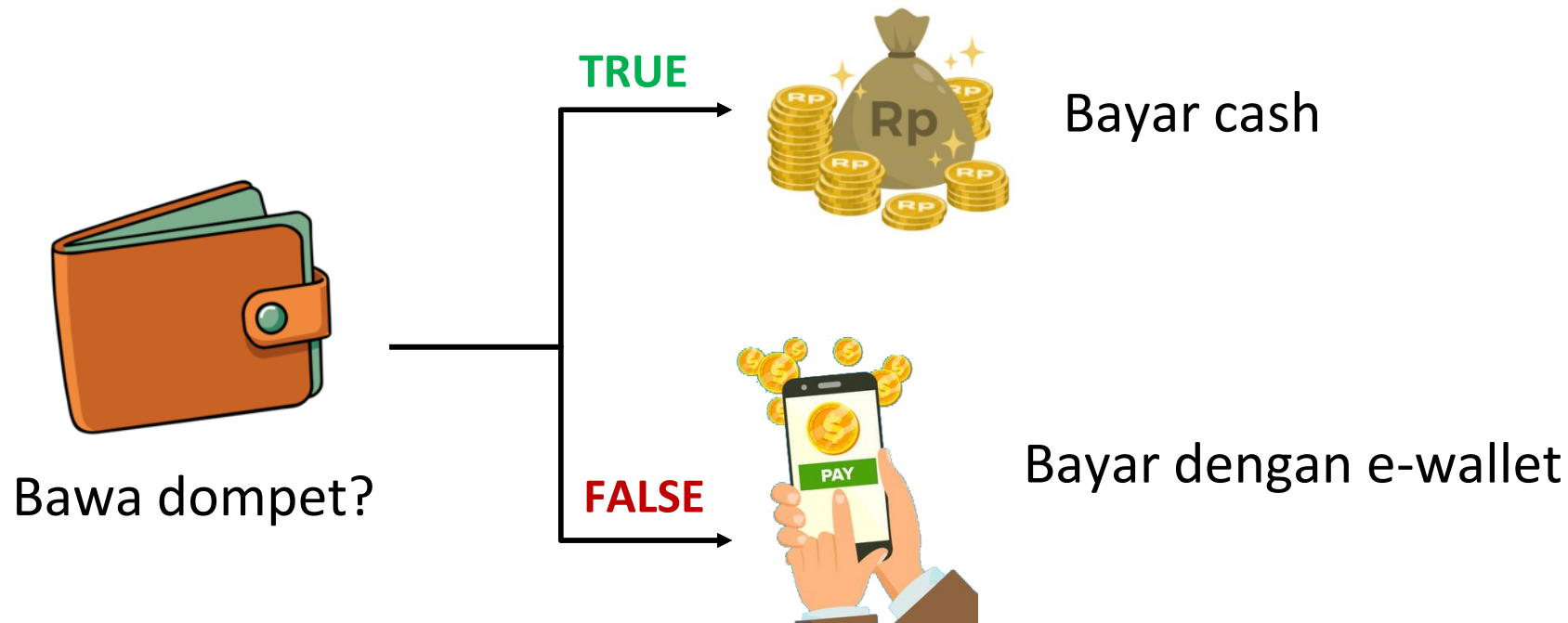
Team Teaching Dasar Pemrograman
2025

Tujuan Pembelajaran

- Mahasiswa mampu memahami struktur kontrol pemilihan
- Mahasiswa mampu menyelesaikan permasalahan sederhana menggunakan if, if-else, switch-case, dan nested if

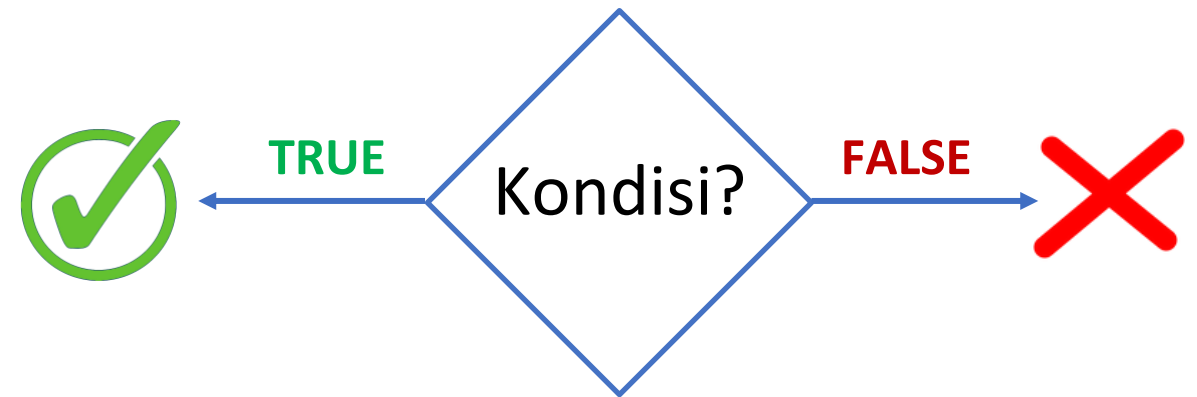
Contoh Pemilihan dalam Kehidupan

- Ketika kita akan melakukan pembayaran, keberadaan dompet adalah kondisi. Cara kita membayar (cash atau e-wallet) adalah aksi yang dipilih berdasarkan kondisi tersebut.



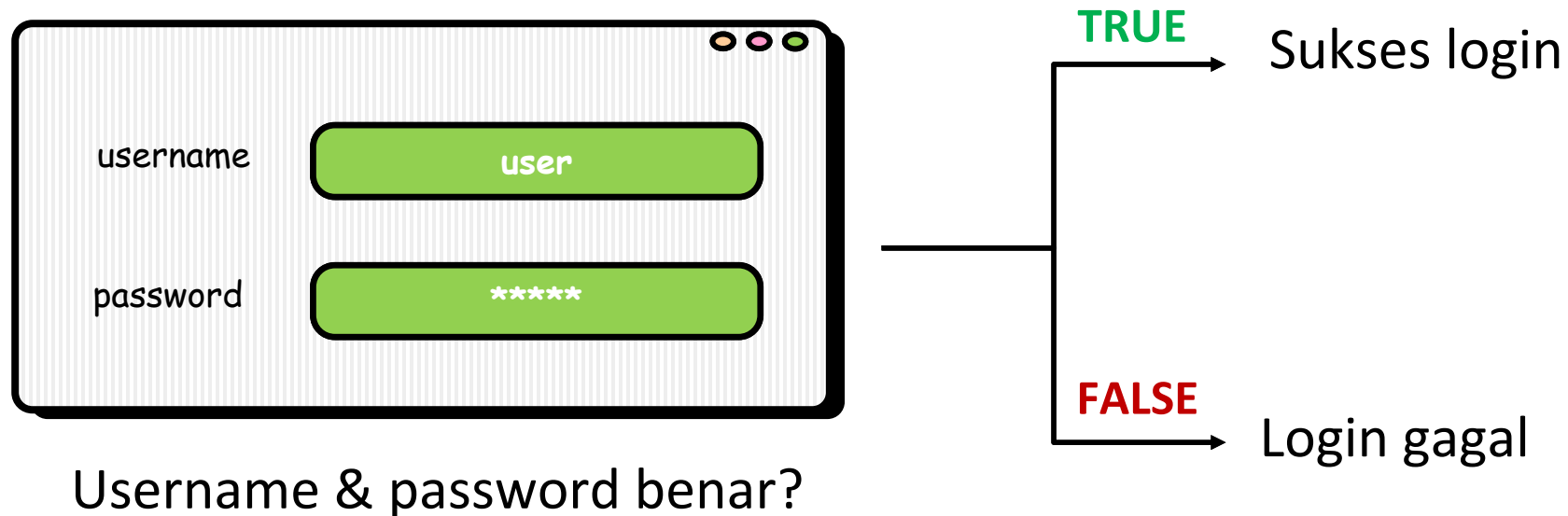
Definisi Pemilihan

- Pemilihan (*selection*) adalah instruksi untuk menentukan **aksi** berdasarkan suatu **kondisi**.
- Kondisi adalah ekspresi logika yang bernilai **BENAR (True)** atau **SALAH (False)**.



Contoh Pemilihan dalam Program

- Program tidak selalu berjalan berurutan (sekuensial).
- Program juga bisa berjalan berdasarkan **kondisi tertentu**.
- Contoh: Login → kondisi username & password



Struktur Pemilihan di Java

- Dalam kehidupan sehari-hari, kita sering mengambil keputusan berdasarkan kondisi.
- Dalam pemrograman, keputusan tersebut dituliskan dengan **struktur pemilihan**.
- Struktur pemilihan di Java terdiri dari:
 - a. IF
 - b. IF-ELSE
 - c. IF-ELSE IF-ELSE
 - d. SWITCH-CASE
 - e. Nested IF
 - f. Ternary Operator (?:)

Struktur Pemilihan IF



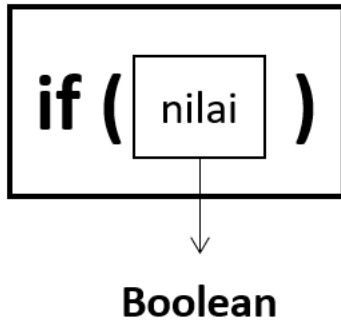
Struktur Pemilihan IF

- Sintaks pada Java:

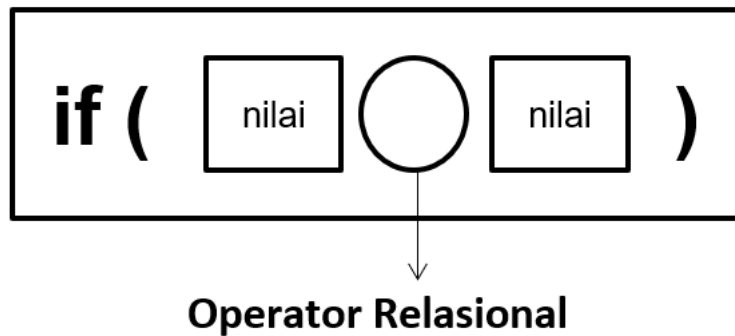
```
if (kondisi)  
{  
    Pernyataan;  
}
```

- Jika **kondisi** bernilai **BENAR**, maka **pernyataan** akan dieksekusi.
- Sebaliknya, jika **kondisi** bernilai **SALAH**, maka **pernyataan** tidak akan dieksekusi.

Contoh Penggunaan IF

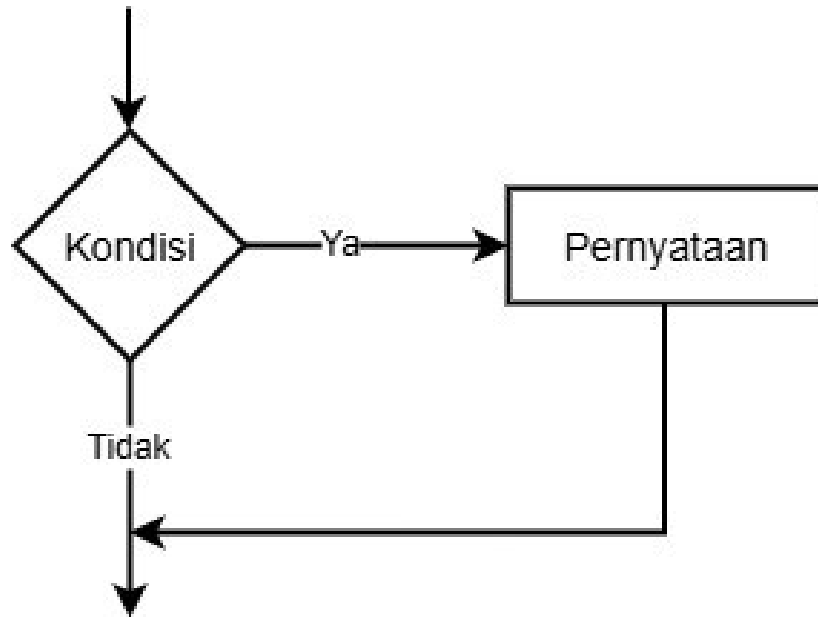


```
if (isLogin) {  
    Print("Selamat datang di SIAKAD")  
}
```



```
if (waktuPengumpulan <= deadline) {  
    Print("Tugas dikumpulkan tepat waktu")  
}
```

Flowchart dan Pseudocode IF



```
IF (kondisi) THEN  
    Pernyataan  
END IF
```

Struktur Pemilihan IF-ELSE

Struktur Pemilihan IF-ELSE

- Sintaks pada Java:

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

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

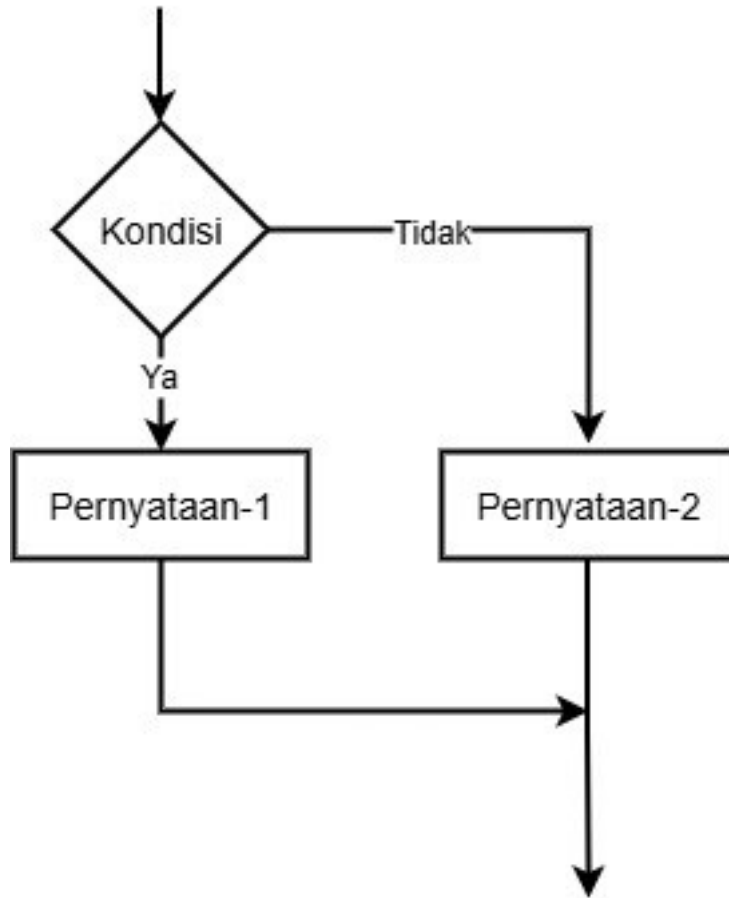


Contoh Penggunaan IF-ELSE

```
if (isLogin) {  
    Print("Selamat datang di SIAKAD")  
} else {  
    Print("Login gagal, silahkan coba lagi")  
}
```

```
if (waktuPengumpulan <= deadline) {  
    Print("Tugas dikumpulkan tepat waktu")  
} else {  
    Print("Tugas terlambat!")  
}
```

Flowchart dan Pseudocode IF-ELSE



```
IF (kondisi) THEN  
    Pernyataan-1  
ELSE  
    Pernyataan-2  
END IF
```

Struktur Pemilihan IF-ELSE IF-ELSE

Contoh Pemilihan dengan Banyak Kondisi

- Saat berkendara menuju kampus, kita sering bertemu lampu lalu lintas



Setiap warna lampu adalah **kondisi**, aksi kita tergantung pada kondisi tersebut.

- **Merah** → Berhenti
- **Hijau** → Jalan terus
- **Kuning** → Bersiap, hati-hati

Dari contoh ini kita bisa lihat kalau **tidak hanya ada dua kemungkinan**, tapi ada beberapa kondisi yang berbeda. Dalam pemrograman, situasi seperti ini bisa ditangani dengan struktur **IF – ELSE IF – ELSE**.



Struktur Pemilihan IF-ELSE IF-ELSE

- Sintaks pada Java:

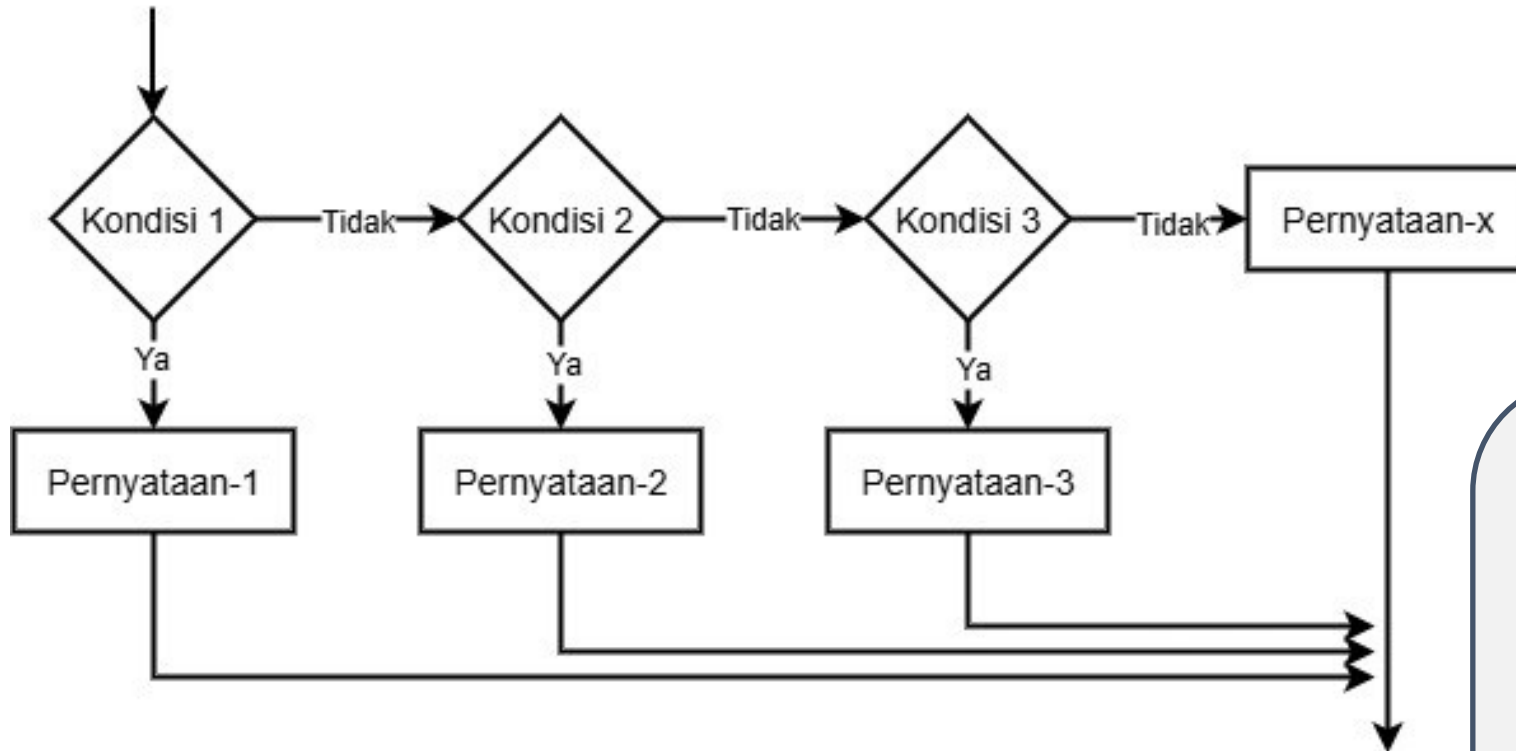
```
if (kondisi-1) {  
    Pernyataan-1;  
} else if (kondisi-2) {  
    Pernyataan-2;  
} else if (kondisi-3) {  
    Pernyataan-3;  
}  
...  
...  
else {  
    Pernyataan-x;  
}
```

- **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, dst.
- Jika tidak ada satupun kondisi yang terpenuhi, maka **Pernyataan-x** akan dieksekusi.

Contoh Penggunaan IF-ELSE IF-ELSE

```
if (kendaraan == "mobil") {  
    Print("Biaya parkir Rp 5.000")  
} else if (kendaraan == "sepeda motor") {  
    Print("Biaya parkir Rp 3.000")  
} else if (kendaraan == "sepeda") {  
    Print("Biaya parkir Rp 2.000")  
} else {  
    Print("Jenis kendaraan tidak dikenal")  
}
```

Flowchart dan Pseudocode IF-ELSE IF-ELSE



```
IF (kondisi-1) THEN  
    Pernyataan-1  
ELSE IF (kondisi-2) THEN  
    Pernyataan-2  
ELSE IF (kondisi-3) THEN  
    Pernyataan-3  
ELSE  
    Pernyataan-x  
END IF
```

Struktur Pemilihan SWITCH-CASE

Struktur Pemilihan SWITCH-CASE

- Sintaks pada Java:

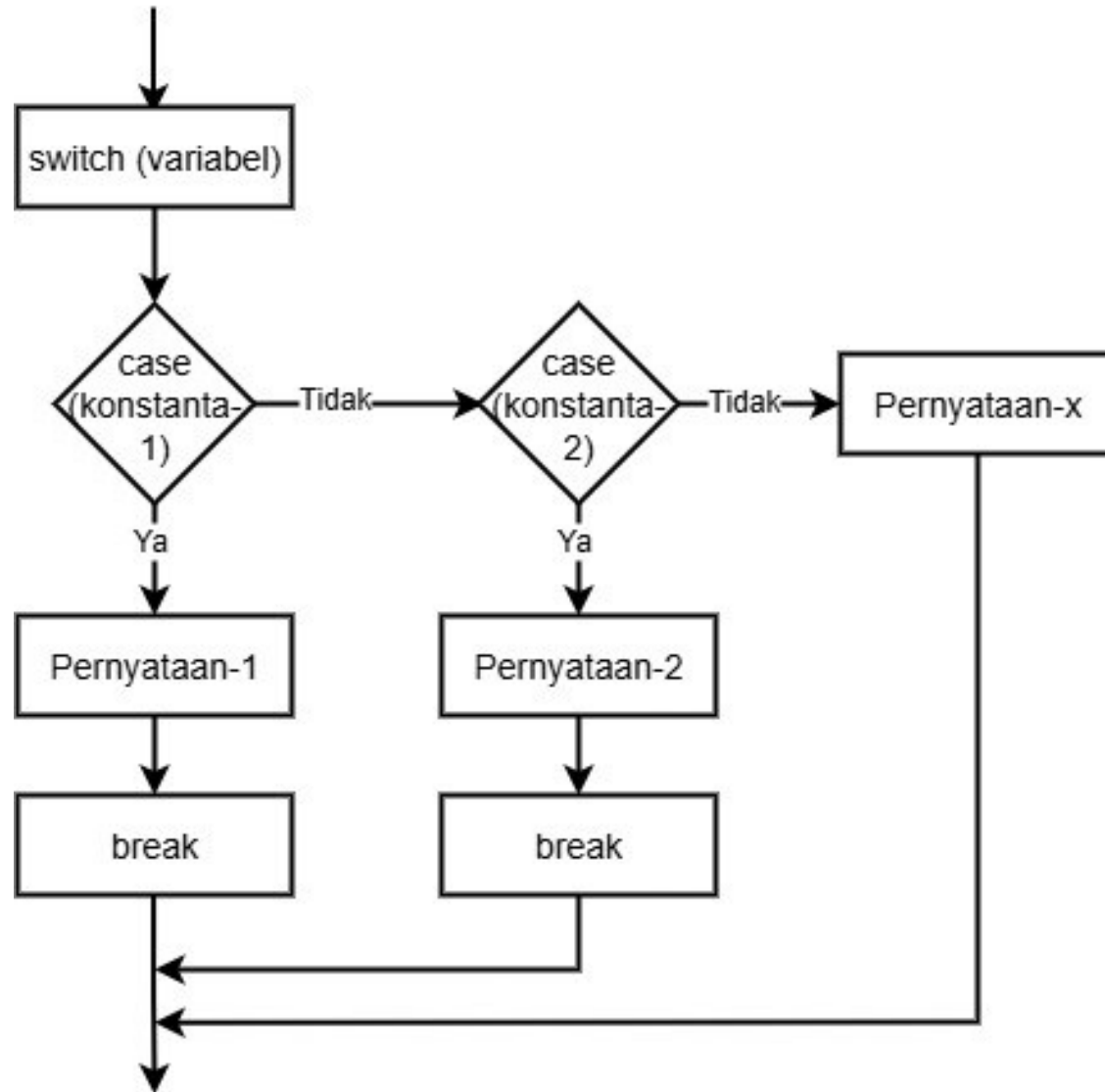
```
Switch (variabel) {  
    case Konstanta-1:  
        Pernyataan-1;  
    break;  
    case Konstanta-2  
        Pernyataan-2;  
    break;  
    ...  
    ...  
    default;  
        Pernyataan-x;  
}
```

- Sintaks pemilihan ini menjalankan **salah satu** dari beberapa pernyataan **case** sesuai dengan nilai **variabel** yang ada di dalam **switch**.
- Proses akan dilanjutkan sampai ditemukan pernyataan **break**.
- Jika tidak ada nilai pada **case** yang sesuai dengan nilai **kondisi**, maka proses akan dilanjutkan ke pernyataan yang ada di dalam **default**.

Contoh Penggunaan SWITCH-CASE

```
switch (menu){  
    case 1:  
        Print ("Input Nilai")  
        break  
    case 2:  
        Print ("Lihat Transkrip")  
        break  
    case 3:  
        Print ("Cetak KRS")  
        break  
    default:  
        Print ("Menu tidak valid")  
}
```

Flowchart dan Pseudocode SWITCH-CASE



```
SWITCH (variabel)  
    CASE konstanta-1:  
        Pernyataan-1  
    CASE konstanta-2:  
        Pernyataan-2  
    DEFAULT:  
        Pernyataan-x  
END SWITCH
```

Pada pseudocode, tidak perlu menuliskan **break**

Nested IF

Pemilihan Bersarang

- Pemilihan bersarang (NESTED IF) merupakan jenis pemilihan yang digunakan untuk mengambil keputusan dalam bentuk level (bertingkat)
- Di dalam suatu pernyataan IF (atau IF-ELSE) bisa saja terdapat pernyataan IF (atau IF-ELSE) yang lain



Struktur Pemilihan Nested-IF

• Bentuk Umum:

```
if (Kondisi-1) {  
    if (Kondisi-2) {  
        Pernyataan-1;  
        ...  
        ...  
        if (Kondisi-n) {  
            Pernyataan-2;  
        } else {  
            Pernyataan-3;  
        }  
    } else {  
        Pernyataan-n;  
    }  
} else {  
    Pernyataan-x;  
}
```

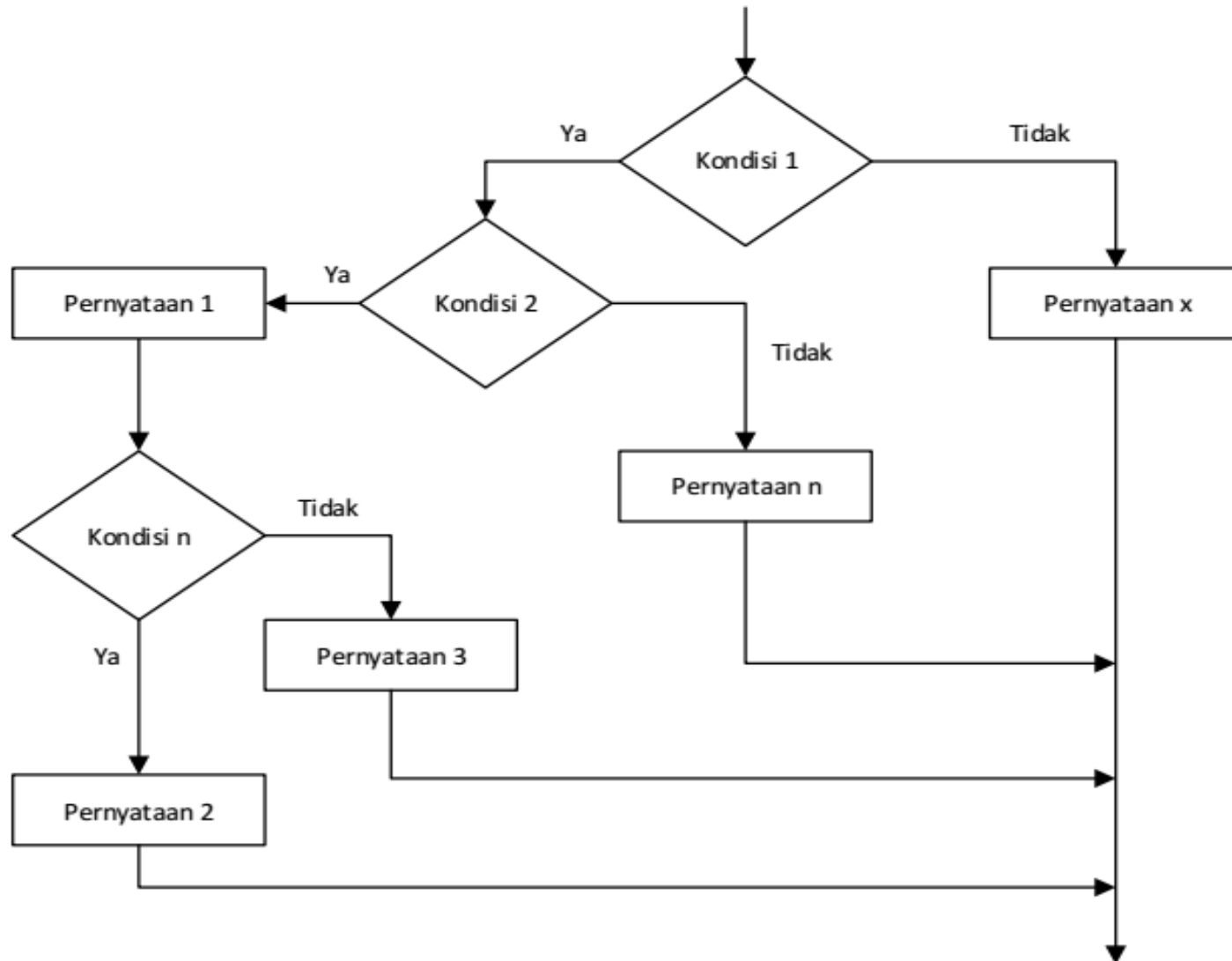
- Kondisi yang akan diseleksi pertama kali adalah kondisi IF yang berada di **posisi terluar (Kondisi-1)**.
- Jika **Kondisi-1** bernilai **SALAH**, maka pernyataan ELSE **terluar** (pasangan dari IF yang bersangkutan) yang akan **diproses**. Namun, jika pernyataan ELSE (pasangan dari IF) **tidak ditulis**, maka penyeleksian kondisi akan **dihentikan**.
- Jika ternyata **Kondisi-1** bernilai **BENAR**, maka kondisi berikutnya yang lebih dalam (**Kondisi-2**) akan diseleksi.
- Jika **Kondisi-2** bernilai **SALAH**, maka pernyataan ELSE (pasangan dari IF yang bersangkutan) yang akan **diproses**. Namun, jika pernyataan ELSE (pasangan dari IF) **tidak ditulis**, maka penyeleksian kondisi akan **dihentikan**, dst.



Contoh Penggunaan Nested IF

```
if (isLogin == true) {  
    if (role == "admin") {  
        Print ("Akses penuh")  
    } else if (role == "user") {  
        if (status == "aktif") {  
            Print ("Akses menu mahasiswa")  
        } else {  
            Print ("Akun tidak aktif")  
        }  
    }  
} else {  
    Print ("Login gagal")  
}
```

Flowchart dan Pseudocode Nested IF



```
IF (kondisi-1) THEN
    IF (kondisi-2) THEN
        Pernyataan-1
        IF (kondisi-n) THEN
            Pernyataan-2
        ELSE
            Pernyataan-3
        END IF
    ELSE
        Pernyataan-n
    END IF
ELSE
    Pernyataan-x
END IF
```

Memilih Struktur Pemilihan yang Tepat

- a. IF → untuk kondisi sederhana
- b. IF-ELSE → jika ada dua kondisi
- c. IF-ELSE IF-ELSE → jika ada banyak kategori (kondisi lebih dari dua)
- d. SWITCH-CASE → jika pilihan tetap dan jumlah banyak
- e. NESTED IF → jika kondisi saling bertingkat

Ternary Operator

Ternary Operator (? :)

- Bentuk singkat dari **IF-ELSE**
- Hanya digunakan jika ada **satu pernyataan** untuk kondisi **BENAR** dan **SALAH**
- Sintaks:

```
(Kondisi) ? aksi_jika_benar : aksi_jika_salah
```

- Ternary operator **hanya cara singkat di kode program**. Dalam pseudocode/flowchart, tetap digunakan IF-ELSE

Contoh Penggunaan Ternary Operator

```
if (saldo >= 20000) {  
    pesan = "Bisa beli kopi kekinian";  
} else {  
    pesan = "Saldo tidak cukup, top-up dulu";  
}
```

Ternary Operator

```
pesan = (saldo >= 20000) ? "Bisa beli kopi kekinian" : "Saldo tidak cukup, top-up dulu";
```




Ekspresi Logika &&, ||, dan !

Ekspresi Logika dalam Pemilihan

- Kadang kita perlu **lebih dari satu kondisi** dalam struktur pemilihan.
- Operator logika:
 - a. **&& (AND)** : **BENAR** jika **semua** kondisi **BENAR**
 - b. **|| (OR)** : **BENAR** jika **salah satu** kondisi **BENAR**
 - c. **! (NOT)** : membalik nilai kondisi



Contoh Penggunaan Ekspresi Logika

```
if (username == "mahasiswa" AND password == "pass"){  
    Print ("Akses diterima")  
} else {  
    Print ("Akses ditolak")  
}
```

```
if ( !(nilai >= 70) ) {  
    Print ("Remidi")  
}
```

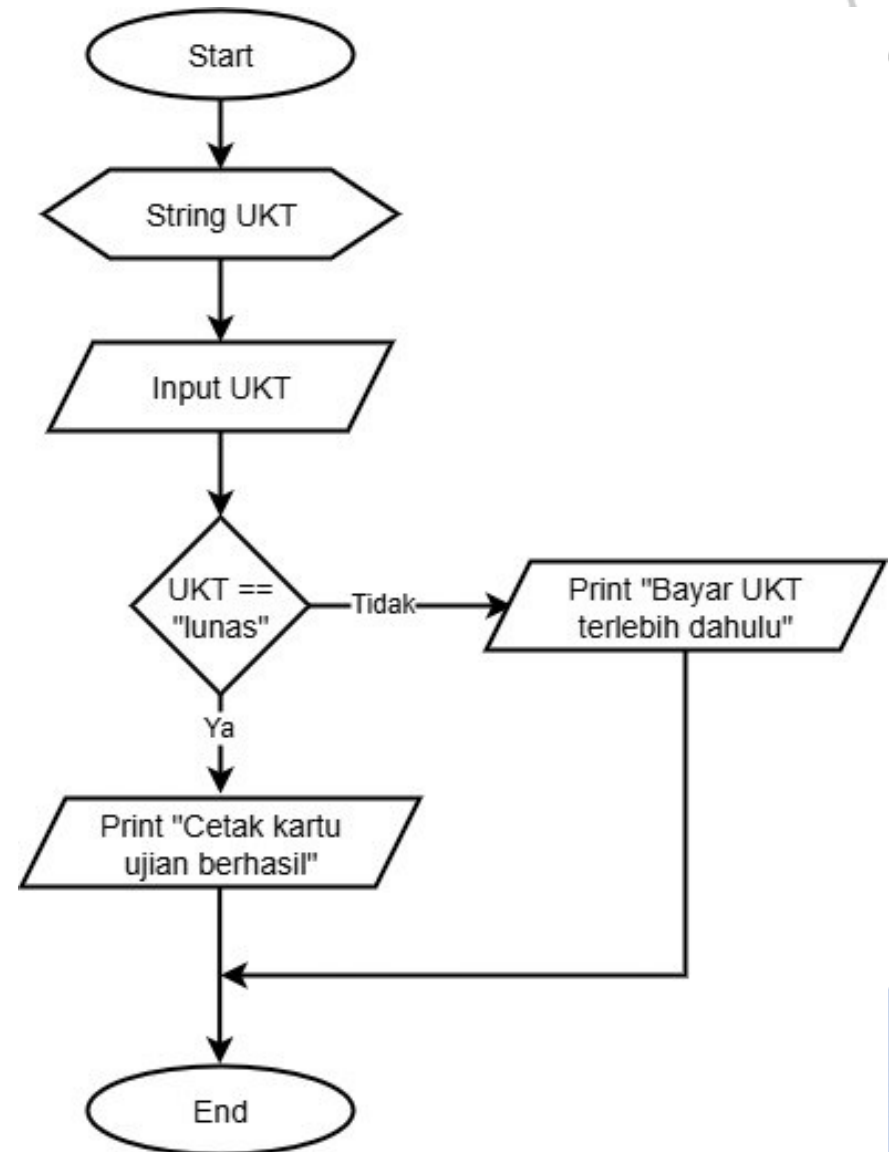
Contoh Studi Kasus

Kasus 1

- Seorang mahasiswa ingin mencetak kartu ujian. Sistem harus mengecek status pembayaran UKT terlebih dahulu.
 - Jika UKT sudah dibayar → mahasiswa dapat mencetak kartu ujian.
 - Jika UKT belum dibayar → sistem menolak pencetakan dan menampilkan pesan “Bayar UKT terlebih dahulu”.
- Buatlah flowchart dan pseudocode untuk menyelesaikan masalah tersebut!

Kasus 1 (Jawaban)

```
IF (UKT == "lunas") THEN  
    Print "Cetak kartu ujian berhasil"  
ELSE  
    tampilkan "Bayar UKT terlebih dahulu"  
END IF
```



Kasus 2

- Di gedung JTI lantai 7 terdapat mesin pemesanan minuman otomatis. Mahasiswa dapat memilih menu:
 - Pilih 1 → Cappuccino
 - Pilih 2 → Matcha
 - Pilih 3 → Oat milk
 - Pilih angka lain → Mencetak keterangan menu tidak tersedia
- Buatlah flowchart dan pseudocode untuk menyelesaikan masalah tersebut!

Kasus 2 (Jawaban)

```
SWITCH (menu)
CASE 1:
    Print "Cappuccino"
CASE 2:
    Print "Matcha"
CASE 3:
    Print "Oat milk"
DEFAULT:
    Print "Menu tidak tersedia"
END SWITCH
```

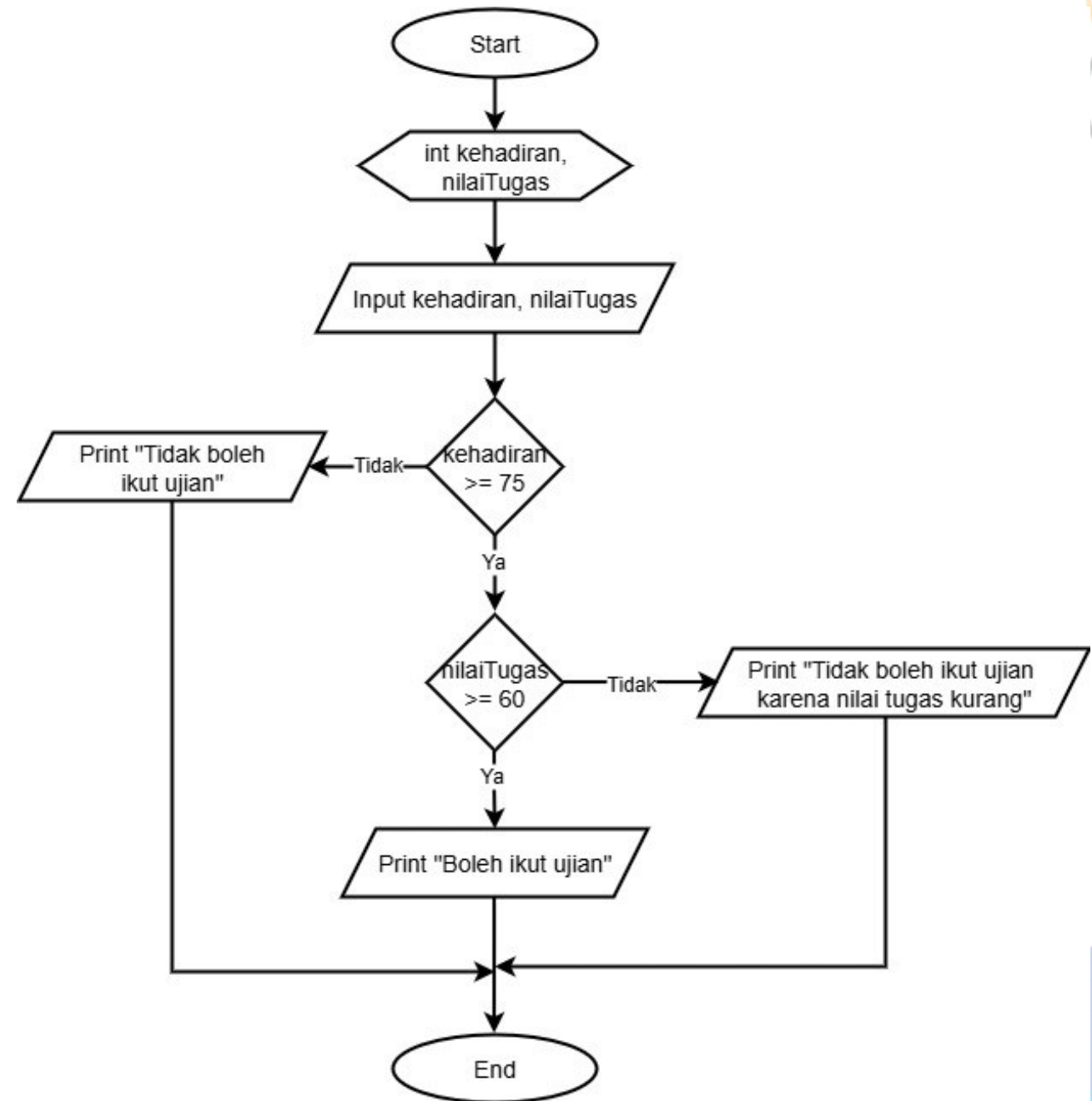


Kasus 3

- Sistem ujian online akan mengecek kelayakan mahasiswa untuk mengikuti ujian.
 - Jika presensi kehadiran $< 75\%$ → tampilkan “Tidak boleh ikut ujian”.
 - Jika presensi kehadiran $\geq 75\%$, maka dicek lagi nilai tugas:
 - Jika nilai tugas ≥ 60 → mahasiswa bisa ikut ujian.
 - Jika nilai tugas < 60 → tampilkan “Tidak boleh ikut ujian karena nilai tugas kurang”.
- Buatlah flowchart dan pseudocode untuk menyelesaikan masalah tersebut!



Kasus 3 (Jawaban Flowchart)



Kasus 3 (Jawaban Pseudocode)

```
IF (kehadiran >= 75) THEN
    IF (nilaiTugas >= 60) THEN
        Print "Boleh ikut ujian"
    ELSE
        Print "Tidak boleh ikut ujian karena nilai tugas kurang"
    END IF
ELSE
    Print "Tidak boleh ikut ujian"
END IF
```

Latihan

1. Sistem perpustakaan kampus memberikan izin masuk dengan dua syarat:
 - Jika mahasiswa membawa kartu mahasiswa, atau sudah melakukan registrasi online, maka boleh masuk.
 - Jika tidak memenuhi salah satu dari dua syarat tersebut, maka ditolak masuk.Buatlah flowchart dan pseudocode untuk menyelesaikan masalah tersebut!
2. Di kampus tersedia layanan WiFi gratis yang hanya bisa diakses oleh civitas akademika. Sistem hotspot kampus akan melakukan pengecekan sebagai berikut:
 - Jika jenis pengguna adalah dosen, maka tampilkan “Akses WiFi diberikan (dosen)”.
 - Jika jenis pengguna adalah mahasiswa, maka sistem akan memeriksa jumlah SKS yang diambil:
 - Jika $SKS \geq 12$, maka tampilkan “Akses WiFi diberikan (mahasiswa aktif)”.
 - Jika $SKS < 12$, maka tampilkan “Akses ditolak, SKS kurang dari 12”.
 - Jika bukan mahasiswa maupun dosen, maka tampilkan “Akses ditolak”.Buatlah flowchart dan pseudocode untuk menyelesaikan masalah tersebut!