

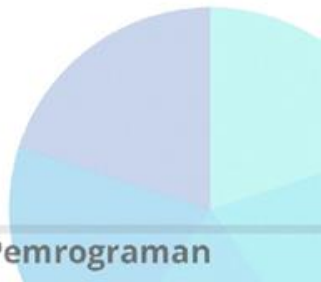
# PEMILIHAN 2

## Team Teaching Dasar Pemrograman 2023



# Tujuan

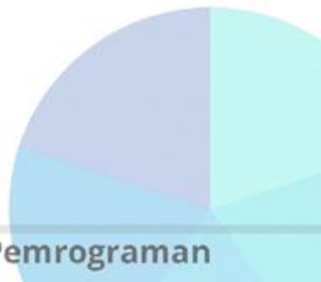
- Mahasiswa mampu memahami definisi dan kegunaan sintaks pemilihan bersarang
- Mahasiswa memahami struktur dasar sintaks pemilihan bersarang
- Mahasiswa mampu menyelesaikan permasalahan dengan membuat sebuah program Java yang memanfaatkan sintaks pemilihan bersarang





# 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





## Bentuk Umum Pemilihan Bersarang:

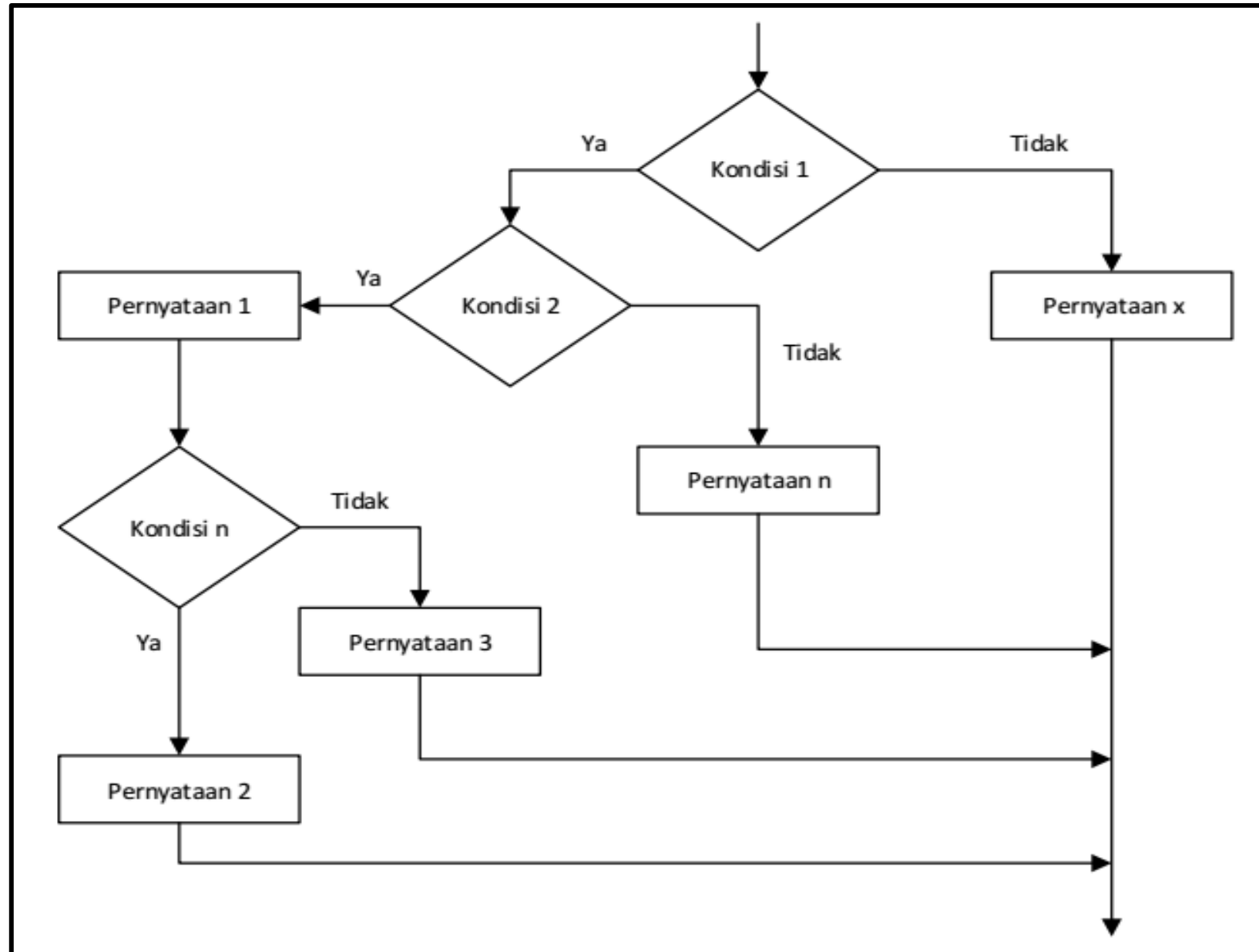
```
if (kondisi 1){  
    if (kondisi 2){  
        pernyataan 1;  
        ...  
        ...  
        if (kondisi n){  
            pernyataan 2;  
        } else {  
            pernyataan 3;  
        }  
    } else {  
        pernyataan n;  
    }  
} else {  
    pernyataan x;  
}
```



# Pemilihan Bersarang

- 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.

## Flowchart Pemilihan Bersarang:



# Contoh Penggunaan Pemilihan Bersarang

Ketika seseorang melakukan pembayaran di kasir. Kasir akan memberikan pertanyaan:

**Apakah pelanggan mempunyai kartu anggota?**

➤ TRUE:

- **Apakah total harga barang belanjaan lebih dari Rp 500.000?**

- ❖ TRUE: Pelanggan mendapatkan diskon Rp 50.000

- ❖ FALSE: Pelanggan mendapatkan diskon Rp 25.000

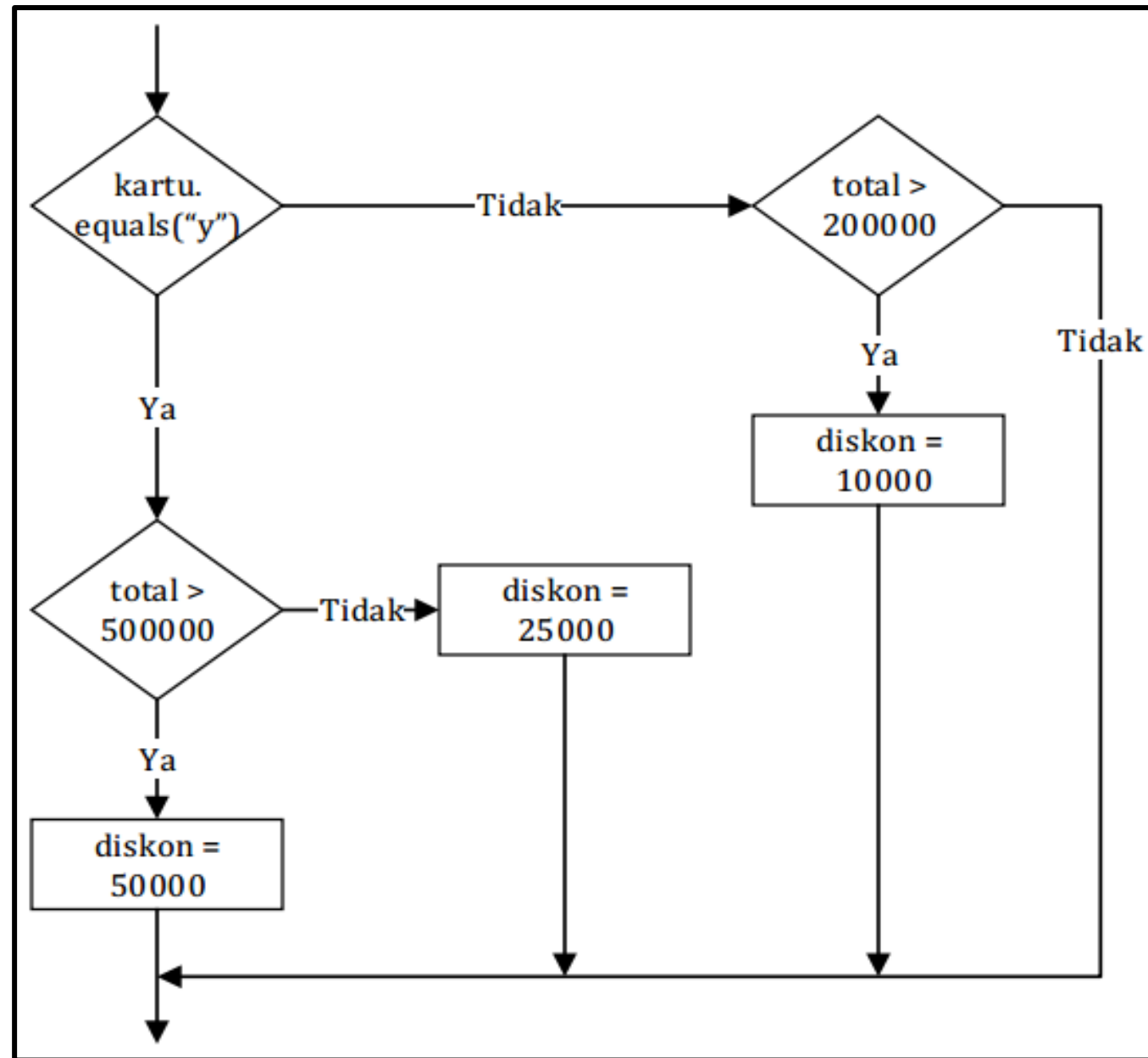
➤ FALSE:

- **Apakah total harga barang belanjaan lebih dari Rp 200.000?**

- ❖ TRUE: Pelanggan mendapatkan diskon Rp 10.000

- ❖ FALSE: Pelanggan tidak mendapatkan diskon

## Contoh Flowchart:







## Contoh Program:

```
import java.util.Scanner;
public class kasir {
    public static void main(String[] args) {
        int total, diskon, bayar;
        String kartu;
        Scanner sc = new Scanner (System.in);
        System.out.print("Apakah pelanggan mempunyai kartu anggota (y atau t)? ");
        kartu = sc.nextLine();
        System.out.print("Berapa total harga barang belanjaan? Rp ");
        total = sc.nextInt();
        if (kartu.equals("y")) {
            if (total > 500000) {
                diskon = 50000;
            } else {
                diskon = 25000;
            }
        } else {
            if (total > 200000) {
                diskon = 10000;
            } else {
                diskon = 0;
            }
        }
        bayar = total - diskon;
        System.out.println("Total yang harus dibayar: Rp " + bayar);
    }
}
```

# Ekspresi Logika

➤ Terdapat 3 jenis operator logika yang digunakan pada pernyataan IF-ELSE, yaitu:

❖ && : AND

❖ || : OR

❖ ! : NOT

➤ **Ekspresi logika** adalah ekspresi yang menggunakan satu atau lebih operator logika.

➤ Operator yang diterapkan pada ekspresi logika akan dievaluasi dari **kiri ke kanan**

# Ekspresi Logika

- Ketika mengevaluasi **( $e_1$  &&  $e_2$ )**, jika  $e_1$  menghasilkan FALSE, maka  $e_2$  tidak akan dievaluasi. Dengan demikian, nilai seluruh ekspresi ( $e_1$  &&  $e_2$ ) akan dianggap salah
- Namun, jika  $e_1$  menghasilkan TRUE, maka selanjutnya  $e_2$  akan dievaluasi untuk menentukan nilai seluruh ekspresi
- Contoh:

```
If (kecepatan == 0 && mesinOn == true)  
    System.out.println("Matikan mesin");
```

# Ekspresi Logika

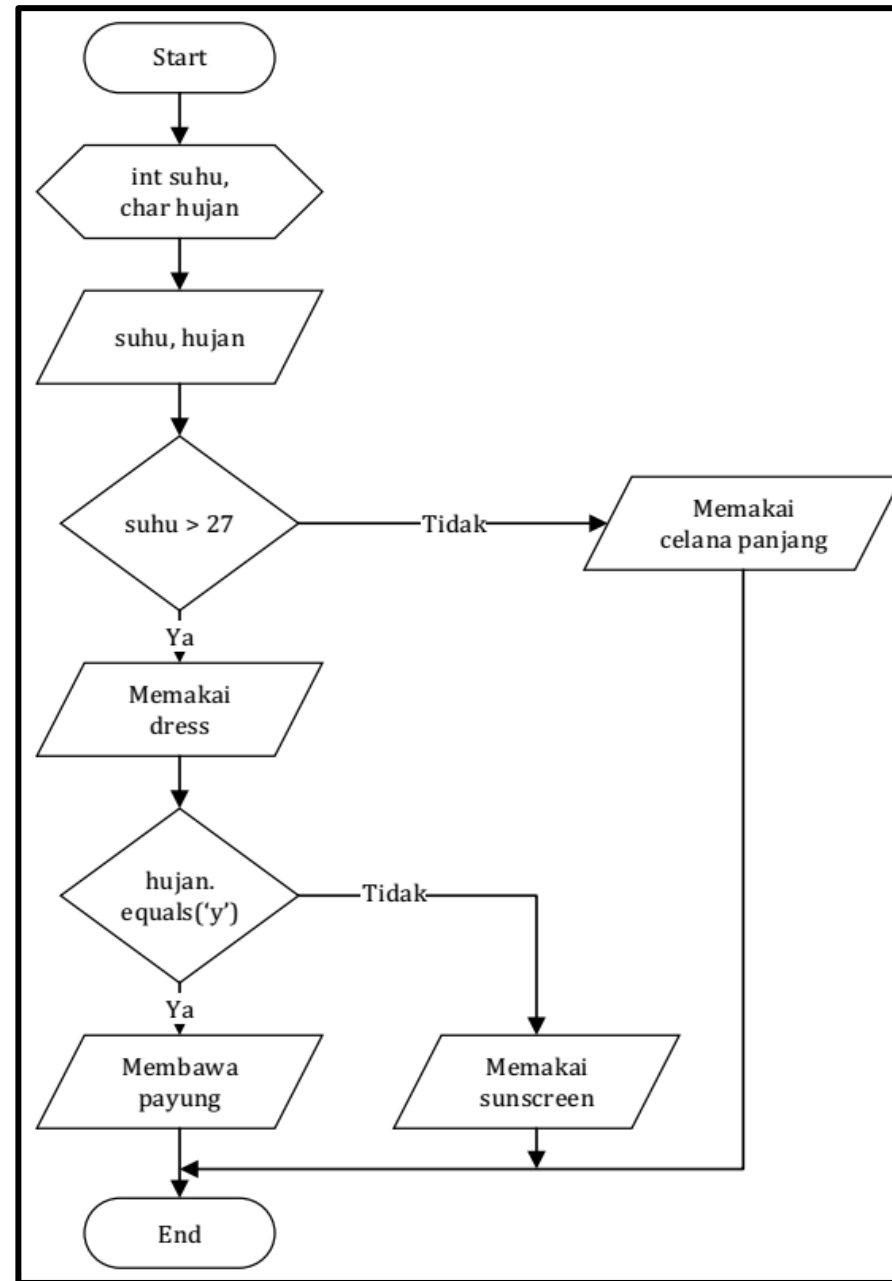
- Ketika mengevaluasi **( $e_1 \parallel e_2$ )**, jika  $e_1$  menghasilkan TRUE, maka  $e_2$  tidak akan dievaluasi. Dengan demikian, nilai seluruh ekspresi ( $e_1 \parallel e_2$ ) akan dianggap benar
- Namun, jika  $e_1$  menghasilkan FALSE, maka selanjutnya  $e_2$  akan dievaluasi untuk menentukan nilai seluruh ekspresi
- Contoh:

```
If (kecepatan == 0 || mesinOn == true)  
    System.out.println("Matikan mesin");
```

# Contoh Studi Kasus

- Sebuah sistem dibuat untuk menentukan pakaian dan peralatan yang harus dibawa pengguna sesuai dengan kondisi cuaca. Jika suhu lebih dari  $27^{\circ}\text{C}$ , maka pengguna disarankan memakai dress, kemudian dilakukan pengecekan apakah saat ini hujan, jika hujan maka pengguna disarankan untuk membawa payung, sedangkan jika tidak hujan maka pengguna disarankan untuk memakai sunscreen. Namun, jika suhu kurang dari atau sama dengan  $27^{\circ}\text{C}$ , maka pengguna disarankan memakai celana panjang
- Buatlah flowchart untuk sistem tersebut!

## Flowchart Studi Kasus:



# LATIHAN MANDIRI

# Latihan 1

Susunlah algoritma dengan menggunakan flowchart untuk menentukan bilangan terbesar dari masukan 3 buah bilangan (tanpa menggunakan operator logika)

*input :*

bil1 = 28

bil2 = 54

bil3 = 15

*output :*

bilangan terbesar : 54



## Latihan 2

Setiap hari Rabu, sebuah toko buku memberikan diskon kepada pelanggannya sesuai jenis buku yang dibeli

- Diskon sebesar 10% diberikan jika buku yang dibeli adalah kamus, kemudian akan diberikan tambahan diskon sebesar 2% jika buku yang dibeli lebih dari 2
- Diskon sebesar 7% diberikan jika buku yang dibeli adalah novel, kemudian akan diberikan tambahan diskon sebesar 2% jika novel yang dibeli lebih dari 3, sedangkan jika novel yang dibeli kurang dari atau sama dengan 3 akan diberikan tambahan diskon sebesar 1%
- Pelanggan akan mendapatkan diskon untuk buku selain kamus dan novel sebesar 5% jika buku yang dibeli lebih dari 3 produk

Buatlah flowchart (gunakan operator logika) untuk menentukan berapa total yang harus dibayar jika input yang dimasukkan adalah jenis dan jumlah buku, dan outputnya adalah jumlah diskon

# Tugas Diskusi Kelompok Project

1. Identifikasi sesuai project masing-masing fitur apa saja yang membutuhkan konsep pemilihan bersarang.
2. Tentukan bentuk pemilihan yang digunakan, serta masing-masing kondisi yang dibutuhkan, dengan atau tanpa operator logika
3. Buatlah algoritma dalam bentuk flowchart sesuai kebutuhan yang telah Anda identifikasi berdasarkan tugas No 1 dan 2