

KUSUMA WEB ACADEMY

DASAR-DASAR JALUR JAVA

Modul 05: Kontrol Aliran - Percabangan & Kondisional

Hak Cipta © 2026 Kusuma Web. All Rights Reserved.

Percabangan Dasar: Struktur if, if-else

Struktur if Tunggal

- ▶ Blok kode hanya dieksekusi jika kondisi bernilai true. Jika false, blok dilewati.

```
if (nilai >= 75) {  
    System.out.println("Lulus KKM!");  
}
```

Struktur if-else Ganda

- ▶ Menyediakan jalur alternatif jika kondisi utama tidak terpenuhi (false).

```
if (saldo >= tarikan) {  
    saldo -= tarikan;  
} else {  
    System.out.println("Saldo Kurang!");  
}
```

Aturan Blok Penulisan

- ▶ Jika blok pernyataan di dalam if hanya terdiri dari **satu baris** instruksi, penggunaan kurung kurawal {} bersifat opsional.
- ▶ *Best Practice:* Selalu gunakan kurung kurawal untuk menghindari *logic error* saat memodifikasi kode di masa depan.

Keuntungan Struktural

Membantu program berpikir dinamis dalam merespons data masukan dari pengguna secara interaktif.

Percabangan Banyak Kondisi (Ladder & Nested)

If-Else-If Ladder

- ▶ Digunakan ketika terdapat lebih dari dua alternatif keputusan saling lepas yang perlu dicek bertahap.

```
int nilai = 82;
if (nilai >= 90) {
    System.out.println("Grade: A");
} else if (nilai >= 80) {
    System.out.println("Grade: B");
} else if (nilai >= 70) {
    System.out.println("Grade: C");
} else {
    System.out.println("Grade: D");
}
```

Nested If (Percabangan Bersarang)

- ▶ Menempatkan blok if di dalam blok if lainnya untuk pemeriksaan berlapis.

```
boolean punyaKartuMember = true;
int totalBelanja = 150000;

if (punyaKartuMember) {
    if (totalBelanja >= 100000) {
        System.out.println("Diskon 10%");
    } else {
        System.out.println("Diskon 5%");
    }
} else {
    System.out.println("Tidak ada diskon");
}
```

Percabangan Efisien: Struktur switch-case

Kapan Menggunakan Switch Case?

- ▶ Sangat efisien dibanding `if-else-if` jika Anda membandingkan satu variabel tunggal dengan banyak nilai konstan (literal).
- ▶ Mendukung tipe data primitif `byte`, `short`, `char`, `int`, serta tipe referensi `String` dan `Enum`.

Keyword `break` itu Wajib!

Lupa menambahkan `break` akan memicu efek *fall-through*, di mana semua case di bawahnya ikut tereksekusi tanpa peduli kondisinya!

Struktur Kode Standar Switch

```
int hari = 3;
switch (hari) {
    case 1:
        System.out.println("Senin");
        break;
    case 2:
        System.out.println("Selasa");
        break;
    case 3:
        System.out.println("Rabu");
        break; // Berhenti di sini
    default:
        System.out.println("Hari tidak valid");
}
```

Sintaks Baru dengan Arrow Operator (->)

- ▶ Menghilangkan keharusan menulis keyword break.
- ▶ Lebih ringkas, aman dari bahaya *fall-through*, dan mengembalikan nilai secara instan.

Contoh Multi-Case:

- ▶ Anda dapat menggabungkan beberapa kondisi pencocokan dalam satu baris menggunakan tanda koma.

Implementasi Switch Expression

```
String jenisHari = "RABU";  
String status = switch (jenisHari) {  
    case "SENIN", "SELASA", "RABU",  
         "KAMIS", "JUMAT" -> "Hari  
                             Kerja";  
    case "SABTU", "MINGGU" -> "Akhir  
                             Pekan";  
    default -> "Hari Tidak Dikenal";  
};  
  
System.out.println(status);
```

Catatan: Sintaks modern ini membutuhkan konfigurasi compiler Java minimal versi 12 atau di atasnya.