

MODUL 5. CONTROL FLOW: SWITCH-CASE

5.1 Tujuan

5.2 Alat dan Bahan

5.3 Dasar Teori

Kita sudah mempelajari tentang percabangan memakai perintah if-else sebelumnya. Sekarang, kita akan berkenalan dengan perintah percabangan yang lain, yaitu switch-case!

5.3.1 Switch vs If

Lalu, apa yang membedakan switch-case ini dengan if-else? Yang membedakan, adalah pada nilai yang ia terima. Kita tahu kalau if-else ini harus menerima boolean, bukan? Bila true, maka blok kode akan dijalankan. Bila tidak, maka akan di-skip. Namun switch-case ini bisa menerima tipe data apapun, lho. Tapi, tentu cara penggunaannya berbeda. Mari kita lihat strukturnya!

```
switch(kondisi) {  
    case nilai_kondisi1:  
        // aksi  
        break;  
    case nilai_kondisi2:  
        // aksi  
        break;  
    default:  
        // aksi default  
        break;  
}
```

Perintah switch ini akan menerima sebuah variabel atau nilai, apapun itu, dan kemudian mencocokkannya kasus-kasus (cases) yang ada. Apakah nilai yang diberikan sesuai dengan nilai yang ditulis dalam kasus? Bila ya, maka kasus tersebut akan dijalankan. Kemudian, ada lagi yang perlu kita perhatikan, yaitu perintah *break*. Perintah ini digunakan untuk menghentikan switch-case. Tidak seperti if-else yang hanya mengeksekusi satu blok saja, bila satu kasus sudah terpenuhi, switch-case akan terus berjalan ke kasus-kasus selanjutnya.

Karena itulah, kita membutuhkan *break* untuk menghentikan jalannya switch-case ketika kondisi yang kita inginkan sudah terpenuhi. Namun, bukan berarti kita akan selalu memakai *break* untuk switch-case. Ada beberapa kondisi yang mana lebih baik bagi kita untuk tidak menggunakan *break* pada kasus tertentu.

Yang terakhir, ada *default*. Mungkin dari namanya, kalian sudah bisa menebak apa default ini. Default ini berlaku seperti *else* pada *if-else* ya? Jadi ia adalah pelarian terakhir bila kasus-kasus sebelumnya tidak ada yang memenuhi.

Kalau begitu, mari kita lihat contoh implementasi sederhananya.

```
import java.util.Scanner;

public class ProgramDira {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkan kelas anda: ");

        String kelas = scanner.next();

        switch(kelas) {
            case "01":
                System.out.println("Selamat datang di kelas 01!");
                break;
            case "02":
                System.out.println("Selamat datang di kelas 02!");
                break;
            case "03":
                System.out.println("Selamat datang di kelas 03!");
                break;
            case "04":
                System.out.println("Selamat datang di kelas 04!");
                break;
            default:
                System.out.println("Kelas tidak ada!");
                break;
        }
    }
}
```

Bila kalian jalankan program di atas, maka kalian akan dimintai input nomor kelas kalian. Dan bila nomor kelas kalian ada pada case, maka kalian akan disambut. Namun bila tidak, maka dia akan lari ke *default* dan mencetak “Kelas tidak ada!”. Dari sini, apakah kalian sudah mengetahui apa perbedaan *if-else* dan *switch-case*, lalu apa perbedaan kegunaan dari mereka?

Bila belum, mari kita bahas bersama-sama! Pertama-tama, cara switch-case bekerja. Berbeda dengan if-else yang memakai boolean, switch-case ini mencocokkan nilai asli nilai yang di-switch (contohnya kelas di atas) dengan nilai yang dituliskan pada case. Sedangkan if-else, ia menggunakan boolean, sehingga kondisi seperti apapun akan selalu dieksekusi apabila kondisi aneh tersebut menghasilkan boolean "true". Dari sini, kita juga bisa simpulkan, lho perbedaan kegunaan dari kedua perintah tersebut.

Yang pertama dan paling jelas, switch-case digunakan untuk nilai yang sudah pasti, atau nilai konstanta. Contoh misalnya tipe pasien di rumah sakit ada tiga, yaitu pasien "BPJS", "Reguler", dan "VIP". Tipe pasien-pasien tersebut tidak mungkin berubah, bukan? Hanya tiga itu saja. Oleh karena itu, di sini sebaiknya kita menggunakan switch-case karena hanya akan ada 3 kasus saja.

```
case "BPJS":  
    break;  
case "Reguler":  
    break;  
case "VIP":  
    break;
```

Sedangkan if-else dapat kita gunakan untuk kondisi yang tidak kita ketahui secara pasti nilainya. Contohnya adalah rentang indeks nilai

```
if (nilai >= 80) {  
    System.out.println("Nilai Anda: A");  
} else if (nilai >= 70) {  
    System.out.println("Nilai Anda: B");  
} else if (nilai >= 50) {  
    System.out.println("Nilai Anda: C");  
} else if (nilai >= 30) {  
    System.out.println("Nilai Anda: D");  
} else {  
    System.out.println("Nilai Anda: E");  
}
```

Tidak mungkin, kan kita memakai switch-case di sini. Masa kita harus menulis 100 kasus? Hahaha, itu bukan kerja cerdas!

5.3.2 Tanpa memakai break?

Terakhir, kita akan membahas kasus di mana *break* bisa tidak dibutuhkan. Kita tidak membutuhkan *break* apabila kita ingin beberapa kasus memiliki proses yang sama persis. Contohnya? Mari kita ubah switch-case yang kita buat sebelumnya, yuk!

```
public class ProgramDira {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkan kelas anda: ");

        String kelas = scanner.next();

        switch(kelas) {
            case "01":
            case "02":
            case "03":
            case "04":
                System.out.println("Kelas ada!");
                break;
            default:
                System.out.println("Kelas tidak ada!");
                break;
        }
    }
}
```

Anggap saja kita sedang membuat program untuk cek kelas mana saja yang ada. Kita hanya membutuhkan dua output di atas, yaitu “Kelas ada!” dan “Kelas tidak ada!”. Ingat, tanpa *break*, switch-case akan terus berjalan ke kasus-kasus berikutnya. Jadi, misal ada yang input “01”, ia akan terus berjalan hingga “04”, dan kemudian menjalankan perintah *println* yang ada di kasus 04. Tanpa memakai *break*, kita bisa mempersingkat kode di atas dan kelas apapun yang valid yang pengguna pilih (01-04), maka semua kasus akan berhenti di kasus 04, dan mencetak “Kelas ada!”. Kita tidak perlu menulis

```
System.out.println("Kelas ada!");
break;
```

terus-menerus lagi untuk setiap kasus. Lebih simpel deh!

Baiklah, itu saja untuk bab switch-case. Sampai jumpa di bab selanjutnya!

Rizza Mandasari, Rahman Hakim

Ilustrasi oleh Nazwa Mursyidan Baldan