

Praktikum IV : Percabangan II

Tujuan Praktikum :

1. Praktikan dapat memahami konsep Switch-Case pada Java.
2. Praktikan dapat memahami Flowchart dari Konsep Switch-case.
3. Praktikan dapat mengimplementasikannya ke dalam program.

Switch-Case

Switch Case merupakan sebuah pernyataan atau *statement* yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan banyak alternatif pilihan. Pernyataan **Switch** lebih jarang digunakan dibandingkan dengan pernyataan **IF** pada java. Tetapi lebih sering digunakan ketika kita ingin menuliskan percabangan ataupun pengambilan keputusan dengan sangat banyak pilihan. Maka dari itu **switch** sangat baik untuk dipelajari.

- ❖ Pernyataan switch case merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah alternatif.
- ❖ Jika pada percabangan if-else kita menggunakan tipe data boolean sebagai kondisional, maka akan sedikit berbeda pada pernyataan switch case.
- ❖ Pada switch case kita dapat menggunakan beberapa tipe data primitif seperti short, char, int, string, serta tipe data bentukan sebagai kondisi yang akan kita uji.
- ❖ Pernyataan switch memungkinkan kita untuk menguji sebuah variabel terhadap daftar nilai. Setiap nilai disebut **case** lalu variabel akan dicek untuk setiap case.
- ❖ Nilai untuk sebuah case harus memiliki tipe data yang sama dengan variabel.

Deklarasi umum Switch-case

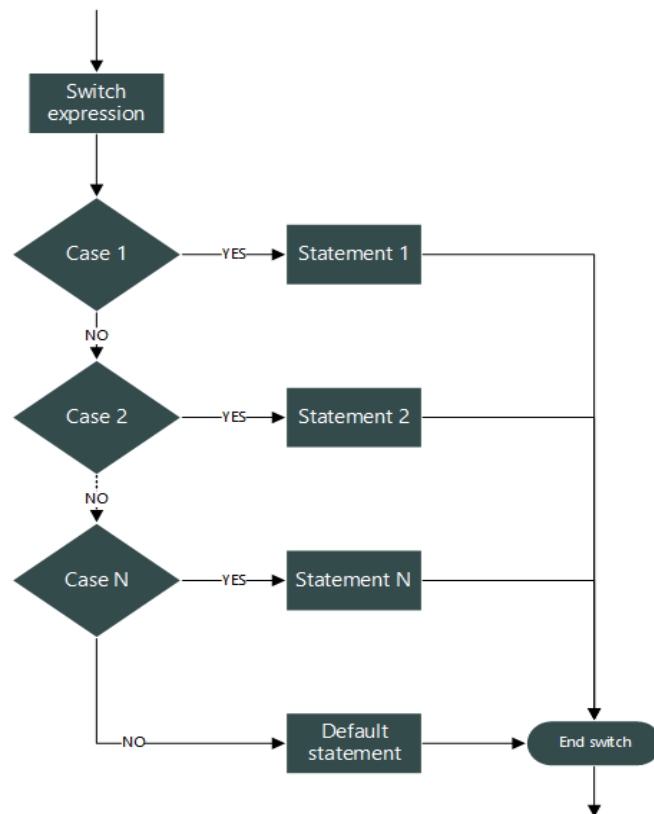
```
switch(ekspresi) {  
    // pernyataan-pernyataan case  
    // setiap nilai harus bertipe data sama dengan ekspresi  
    case nilai1 :  
        //statement yang akan dieksekusi
```

```

...
break; // break opsional
case nilai2 :
//statement yang akan dieksekusi
...
break; // break opsional
...
default :
//statement alternatif apabila semua case tidak
terpenuhi, tidak memerlukan break}
}

```

Flowchart



Perbedaan switch case dengan if-else if

1. Apa perbedaan if dengan switch ?

- Switch, kondisi hanya dinyatakan dengan bilangan bulat atau karakter/string sedangkan if-else yang dapat menggunakan operasi seperti <, >, <= dan >=.
- Tidak ada konstanta yang sama dalam sebuah switch.

- Angka yang digunakan harus berkisar diantara 0 – 255.
- Harus menggunakan break, Gunanya untuk mengakhiri statement.
- Ada default, ini bisa dibilang sebagai 'Else'nya Switch Case.

2. Kapan Menggunakan If ?

3. Kapan Menggunakan Switch ?

IF digunakan apabila seleksi memiliki lebih dari 1 kondisi dan nilai.

SWITCH digunakan apabila hanya ada 1 kondisi dengan nilai yang bervariasi.

❖ Contoh Program 1

Program switch case menggunakan multiple case tanpa break :

```
package latihan.pkgswitch.pkgcase;

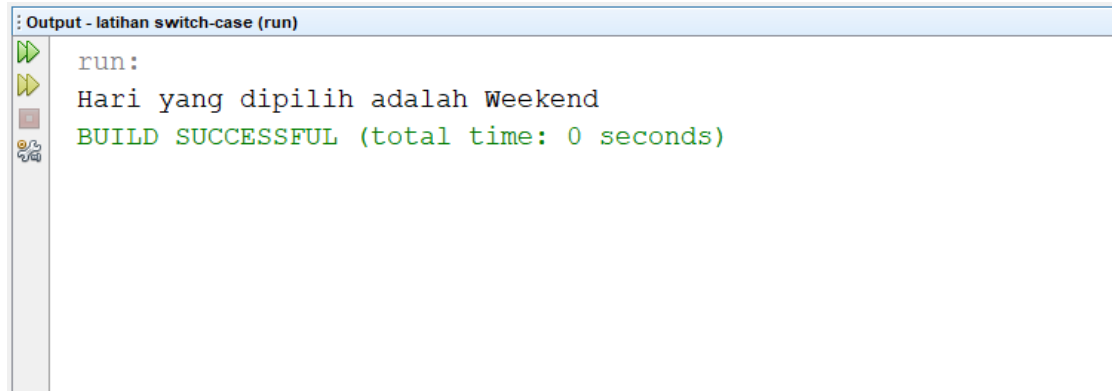
public class SwitchCase {
    public static void main(String[] args) {
        int hari = 7;
        String tipeHari;
        switch (hari) {
// multiple cases without break statements
            case 1:
            case 2:
            case 3:
            case 4:
            case 5:
                tipeHari = "Hari kerja";
                break;
            case 6:
            case 7:
                tipeHari = "Weekend";
                break;
            default:
                tipeHari = "Tipe hari yang dipilih salah";
        }
    }
}
```

```

        System.out.println("Hari yang dipilih adalah " +
tipeHari);
    }
}

```

Output Program



```

Output - latihan switch-case (run)
run:
Hari yang dipilih adalah Weekend
BUILD SUCCESSFUL (total time: 0 seconds)

```

❖ Contoh Program 2

```

package latihan.pkgswitch.pkgcase;
import java.util.Scanner;
public class switchCaseFour {
    public static void main(String[] hady) {
        Scanner in = new Scanner(System.in);
        String pilih;
        System.out.println("---malam minggunya programmer-
--\n");
        System.out.print("Kamu pilih aku atau dia ??? ");
        pilih = in.nextLine();
        switch (pilih) {
            case "kamu":
                System.out.print("\nAku pasti akan lebih
baik dari dia !");
                break;
            case "dia":
                System.out.print("Lebih baik aku pergi
jauh dari hidupmu");
                break;

```

```

        default:
            System.out.println("Jawaban Tidak
Sesuai");
        }
    }
}

```

Output Program

```

: Output - latihan switch-case (run) #2
run:
---malam minggunya programmer---

Kamu pilih aku atau dia ??? dia
Lebih baik aku pergi jauh dari hidupmuBUILD SUCCESSFUL (total time: 5 seconds)

```

❖ Contoh Program 3

- ❖ Misalnya kita ingin membuat percabangan jika kita menginputkan angka 1 maka yang keluar adalah nilai D , jika 3 maka nilai B. Perhatikan codingnya di bawah :

```

package latihan.pkgswitch.pkgcase;
import java.util.Scanner;
public class Switch {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Masukan nilai anda (1-4): ");
        int nilai = input.nextInt();
        switch (nilai) {
            case 1:
                System.out.println("Anda mendapatkan D");
                break;
            case 2:
                System.out.println("Anda mendapatkan C");
                break;

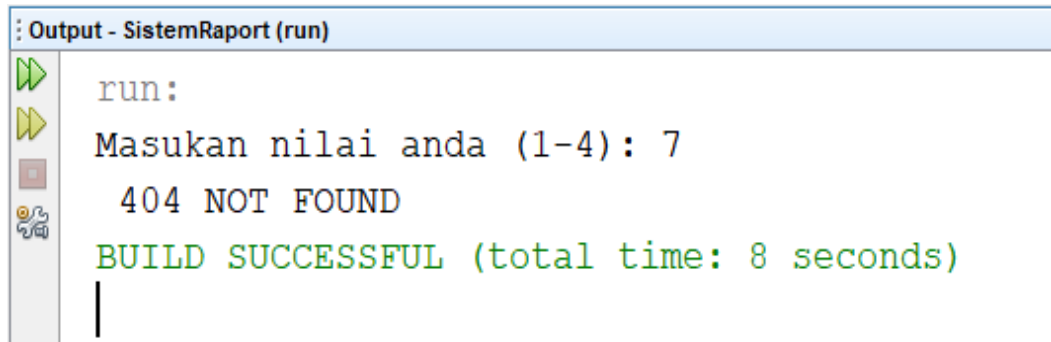
```

```

        case 3:
            System.out.println("Anda mendapatkan B");
            break;
        case 4:
            System.out.println("Anda mendapatkan A");
            break;
        default:
            System.out.println(" 404 NOT FOUND ");
    }
}

```

Contoh output Program



```

Output - SistemRaport (run)
run:
Masukan nilai anda (1-4): 7
404 NOT FOUND
BUILD SUCCESSFUL (total time: 8 seconds)

```

❖ Contoh Program 4

```

package challenge;
import java.util.Scanner;
public class Challenge {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int menu;
        System.out.println("1. Cek KTP");
        System.out.println("2. Print Semangat");
        System.out.println("3. Keluar");

        System.out.print("masukkan pilihan anda =====>");

        menu = in.nextInt();
    }
}

```

```

        switch (menu) {
            case 1:
                int thn1,
                    thn2;
                System.out.println("seleksi umur pembuatan
ktp");
                System.out.print("masukkan tahun lahir
:");

                thn1 = in.nextInt();
                thn2 = 2019 - thn1;
                System.out.println("umur anda sekarang
adalah " + thn2 + " tahun");
                if (thn2 > 16) {
                    System.out.println("anda bisa buat
ktp");
                } else {
                    System.out.println("Tidak bisa buat
ktp");
                }
                break;
            case 2:
                System.out.print("masukkan angka (1-10):
");

                angka = in.nextInt();
                switch (angka) {
                    case 1:
                    case 2:
                    case 3:
                    case 4:
                        System.out.println("Semangat
Awal");
                        break;
                    case 5:
                    case 6:
                    case 7:
                        System.out.println("Semangat
tengah");

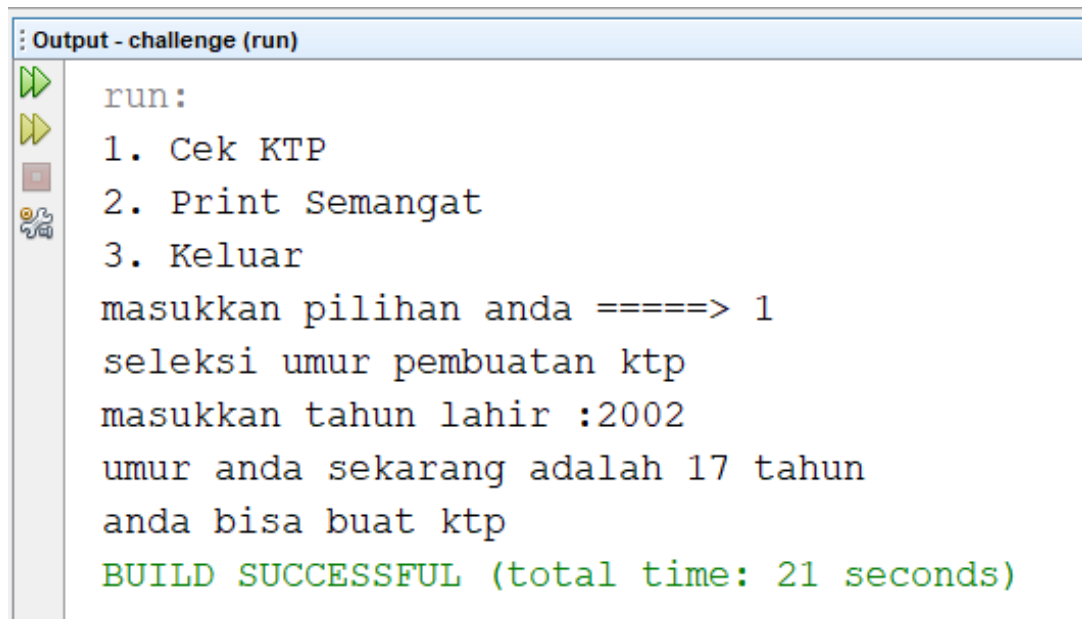
```

```

        break;
        case 8:
        case 9:
        case 10:
            System.out.println("Semangat
akhir");
            break;
        default:
            System.out.println("Kelewatan
Semangat");
    }
    break;
    default:
        System.out.println("Keluar");
        break;
    }
}
}

```

Output Program:



The screenshot shows an IDE output window titled "Output - challenge (run)". On the left side, there is a vertical toolbar with icons for running (a green play button), stepping through (a yellow play button), stopping (a red square), and debugging (a magnifying glass over a bug). The output text is as follows:

```

run:
1. Cek KTP
2. Print Semangat
3. Keluar
masukkan pilihan anda ====> 1
seleksi umur pembuatan ktp
masukkan tahun lahir :2002
umur anda sekarang adalah 17 tahun
anda bisa buat ktp
BUILD SUCCESSFUL (total time: 21 seconds)

```