



# Algoritma dan Pemrograman Komputer 1

## Bab 6: Struktur Kontrol 2

---

Aslam Pandu Tasminto – 5002241025

M. Ma'ruf Qomaruddin Kafi – 5002241095

October 6, 2025

Departemen Matematika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

# Daftar Isi

- Pengenalan Pernyataan SWITCH
- Sintaks Dasar SWITCH
- SWITCH dengan Tipe Data Berbeda
- Pentingnya BREAK Statement
- Multiple Case Statements
- Nested SWITCH Statements
- Best Practices SWITCH
- Latihan
- Kesimpulan
- Referensi

# Pengenalan Pernyataan SWITCH

---

# Pernyataan SWITCH dalam Java

## Definisi

Pernyataan `switch` adalah struktur kontrol percabangan yang digunakan ketika kita memiliki banyak kondisi yang identik dan ingin membandingkan suatu ekspresi dengan beberapa nilai konstan.

## Keunggulan SWITCH

- Lebih rapi dan mudah dibaca dibanding multiple if-else
- Efisien untuk pengecekan nilai yang spesifik
- Mendukung berbagai tipe data: `char`, `byte`, `short`, `int`, `String` (JDK 7+)

# Sintaks Dasar SWITCH

---

# Sintaks Dasar Pernyataan SWITCH

## Struktur Dasar SWITCH

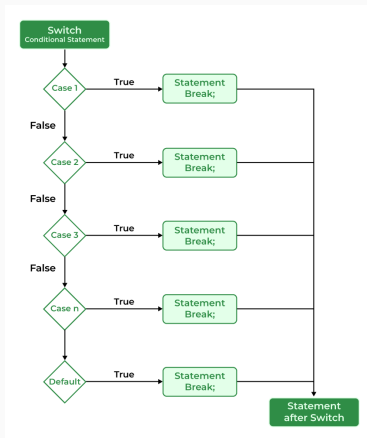
```
1 switch (ekspresi) {  
2     case nilai1:  
3         pernyataan1;  
4         break;  
5     case nilai2:  
6         pernyataan2;  
7         break;  
8     case nilai3:  
9         pernyataan3;  
10        break;  
11    default:  
12        pernyataanDefault;  
13        break;  
14 }  
15
```

# Komponen SWITCH Statement

Komponen	Deskripsi
<code>switch</code> (ekspresi)	Ekspresi yang akan dievaluasi (char, byte, short, int, String)
<code>case nilai:</code>	Nilai konstanta yang dibandingkan dengan ekspresi
<code>pernyataan;</code>	Blok kode yang dieksekusi jika case match
<code>break;</code>	Menghentikan eksekusi dan keluar dari switch
<code>default:</code>	Blok yang dieksekusi jika tidak ada case yang match

**Tabel 1:** Komponen-komponen dalam pernyataan SWITCH

# Flowchart Pernyataan SWITCH



**Gambar 1:** Alur eksekusi pernyataan SWITCH dalam Java

**Sumber:** Programiz - Java switch Statement



## **SWITCH dengan Tipe Data Berbeda**

---

# SWITCH dengan Tipe Data CHAR

## Contoh Program: SWITCH dengan Char

```
1 import java.util.Scanner;
2 // import java.io.IOException; // Untuk alternatif System.in.
   read()
3
4 public class SwitchChar {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         System.out.println("Masukkan inisial nama anda:");
8
9         // Mengambil karakter pertama dari input
10        char inisial = scanner.next().charAt(0);
11        // Bisa juga menggunakan:
12        // char inisial = (char)System.in.read(); (dengan throws
        IOException)
13    }
```

# SWITCH dengan Tipe Data CHAR

## Contoh Program: SWITCH dengan Char

```
1      switch(Character.toLowerCase(inisial)) {  
2          case 'a':  
3              System.out.println("Nama anda pasti Angga");  
4              break;  
5          case 'b':  
6              System.out.println("Nama anda pasti Budi");  
7              break;  
8          case 'c':  
9              System.out.println("Nama anda pasti Cici");  
10             break;  
11         default:  
12             System.out.println("Nama anda tidak terkenal");  
13             break;  
14     }  
15 }  
16 }  
17 }
```

# Penjelasan Input Char

## Cara Kerja `scanner.next().charAt(0)`

- `scanner.next()` membaca input sebagai **String**
- `.charAt(0)` mengambil **karakter pertama** dari string tersebut
- Contoh: Input "Budi" → `charAt(0) = 'B'`

## Alternatif: `System.in.read()`

- Membaca langsung 1 karakter dari input stream
- Butuh `throws IOException` di method signature
- Lebih "pure" untuk input karakter tunggal

# Output SWITCH dengan CHAR

## Contoh Eksekusi Program

```
Masukkan karakter nama anda:  a  
Nama anda pasti Angga
```

## Contoh Lain

```
Masukkan karakter nama anda:  c  
Nama anda tidak terkenal
```

## Catatan Penting

- Gunakan single quote ( ' ' ) untuk karakter
- `System.in.read()` membutuhkan throws `IOException`
- `Character.toLowerCase()` membuat program case-insensitive. 'A' dan 'a' akan diperlakukan sama

# SWITCH dengan Tipe Data INT

## Contoh Program: Konversi Bulan

```
1 public class SwitchInt {
2     public static void main(String[] args) {
3         int bulan = 8;
4         String bulanString;
5
6         switch (bulan) {
7             case 1: bulanString = "Januari"; break;
8             case 2: bulanString = "Februari"; break;
9             case 3: bulanString = "Maret"; break;
10            case 4: bulanString = "April"; break;
11            case 5: bulanString = "Mei"; break;
12            case 6: bulanString = "Juni"; break;
13            case 7: bulanString = "Juli"; break;
14            case 8: bulanString = "Agustus"; break;
15        }
```

# SWITCH dengan Tipe Data INT

## Contoh Program: Konversi Bulan

```
1         case 9: bulanString = "September"; break;
2         case 10: bulanString = "Oktober"; break;
3         case 11: bulanString = "Nopember"; break;
4         case 12: bulanString = "Desember"; break;
5         default: bulanString = "Bulan tidak valid"; break;
6     }
7     System.out.println(bulanString);
8 }
9 }
10
```

# Output SWITCH dengan INT

## Contoh Eksekusi Program

Agustus

## Penjelasan

- Variabel `bulan` bernilai 8
- Program mencari case yang match dengan nilai 8
- Ditemukan di case 8: dan mengeksekusi `bulanString = "Agustus"`
- `break` menghentikan eksekusi ke case berikutnya



# SWITCH dengan Tipe Data STRING (JDK 7+)

## Contoh Program: SWITCH dengan String

```
1 public class SwitchString {  
2     public static void main(String[] args) {  
3         String hari = "Senin";  
4         String jenisHari;  
5  
6         switch (hari.toLowerCase()) {  
7             case "senin":  
8             case "selasa":  
9             case "rabu":  
10            case "kamis":  
11            case "jumat":  
12                jenisHari = "Hari kerja"; break;  
13        }
```

# SWITCH dengan Tipe Data STRING (JDK 7+)

## Contoh Program: SWITCH dengan String

```
1         case "sabtu":  
2         case "minggu":  
3             jenisHari = "Hari libur"; break;  
4         default:  
5             jenisHari = "Hari tidak valid"; break;  
6     }  
7     System.out.println(jenisHari);  
8 }  
9 }  
10
```

## **Pentingnya BREAK Statement**

---

# Konsekuensi Tanpa BREAK

## SWITCH Tanpa BREAK - Fall Through

```
1 public class SwitchNoBreak {
2     public static void main(String[] args) throws IOException {
3         System.out.println("Masukkan karakter nama anda");
4         char inisial = (char)System.in.read();
5
6         switch(inisial) {
7             case 'a': System.out.println("Nama anda pasti Angga")
8             ;
9             case 'b': System.out.println("Nama anda pasti Budi");
10            default: System.out.println("Nama anda tidak terkenal
11            ");
12        }
13    }
```

# Output SWITCH Tanpa BREAK

## Contoh Eksekusi Program

```
Masukkan karakter nama anda:  a
Nama anda pasti Angga
Nama anda pasti Budi
Nama anda tidak terkenal
```

## Fenomena Fall Through

- Tanpa break, eksekusi akan "jatuh" ke case berikutnya
- Semua pernyataan setelah case yang match akan dieksekusi
- Dapat dimanfaatkan untuk multiple case dengan aksi sama

# Multiple Case Statements

---

# Multiple Case untuk Aksi yang Sama

## Contoh Program: Jumlah Hari dalam Bulan

```
1 public class SwitchMultipleCase {  
2     public static void main(String[] args) {  
3         int bulan = 2;  
4         int tahun = 2000;  
5         int jmlHari = 0;  
6  
7         switch (bulan) {  
8             case 1: case 3: case 5: case 7:  
9                 case 8: case 10: case 12:  
10                    jmlHari = 31;  
11                    break;  
12                 case 4: case 6: case 9: case 11:  
13                    jmlHari = 30;  
14                    break;  
15                }
```

# Multiple Case untuk Aksi yang Sama

## Contoh Program: Jumlah Hari dalam Bulan

```
1      case 2:
2          if (((tahun % 4 == 0) && !(tahun % 100 == 0))
3              || (tahun % 400 == 0))
4              jmlHari = 29;
5          else
6              jmlHari = 28;
7          break;
8      default:
9          System.out.println("Bulan tidak valid.");
10         break;
11    }
12    System.out.println("Jumlah hari = " + jmlHari);
13 }
14 }
15
```



## Contoh Eksekusi Program

```
Jumlah hari = 29
```

## Penjelasan

- Bulan 2 (Februari) dengan tahun 2000 (tahun kabisat)
- Tahun kabisat: habis dibagi 4, tapi tidak habis dibagi 100, atau habis dibagi 400
- 2000 habis dibagi 400  $\rightarrow$  tahun kabisat  $\rightarrow$  29 hari

## **Nested SWITCH Statements**

---

# Konsep Nested SWITCH

## SWITCH di dalam SWITCH

- SWITCH statement dapat nested (bersarang)
- Berguna untuk menu bertingkat atau klasifikasi kompleks
- Perlu perhatian ekstra pada break statement

### Contoh Program: Menu Bertingkat

```
1 import java.util.Scanner;
2
3 public class NestedSwitch {
4     public static void main(String[] args) {
5         Scanner baca = new Scanner(System.in);
6
7         System.out.println("Pilih program:");
8         System.out.println("1. Konversi Bulan");
9         System.out.println("2. Cek Hari dalam Bulan");
10    }
```

# Konsep Nested SWITCH

## Contoh Program: Menu Bertingkat

```
1      System.out.print("Masukkan pilihan (1-2): ");
2      int pilihan = baca.nextInt();
3
4      switch(pilihan) {
5          case 1:
6              // switch untuk konversi bulan
7              break;
8          case 2:
9              // switch untuk cek hari
10             break;
11         default:
12             System.out.println("Pilihan tidak valid");
13             break;
14     }
15 }
16 }
17 }
```

# Best Practices SWITCH

---

# Best Practices menggunakan SWITCH

## Tips dan Rekomendasi

- **Selalu gunakan break** kecuali untuk fall through yang disengaja
- **Gunakan default case** untuk menangani nilai tak terduga
- **Group case yang sama** untuk menghindari duplikasi kode
- **Pertimbangkan readability** - jika terlalu kompleks, gunakan if-else

## Kapan Memilih SWITCH vs IF-ELSE?

- **SWITCH:** Untuk equality check dengan nilai konstan yang terbatas
- **IF-ELSE:** Untuk range check, kondisi kompleks, atau banyak kondisi

# Perbandingan SWITCH vs IF-ELSE IF

SWITCH Statement	IF-ELSE IF Statement
Cocok untuk pengecekan nilai tetap	Cocok untuk pengecekan range
Lebih rapi untuk banyak pilihan	Lebih fleksibel untuk kondisi rumit
Lebih cepat untuk banyak case	Mudah ditambah kondisi baru
Perlu break untuk setiap case	Tidak perlu break
Umumnya untuk: char, int, String	Untuk semua tipe data dan ekspresi

**Tabel 2:** Perbandingan SWITCH vs IF-ELSE IF

# Latihan

---



# Latihan 1: Kalkulator dengan Validasi

## Soal 1: Kalkulator dengan Validasi

Buat program kalkulator menggunakan SWITCH yang:

- Meminta input 2 bilangan
- Meminta operator (+, -, \*, /, %)
- Menggunakan SWITCH untuk memilih operasi
- **Fitur validasi:**
  - Untuk operator '/', cek pembagi tidak boleh 0
  - Untuk operator '%', cek pembagi tidak boleh 0
  - Tampilkan pesan error untuk operator tidak valid
- Tampilkan hasil operasi dengan format:  $10 + 5 = 15$

## Latihan 2: Sistem Klasifikasi Nilai

### Soal 2: Konversi dan Klasifikasi Nilai

Buat program yang mengkombinasikan IF dan SWITCH untuk Konversi nilai ke huruf sesuai Peraturan Akademik ITS.

- Nilai angka (86-100) : Nilai A / Istimewa
- Nilai angka (76-85) : Nilai AB / Baik Sekali
- Nilai angka (66-75) : Nilai B / Baik
- Nilai angka (61-65) : Nilai BC / Cukup Baik
- Nilai angka (56-60) : Nilai C / Cukup
- Nilai angka (41-55) : Nilai D / Kurang
- Nilai angka (0-40) : Nilai E / Kurang Sekali

**Input:** Nilai angka (0-100)

## Latihan 3: Kalkulator Geometri Bertingkat

### Soal 3: Kalkulator Geometri Bertingkat

Buat program dengan **nested SWITCH** untuk menghitung bangun ruang:

**Menu Level 1:** Pilih bangun ruang

- 1: Lingkaran
- 2: Tabung
- 3: Kerucut
- 4: Bola

**Menu Level 2:** Setiap bangun memiliki pilihan berbeda:

## Latihan 3: Kalkulator Geometri Bertingkat

- **Tabung:** 1-Volume ( $\pi \times r^2 \times t$ ), 2-Luas Permukaan ( $2 \times \pi \times r \times (r + t)$ )
- **Kerucut:** 1-Volume ( $\frac{1}{3} \times \pi \times r^2 \times t$ ), 2-Luas Permukaan ( $\pi \times r \times (r + s)$ )
- **Bola:** 1-Volume ( $\frac{4}{3} \times \pi \times r^3$ ), 2-Luas Permukaan ( $4 \times \pi \times r^2$ )

### Catatan:

- Gunakan `Math.PI` untuk nilai
- Tampilkan rumus yang digunakan dalam perhitungan

## Kesimpulan

---

## Inti Bab 6: Struktur Kontrol 2 - SWITCH

- **SWITCH** alternatif yang elegan untuk multiple if-else
- Mendukung berbagai tipe data: `char`, `int`, `String`, dan lain-lain
- **BREAK** statement penting untuk mencegah fall through
- **DEFAULT** case menangani nilai tak terduga
- **Multiple case** untuk grouping aksi yang sama
- **Nested switch** untuk menu bertingkat yang kompleks

## Referensi

---

## Referensi:

- **Modul Praktikum Algoritma dan Pemrograman - Modul 6**  
Departemen Matematika FSAD ITS
- **Oracle Java Tutorials - The switch Statement**  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>



## Referensi:

- **Programiz - Java switch Statement**

[https://www.programiz.com/java-programming/  
switch-statement](https://www.programiz.com/java-programming/switch-statement)

- **GeeksforGeeks - Switch Statements in Java**

[https:  
//www.geeksforgeeks.org/switch-statement-in-java/](https://www.geeksforgeeks.org/switch-statement-in-java/)

# Terima Kasih

Pertanyaan dan Diskusi