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

Pernyataan SWITCH dalam Java

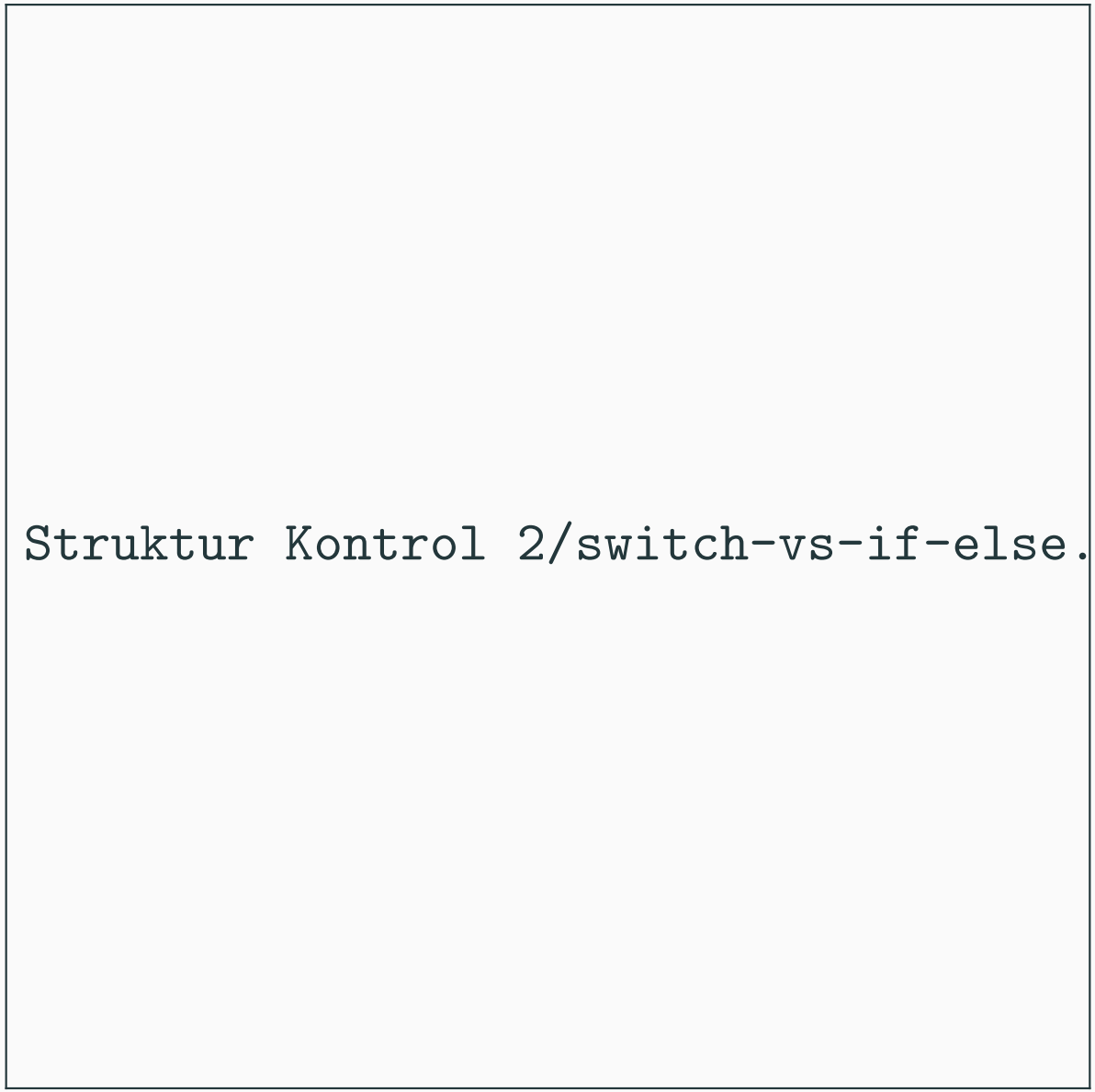
Definisi

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+)

Kapan Menggunakan SWITCH?



Gambar 1: Perbandingan penggunaan SWITCH vs IF-ELSE untuk

Sintaks Dasar SWITCH

Sintaks Dasar Pernyataan SWITCH

Struktur Dasar SWITCH

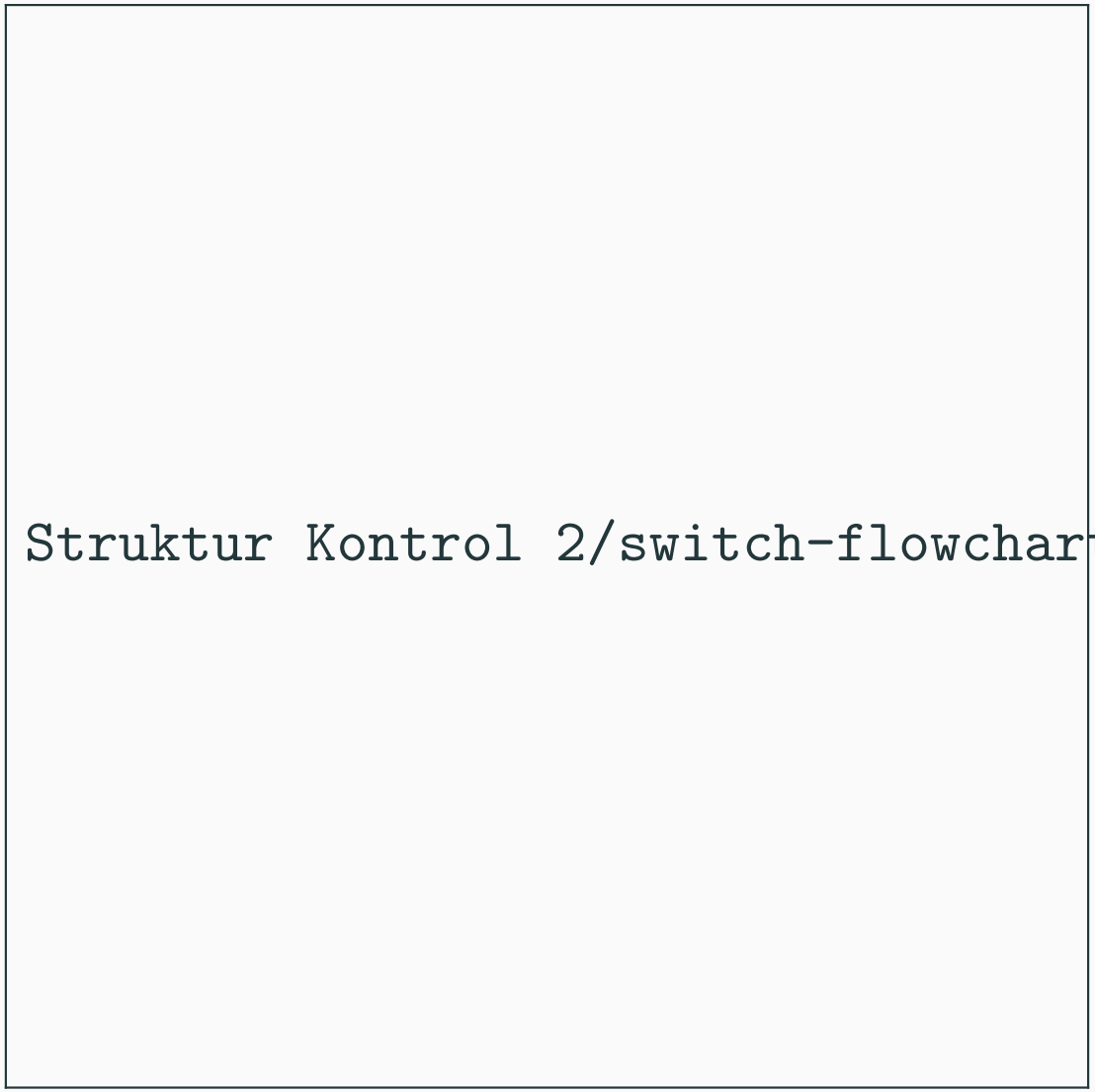
```
switch (ekspresi) {
    case nilai1:
        pernyataan1;
        break;
    case nilai2:
        pernyataan2;
        break;
    case nilai3:
        pernyataan3;
        break;
    default:
        pernyataanDefault;
        break;
}

```

Komponen SWITCH Statement

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

Tabel 1: Komponen-komponen dalam pernyataan SWITCH

Flowchart Pernyataan SWITCH



Gambar 2: Alur eksekusi pernyataan SWITCH dalam Java

SWITCH dengan Tipe Data Berbeda

SWITCH dengan Tipe Data CHAR

Contoh Program: SWITCH dengan Char

```
1  import java.io.IOException;
2
3  public class SwitchChar {
4      public static void main(String[] args) throws IOException {
5          System.out.println("Masukkan karakter nama anda");
6          char inisial = (char)System.in.read();
7
8          switch(inisial) {
9              case 'a':
10                 System.out.println("Nama anda pasti Angga");
11                 break;
12             case 'b':
13                 System.out.println("Nama anda pasti Budi");
14                 break;
15             default:
16                 System.out.println("Nama anda tidak terkenal");
17                 break;
18         }
```

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`
- Case-sensitive: 'A' berbeda dengan 'a'

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             case 9: bulanString = "September"; break;
16             case 10: bulanString = "Oktober"; break;
17             case 11: bulanString = "Nopember"; break;
18             case 12: bulanString = "Desember"; break;
```

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             case "sabtu":
14             case "minggu":
15                 jenisHari = "Hari libur"; break;
16             default:
17                 jenisHari = "Hari tidak valid"; break;
18         }
```

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              case 'b': System.out.println("Nama anda pasti Budi");
9              default: System.out.println("Nama anda tidak terkenal
       ");
10         }
11     }
12 }
13
```

Output SWITCH Tanpa BREAK

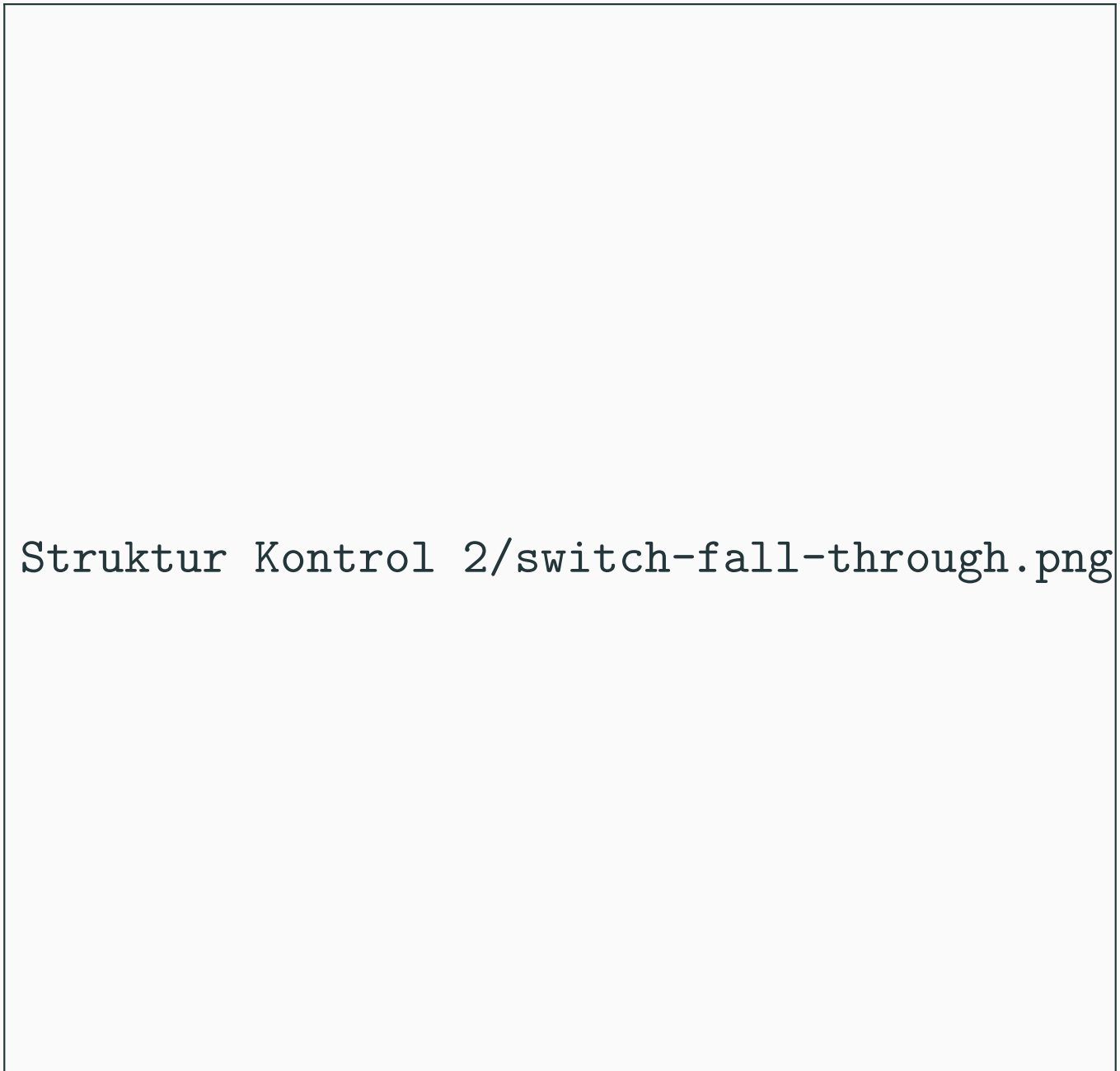
Contoh Eksekusi Program

```
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

Visualisasi Fall Through Behavior



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             case 2:
16                 if (((tahun % 4 == 0) && !(tahun % 100 == 0))
17                     || (tahun % 400 == 0))
18                     jmlHari = 29;
```

Output Multiple Case

Contoh Eksekusi Program

```
Jumlah hari = 29
```

Penjelasan Logic

- 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         System.out.print("Masukkan pilihan (1-2): ");
11
```

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
- **Validasi input** sebelum masuk ke switch statement

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
Equality check dengan konstanta	Range check dan kondisi kompleks
Lebih rapi untuk banyak kondisi	Lebih fleksibel untuk logika kompleks
Potensi lebih efisien	Mudah untuk menambahkan kondisi baru
Harus break secara eksplisit	Tidak perlu break statement
Terbatas pada tipe data tertentu	Mendukung semua tipe data boolean

Tabel 2: Perbandingan SWITCH vs IF-ELSE IF

Latihan

Latihan 1: Kalkulator dengan Validasi

Soal 1: Kalkulator Cerdas

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:

- **Input**: nilai angka (0-100)
- **Step 1**: Konversi ke nilai huruf menggunakan IF-ELSE IF:
  - A: 86-100, AB: 76-85, B: 66-75, BC: 61-65, C: 56-60, D: 41-55, E: 0-40
- **Step 2**: Konversi ke predikat menggunakan SWITCH:
  - A = Luar Biasa, B/AB = Baik, BC/C = Cukup, D/E = Perlu Improvement
- **Output**: `Nilai 85 → AB → Baik`

Latihan 3: Sistem Menu Geometri Bertingkat

Soal 3: Kalkulator Geometri 2 Level

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**: Pilih perhitungan
  - Lingkaran: 1-Luas, 2-Keliling
  - Tabung: 1-Volume, 2-Luas Permukaan
  - Kerucut: 1-Volume, 2-Luas Permukaan
  - Bola: 1-Volume, 2-Luas Permukaan
- **Fitur**:
  - Validasi input menu
  - Gunakan `Math.PI` untuk nilai
  - Tampilkan rumus yang digunakan

Kesimpulan

Kesimpulan

Inti Bab 6: Struktur Kontrol 2 - SWITCH

- **SWITCH** alternatif yang elegan untuk multiple if-else
- Mendukung berbagai tipe data: `char`, `byte`, `short`, `int`, `String`
- **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

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`
- **GeeksforGeeks - Switch Statements in Java**
  `https://www.geeksforgeeks.org/switch-statement-in-java/`

Referensi

Referensi:

- **Programiz - Java switch Statement**
  `https://www.programiz.com/java-programming/switch-statement`
- **Baeldung - Java Switch Statement**
  `https://www.baeldung.com/java-switch`

Terima Kasih

Pertanyaan dan Diskusi