



Algoritma dan Pemrograman Komputer 1

Bab 5: Struktur Kontrol I

Aslam Pandu Tasminto – 5002241025

M. Ma'ruf Qomaruddin Kafi – 5002241095

October 5, 2025

Departemen Matematika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Daftar Isi

Pengenalan Struktur Kontrol

Pernyataan IF

IF-ELSE

IF-ELSE IF-ELSE

IF-ELSE IF-ELSE

Nested IF

Sequential IF

Operator Logika dalam IF

Latihan

Kesimpulan

Referensi

Pengenalan Struktur Kontrol

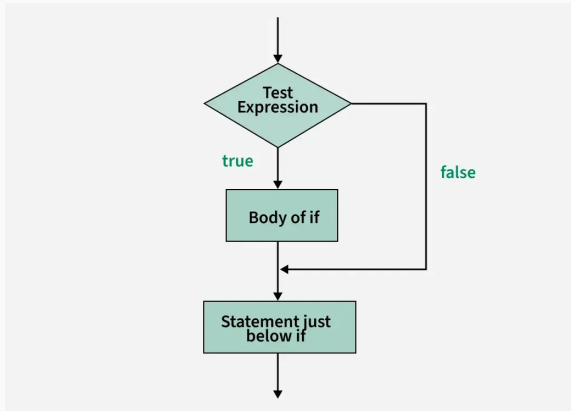
Definisi

Struktur kontrol adalah pernyataan yang digunakan untuk mengatur alur eksekusi program berdasarkan kondisi tertentu.

Jenis Struktur Kontrol

- **Percabangan (Selection):** if, if-else, if-else if-else, switch
- **Perulangan (Looping):** for, while, do-while
- **Pengontrol Alur:** break, continue, return

Pentingnya Struktur Kontrol



Gambar 1: Diagram alur proses pengambilan keputusan dalam programming

Sumber: GeeksforGeeks - Decision Making in Programming

Pernyataan IF

Pernyataan IF Dasar

Sintaks IF Sederhana

```
1 if (kondisi) {  
2     Pernyataan1 // kode yang dijalankan jika kondisi true  
3 }  
4
```

Contoh Program

```
1 public class CekBilangan {  
2     public static void main(String[] args) {  
3         int bilangan = 15;  
4         if (bilangan > 10) {  
5             System.out.println("Bilangan lebih besar dari 10");  
6         }  
7     }  
8 }
```

Contoh Program IF

```
1 import java.util.Scanner;
2
3 public class IfSederhana {
4     public static void main(String[] args) {
5         System.out.println("Masukkan sebuah bilangan:");
6         Scanner baca = new Scanner(System.in);
7         int bil = baca.nextInt();
8
9         if (bil >= 10) {
10             System.out.println("Ini bukan bilangan kecil");
11         }
12         System.out.println("Terima kasih");
13     }
14 }
```


Contoh Eksekusi Program

```
Masukkan sebuah bilangan: 15  
Ini bukan bilangan kecil  
Terima kasih
```

Penjelasan

- User memasukkan bilangan: **15**
- Karena $15 \geq 10$ bernilai **true**
- Program mengeksekusi "Ini bukan bilangan kecil"
- Statement "Terima kasih" selalu dieksekusi (di luar blok if)

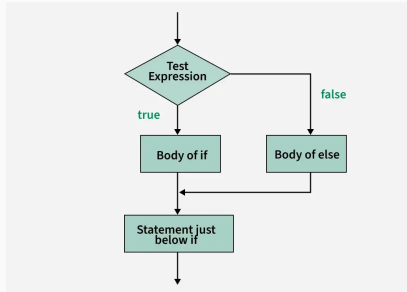
IF-ELSE

Pernyataan IF-ELSE

Sintaks IF-ELSE

```
1 if (kondisi) {  
2     // kode jika kondisi true  
3 } else {  
4     // kode jika kondisi false  
5 }  
6
```

Pernyataan IF-ELSE



Gambar 2: Alur eksekusi pernyataan if-else

Sumber: GeeksforGeeks - Decision Making in Programming

Contoh Program IF-ELSE

```
1 import java.util.Scanner;
2
3 public class CekNilaiAlpro {
4     public static void main(String[] args) {
5         System.out.println("Masukkan nilai Algoritma Pemrograman:
6         ");
7         Scanner baca = new Scanner(System.in);
8         double nilai = baca.nextDouble();
9
10        if (nilai >= 76) {
11            System.out.println("keren banget maniez :)");
12        } else {
13            System.out.println("belajar lagi yaa maniez :(");
14        }
15    }
```

Output Program IF-ELSE

Contoh Eksekusi Program

```
Masukkan nilai Algoritma Pemrograman: 85  
keren banget maniez :)
```

Contoh Lain

```
Masukkan nilai Algoritma Pemrograman: 65  
belajar lagi yaa maniez :(
```

Penjelasan

- Jika nilai ≥ 76 : menampilkan pesan positif dengan emoticon senang
- Jika nilai < 76 : menampilkan pesan motivasi dengan emoticon sedih
- Hanya satu blok yang dieksekusi (if ATAU else)

IF-ELSE IF-ELSE

Pernyataan IF-ELSE IF-ELSE

Sintaks Multi Kondisi

```
1 if (kondisi1) {  
2     // kode jika kondisi1 true  
3 } else if (kondisi2) {  
4     // kode jika kondisi2 true  
5 } else if (kondisi3) {  
6     // kode jika kondisi3 true  
7 } else {  
8     // kode jika semua kondisi false  
9 }  
10
```

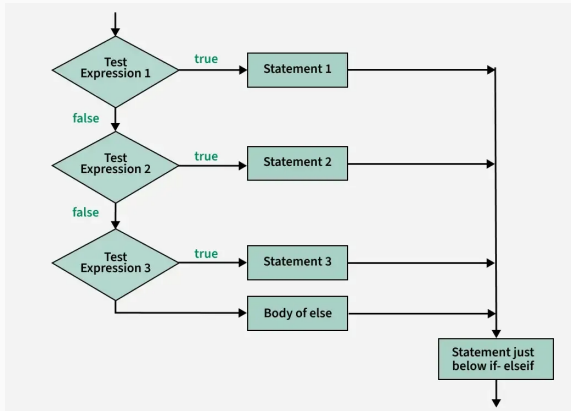

IF-ELSE IF-ELSE

Pernyataan IF-ELSE IF-ELSE

Sintaks Multi Kondisi

```
1 if (kondisi1) {  
2     // kode jika kondisi1 true  
3 } else if (kondisi2) {  
4     // kode jika kondisi2 true  
5 } else if (kondisi3) {  
6     // kode jika kondisi3 true  
7 } else {  
8     // kode jika semua kondisi false  
9 }  
10
```

Flowchart IF-ELSE IF-ELSE



Gambar 3: Alur eksekusi pernyataan IF-ELSE IF-ELSE

Sumber: GeeksforGeeks - Decision Making in Programming

Karakteristik IF-ELSE IF-ELSE

- Dievaluasi secara berurutan dari atas ke bawah
- Hanya satu blok yang akan dieksekusi
- Evaluasi berhenti ketika menemukan kondisi yang true
- Blok `else` bersifat opsional

Contoh Program IF-ELSE IF-ELSE

```
1 import java.util.Scanner;
2
3 public class KonversiNilai {
4     public static void main(String[] args) {
5         System.out.println("Masukkan nilai angka anda:");
6         Scanner baca = new Scanner(System.in);
7         int nilai = baca.nextInt();
8
9         if (nilai >= 86 && nilai <= 100) {
10             System.out.println("Nilai Huruf Anda A / Istimewa");
11         } else if (nilai >= 76 && nilai <= 85) {
12             System.out.println("Nilai Huruf Anda AB / Baik Sekali
13 ");
14         } else if (nilai >= 66 && nilai <= 75) {
15             System.out.println("Nilai Huruf Anda B / Baik");
16         }
17     }
18 }
```

Contoh Program IF-ELSE IF-ELSE

```
1      else if (nilai >= 61 && nilai <= 65) {  
2          System.out.println("Nilai Huruf Anda BC / Cukup Baik"  
3      );  
4      } else if (nilai >= 56 && nilai <= 60) {  
5          System.out.println("Nilai Huruf Anda C / Cukup");  
6      } else if (nilai >= 41 && nilai <= 55) {  
7          System.out.println("Nilai Huruf Anda D / Kurang");  
8      } else if (nilai >= 0 && nilai <= 40) {  
9          System.out.println("Nilai Huruf Anda E / Kurang  
10         Sekali");  
11     } else {  
12         System.out.println("Nilai tidak valid!");  
13     }
```

Tabel Rentang Nilai Berdasarkan Peraturan Akademik 2018

Rentang Nilai	Nilai Huruf	Keterangan
86 - 100	A	Istimewa
76 - 85	AB	Baik Sekali
66 - 75	B	Baik
61 - 65	BC	Cukup Baik
56 - 60	C	Cukup
41 - 55	D	Kurang
0 - 40	E	Kurang Sekali

Tabel 1: Tabel konversi nilai berdasarkan Peraturan Akademik 2018

Output Program IF-ELSE IF-ELSE

Contoh Eksekusi Program

Masukkan nilai angka anda: 92

Nilai Huruf Anda A / Istimewa

Contoh Lain

Masukkan nilai angka anda: 78

Nilai Huruf Anda AB / Baik Sekali

Masukkan nilai angka anda: 63

Nilai Huruf Anda BC / Cukup Baik

Masukkan nilai angka anda: 35

Nilai Huruf Anda E / Kurang Sekali

Nested IF

Nested IF (IF Bersarang)

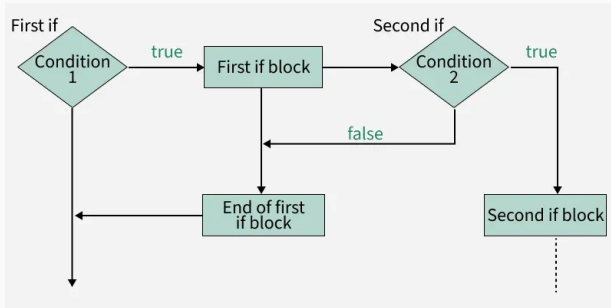
Konsep Nested IF

- IF di dalam IF
- Digunakan untuk kondisi yang kompleks
- Setiap kondisi harus dipenuhi secara berurutan

Sintaks Nested IF

```
1 if (kondisi1) {  
2     if (kondisi2) {  
3         // kode jika kedua kondisi true  
4     }  
5 }  
6
```

Flowchart Nested IF



Gambar 4: Alur eksekusi Nested IF (IF Bersarang)

Sumber: GeeksforGeeks - Decision Making in Programming

Karakteristik Nested IF

- Kondisi dievaluasi secara bertingkat
- Kondisi IF ke-2 hanya dieksekusi jika kondisi IF ke-1 bernilai true
- Cocok untuk validasi bertingkat

Contoh Program Nested IF

```
1 import java.util.Scanner;
2
3 public class MenuProgram {
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. Cek Bilangan");
9         System.out.println("2. Cek Nilai");
10        System.out.println("3. Keluar");
11        System.out.print("Masukkan pilihan (1-3): ");
12
13        int pilihan = baca.nextInt();
14
15        if (pilihan == 1) {
16            System.out.print("Masukkan bilangan: ");
17            int bil = baca.nextInt();
```

Contoh Program Nested IF

```
1      if (bil >= 10) {  
2          System.out.println("Bilangan BESAR");  
3      } else {  
4          System.out.println("Bilangan KECIL");  
5      }  
6  
7  } else if (pilihan == 2) {  
8      System.out.print("Masukkan nilai: ");  
9      double nilai = baca.nextDouble();  
10  
11     if (nilai >= 76) {  
12         System.out.println("LULUS - Keren!");  
13     } else {  
14         System.out.println("REMIDIAL - Semangat!");  
15     }
```

Contoh Program Nested IF

```
1      } else if (pilihan == 3) {  
2          System.out.println("Terima kasih!");  
3  
4      } else {  
5          System.out.println("Pilihan tidak valid!");  
6      }  
7  }  
8  }
```

Output Program Nested IF

Contoh 1: Cek Bilangan

Pilih program:

1. Cek Bilangan
2. Cek Nilai
3. Keluar

Masukkan pilihan (1-3): 1

Masukkan bilangan: 15

Bilangan BESAR

Contoh 2: Cek Nilai

Pilih program:

1. Cek Bilangan
2. Cek Nilai
3. Keluar

Masukkan pilihan (1-3): 2

Masukkan nilai: 65

REMIDIAL - Semangat!

Output Program Nested IF

Contoh 3: Keluar Program

```
Pilih program:  
1. Cek Bilangan  
2. Cek Nilai  
3. Keluar  
Masukkan pilihan (1-3): 3  
Terima kasih!
```

Contoh 4: Pilihan Invalid

```
Pilih program:  
1. Cek Bilangan  
2. Cek Nilai  
3. Keluar  
Masukkan pilihan (1-3): 5  
Pilihan tidak valid!
```

Penjelasan Output Program Nested IF

Penjelasan

- **IF 1:** Validasi pilihan menu (1, 2, 3)
- **IF 2:** Jika pilihan valid, eksekusi secara spesifik
- Setiap menu memiliki nested if untuk logika internal

Sequential IF

Sequential IF (IF Bersebelahan)

Konsep IF Bersebelahan

- Beberapa pernyataan IF yang berurutan
- Setiap IF dievaluasi secara independen
- Bisa beberapa kondisi yang true sekaligus

Contoh Program IF Bersebelahan

```
1 import java.util.Scanner;
2
3 public class ValidasiInput {
4     public static void main(String[] args) {
5         System.out.println("Masukkan nilai (0-100):");
6         Scanner baca = new Scanner(System.in);
7         int bilangan = baca.nextInt();
8
9         // IF bersebelahan - semua dievaluasi
10        if (bilangan > 100) {
11            System.out.println("Nilai terlalu besar");
12        }
```

Contoh Program IF Bersebelahan

```
1      if (bilangan < 0) {  
2          System.out.println("Nilai terlalu kecil");  
3      }  
4      if (bilangan >= 0 && bilangan <= 100) {  
5          System.out.println("Nilai sesuai");  
6      }  
7  }  
8  }
```

Perbandingan IF Bertingkat vs IF Bersebelahan

IF Bertingkat	IF Bersebelahan
Hanya satu blok yang dieksekusi	Banyak blok bisa dieksekusi
Evaluasi berhenti saat kondisi true	Semua kondisi dievaluasi
Efisien untuk pilihan bergantian	Fleksibel untuk pengecekan terpisah
Cocok untuk klasifikasi	Cocok untuk validasi

Tabel 2: Perbandingan IF bertingkat dan IF bersebelahan

Operator Logika dalam IF

Operator Logika untuk Kondisi Kompleks

Operator Logika yang Sering Digunakan

```
1 // AND (&&) - kedua kondisi harus true
2 if (nilai >= 60 && nilai <= 100) {
3     System.out.println("Lulus");
4 }
5
6 // OR (||) - salah satu kondisi true
7 if (hari == "Sabtu" || hari == "Minggu") {
8     System.out.println("Weekend");
9 }
10
11 // NOT (!) - negasi
12 if (!isTamat) {
13     System.out.println("Belum tamat");
14 }
15
```

Tabel Operator Logika

Operator	Contoh	Deskripsi
&&	a && b	True jika a DAN b true
	a b	True jika a ATAU b true
!	!a	True jika a false, dan sebaliknya
==	a == b	True jika a sama dengan b
!=	a != b	True jika a tidak sama dengan b
>, <	a > b	Membandingkan nilai lebih besar/lebih kecil
>=, <=	a >= b	Membandingkan nilai lebih besar/kecil sama dengan

Tabel 3: Operator logika dan perbandingan dalam Java

Latihan

Latihan 1: Penentu Kelulusan

Soal 1: Penentu Kelulusan

Buat program yang meminta input:

- Nilai ujian (0-100)
- Kehadiran (persentase 0-100%)

Kelulusan ditentukan oleh:

- Lulus jika: nilai ≥ 60 DAN kehadiran $\geq 75\%$
- Remedial jika: nilai ≥ 40 DAN kehadiran $\geq 75\%$
- Tidak lulus untuk kondisi lainnya

Latihan 2: Kategori Usia

Soal 2: Kategori Usia

Buat program yang mengkategorikan usia:

- Anak-anak: 0-12 tahun
- Remaja: 13-17 tahun
- Dewasa muda: 18-25 tahun
- Dewasa: 26-45 tahun
- Lansia: >45 tahun

Gunakan struktur if-else if-else!

Latihan 3: Kalkulator Sederhana

Soal 3: Kalkulator Sederhana

Buat program kalkulator yang:

- Meminta input 2 bilangan
- Meminta operator (+, -, *, /, %)
- Menggunakan nested if untuk validasi:
 - Jika operator '/', cek pembagi tidak boleh 0
 - Jika operator tidak valid, tampilkan pesan error
- Tampilkan hasil operasi

Kesimpulan

Inti Bab 5: Struktur Kontrol I

- **IF** digunakan untuk eksekusi kode berdasarkan kondisi boolean
- **IF-ELSE** memberikan alternatif ketika kondisi false
- **IF-ELSE IF-ELSE** menangani multiple kondisi secara berurutan
- **Nested IF** untuk kondisi yang kompleks dan bertingkat
- **Sequential IF** untuk kondisi yang independen dan perlu semua dievaluasi
- Pemilihan struktur kontrol tergantung pada logika program yang dibutuhkan

Referensi

Referensi:

- **Modul Praktikum Algoritma dan Pemrograman**
Departemen Matematika FSAD ITS
- **Oracle Java Tutorials - Control Flow Statements**
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/flow.html>
- **GeeksforGeeks - Decision Making in Java**
<https://www.geeksforgeeks.org/decision-making-java-if-else-switch-break-continue-jump>

Terima Kasih

Pertanyaan dan Diskusi