

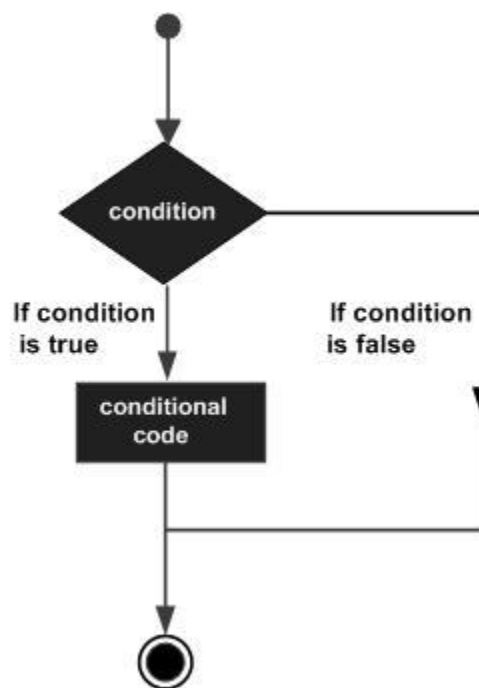
## MODUL 3

### PEMROGRAMAN JAVA DASAR II

#### 3.1 Struktur Kontrol

Dalam pemrograman, pengambilan keputusan digunakan untuk mengontrol aliran eksekusi berdasarkan kondisi tertentu. Java menyediakan berbagai struktur kontrol untuk pengambilan keputusan, seperti if, if-else, switch, dan lainnya.

Berikut ini adalah bentuk umum dari struktur pengambilan keputusan yang umum ditemukan di sebagian besar bahasa pemrograman:



Bahasa pemrograman Java menyediakan beberapa jenis pernyataan pengambilan keputusan. Berikut adalah jenis-jenis pengambilan keputusan dalam Java:

No.	Pernyataan	Deskripsi
1	<b>if</b> statement	Pernyataan <b>if</b> terdiri dari sebuah ekspresi boolean yang diikuti oleh satu atau lebih pernyataan. Jika ekspresi bernilai true, maka blok kode akan dieksekusi.
2	<b>if...else</b> statement	Pernyataan <b>if</b> dapat diikuti oleh pernyataan <b>else</b> opsional, yang akan dieksekusi jika ekspresi boolean bernilai false.
3	nested <b>if</b> statement	Pernyataan <b>if</b> atau <b>else if</b> dapat ditempatkan di dalam pernyataan <b>if</b> atau <b>else if</b> lainnya untuk membuat keputusan yang lebih kompleks.
4	<b>switch</b> statement	Pernyataan <b>switch</b> memungkinkan sebuah variabel diuji untuk kesamaan terhadap daftar nilai tertentu.

		Setiap nilai dibandingkan dalam case-case yang berbeda, dan jika cocok, blok kode yang sesuai akan dijalankan.
--	--	--

Setiap pernyataan ini digunakan untuk mengontrol alur eksekusi program berdasarkan kondisi tertentu.

#### a. If-else

Di Java, pernyataan if else digunakan untuk mengeksekusi dua blok kode berdasarkan kondisi yang diberikan. Pernyataan if dieksekusi ketika ekspresi Boolean untuk pernyataan if bernilai benar. Pernyataan if dapat diikuti dengan pernyataan else opsional, yang dijalankan ketika ekspresi Boolean bernilai salah.

Sintaks dasar if-else:

```
if (Boolean_expression) {
    // Executes when the Boolean expression is true
} else {
    // Executes when the Boolean expression is false
}
```

Jika ekspresi boolean bernilai benar, maka blok kode if akan dieksekusi, jika tidak, blok kode else akan dieksekusi.

Contoh:

```
boolean isJavaFun = true;

if (isJavaFun) {
    System.out.println("Ini adalah if statement");
} else {
    System.out.println("Ini adalah else statement");
}

// Output: "Ini adalah if statement"
```

Selanjutnya ada pernyataan `if...else if...else` digunakan untuk mengeksekusi beberapa blok kode berdasarkan kondisi yang diberikan (ekspresi Boolean).

Pernyataan `if` dapat diikuti dengan pernyataan `else if...else` yang bersifat opsional, yang sangat berguna untuk menguji berbagai kondisi dengan menggunakan satu pernyataan `if...else if`.

Contoh:

```
int a = 10;

if (a > 0) {
    System.out.println("a adalah bilangan positif");
} else if (a < 0) {
    System.out.println("a adalah bilangan negatif");
} else {
    System.out.println("a adalah nol");
}

// Output: "a adalah bilangan positif"
```

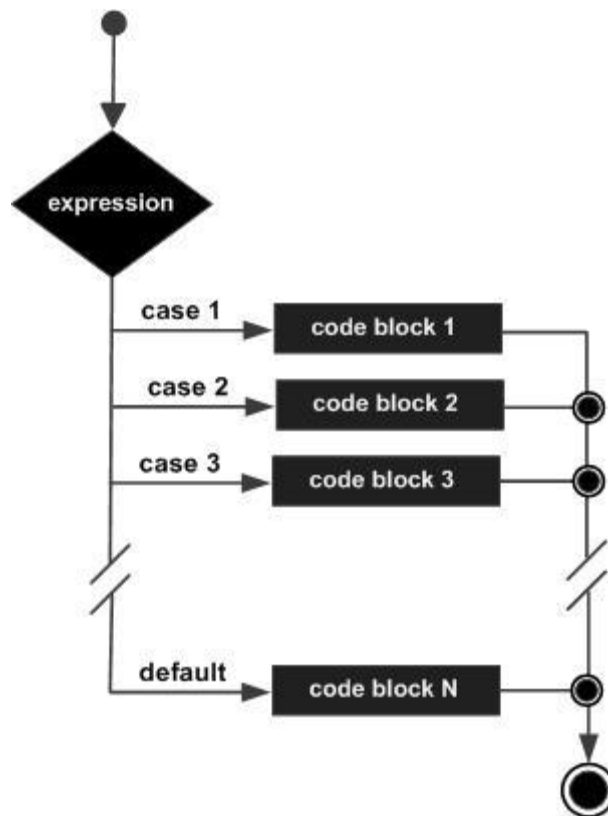
Bentuk if else lain adalah **nested if else (percabangan bersarang)** yang digunakan untuk pengambilan keputusan yang lebih baik ketika kondisi lain harus diperiksa ketika kondisi yang diberikan benar. Dalam pernyataan bersarang if else, kita dapat memiliki pernyataan if-else yang memblokir blok if (atau, else) yang lain.

Contoh:

```
int nilai = 75;
if (nilai >= 80) {
    System.out.println("Nilai A");
    if (nilai == 100) {
        System.out.println("Nilai sempurna!");
    }
} else if (nilai >= 60) {
    System.out.println("Nilai B");
} else if (nilai >= 40) {
    System.out.println("Nilai C");
} else {
    System.out.println("Nilai D");
}
```

## b. Switch-case

Pernyataan switch di Java digunakan untuk menguji nilai suatu variabel terhadap sejumlah nilai yang telah ditentukan. Setiap nilai yang dibandingkan disebut sebagai case, dan jika ada yang cocok, blok kode yang sesuai akan dieksekusi.



Cara kerja switch:

- 1) Variabel yang akan diuji ditempatkan di dalam switch.
- 2) Setiap kemungkinan nilai yang diuji ditentukan dalam case.
- 3) Jika nilai variabel cocok dengan salah satu case, kode dalam blok tersebut akan dijalankan.
- 4) Jika tidak ada yang cocok, blok default (jika ada) akan dieksekusi.
- 5) Gunakan break untuk mencegah eksekusi berlanjut ke case berikutnya setelah menemukan kecocokan.

Sintaks dasar:

```
switch(expression) {
    case value :
        // Statements
        break; // optional

    case value :
        // Statements
        break; // optional

    // You can have any number of case statements.
    default : // Optional
        // Statements
}
```

Contoh:

```
// Switch Case
int day = 1;
switch (day) {
    case 1:
        System.out.println("Senin");
        break;
    case 2:
        System.out.println("Selasa");
        break;
    case 3:
        System.out.println("Rabu");
        break;
    case 4:
        System.out.println("Kamis");
        break;
    case 5:
        System.out.println("Jumat");
        break;
    case 6:
        System.out.println("Sabtu");
        break;
    case 7:
        System.out.println("Minggu");
        break;
    default:
        System.out.println("Hari tidak valid");
        break;
}
```

Switch bisa digunakan dengan tipe data berikut:

- byte, short, char, int
- String (sejak Java 7)
- enum

Kapan Menggunakan switch daripada if-else?

**Gunakan switch jika:**

- Ada banyak kondisi dengan nilai tetap.
- Variabel yang diuji adalah int, char, String, atau enum.

**Gunakan if-else jika:**

- Kondisi melibatkan range nilai (misalnya  $x > 10$ ).
- Perlu mengevaluasi ekspresi logis yang kompleks.

### c. Ternary Operator

Ternary operator adalah operator kondisional singkat dalam Java yang digunakan untuk menggantikan `if-else` dalam ekspresi sederhana. Operator ini dikenal sebagai conditional operator dan menggunakan simbol `?:`.

Sintaks:

```
variabel = (kondisi) ? nilai_jika_true : nilai_jika_false;
```

- `kondisi` → ekspresi boolean (true atau false).
- `nilai_jika_true` → akan dieksekusi jika kondisi bernilai true.
- `nilai_jika_false` → akan dieksekusi jika kondisi bernilai false.

Contoh:

```
int angka = 10;  
String hasil = (angka % 2 == 0) ? "Genap" : "Ganjil";  
  
System.out.println("Angka " + angka + " adalah " + hasil);
```

## 3.2 Perulangan / Looping

Looping adalah proses menjalankan perintah yang sama berulang kali selama suatu kondisi tertentu terpenuhi. Dalam Java, terdapat empat jenis perulangan:

- for loop
- while loop
- do-while loop
- foreach

### a. For loop

Digunakan ketika jumlah iterasi sudah diketahui sebelumnya.

Sintaks:

```
for (inisialisasi; kondisi; increment/decrement) {  
    // kode yang akan dieksekusi  
}
```

- Inisialisasi → Nilai awal variabel kontrol.
- Kondisi → Loop akan terus berjalan selama kondisi bernilai true.
- Increment/Decrement → Perubahan nilai variabel kontrol setiap iterasi.

Contoh:

```
for (int i = 0; i < 5; i++) {  
    System.out.println("Perulangan ke-" + (i + 1));  
}
```

Output:

Perulangan ke-1

Perulangan ke-2

Perulangan ke-3

Perulangan ke-4

Perulangan ke-5

## b. While loop

Digunakan ketika jumlah iterasi tidak diketahui sebelumnya dan bergantung pada suatu kondisi.

Sintaks:

```
while (kondisi) {  
    // kode yang akan dieksekusi selama kondisi true  
}
```

- Loop akan terus berjalan selama kondisi masih true.
- Cocok digunakan jika jumlah iterasi tidak pasti.

Contoh:

```
int j = 0;  
while(j < 5) {  
    System.out.println("Perulangan ke-" + (j + 1));  
    j++;  
}
```

## c. Do-while loop

Mirip dengan while, tetapi pasti dijalankan minimal sekali, karena pengecekan kondisi dilakukan setelah menjalankan kode.

Sintaks:

```
do {  
    // kode yang akan dieksekusi  
} while (kondisi);
```

- Kode dalam do akan selalu dijalankan minimal sekali, meskipun kondisi awalnya false.

Contoh:

```
int k = 0;
do {
    System.out.println("Perulangan ke-" + (k + 1));
    k++;
} while (k < 5);
```

#### d. For-each loop

Digunakan untuk iterasi pada array atau Collection seperti ArrayList.

Sintaks:

```
for (tipe_data variabel : koleksi) {
    // kode yang akan dieksekusi
}
```

Contoh:

```
int[] numbers = { 1, 2, 3, 4, 5 };

for (int num : numbers) {
    System.out.println("numbers: " + num);
}

// Output:
// numbers: 1
// numbers: 2
// numbers: 3
// numbers: 4
// numbers: 5
```

#### e. `break` dan `continue` dalam Looping

##### 1) Break

Digunakan untuk menghentikan perulangan sebelum kondisi selesai.

Contoh:

```
for (int i = 1; i <= 5; i++) {
    if (i == 3) {
        break; // Loop berhenti saat i = 3
    }
}
```



```

        System.out.println("Iterasi ke-" + i);
    }

    // Output:
    // Iterasi ke-1
    // Iterasi ke-2

```

## 2) Continue

Digunakan untuk melewati iterasi tertentu tanpa menghentikan loop.

Contoh:

```

for (int i = 1; i <= 5; i++) {
    if (i == 3) {
        continue; // Melewati iterasi saat i = 3
    }
    System.out.println("Iterasi ke-" + i);
}

// Output:
// Iterasi ke-1
// Iterasi ke-2
// Iterasi ke-4
// Iterasi ke-5

```

## f. Kesimpulan

Jenis Looping	Kapan Digunakan?
for	Jika jumlah iterasi sudah diketahui sebelumnya.
while	Jika jumlah iterasi tidak pasti, tetapi bergantung pada kondisi.
do-while	Jika ingin menjalankan kode minimal sekali sebelum mengecek kondisi.
for-each	Jika ingin mengakses elemen dari array atau Collection.

## 3.3 Array dan String

### a. Pengantar Array

Array adalah struktur data yang menyimpan sekumpulan elemen dengan tipe data yang sama dalam satu variabel. Setiap elemen dalam array memiliki indeks yang dimulai dari 0.

Contoh deklarasi dan inisialisasi array di Java:

```

int[] angka = { 1, 2, 3, 4, 5 }; // Deklarasi dan inisialisasi langsung

```

```
int[] data = new int[5]; // Array dengan ukuran 5, nilai awal default 0
```

## b. Operasi dasar pada Array

### 1) Akses Elemen Array

```
System.out.println(angka[0]); // Output: 1
```

### 2) Mengubah Nilai Elemen

```
angka[2] = 10; // Mengubah elemen indeks ke-2 menjadi 10
```

### 3) Menampilkan Semua Elemen

```
for (int num : angka) {  
    System.out.print(num + " ");  
}
```

### 4) Menghitung Panjang Array

```
System.out.println(angka.length); // Output: 5
```

## c. String dan Metode String

String adalah objek di Java yang mewakili sekumpulan karakter. String bersifat immutable (tidak bisa diubah setelah dibuat).

Contoh deklarasi String:

```
String teks = "Halo, Java!";
```

Beberapa metode umum pada String:

### 1) **.length()** → Mengembalikan panjang string

Contoh:

```
String teks = "Halo, Java!";  
System.out.println(teks.length()); // Output: 11
```

### 2) **.charAt(index)** → Mengambil karakter berdasarkan indeks

Contoh:

```
System.out.println(teks.charAt(1)); // Output: a
```

- 3) **.toUpperCase()** dan **.toLowerCase()** → Mengubah huruf besar/kecil

Contoh:

```
System.out.println(teks.toUpperCase()); // Output: HALO, JAVA!
```

- 4) **.substring(start, end)** → Mengambil sebagian string

Contoh:

```
System.out.println(teks.substring(0, 4)); // Output: Halo
```

#### d. Manipulasi String

Manipulasi string dilakukan dengan berbagai metode bawaan, seperti:

- 1) Menggabungkan String (concat)

```
String s1 = "Hello";  
String s2 = "World";  
System.out.println(s1.concat(" " + s2)); // Output: Hello  
World
```

- 2) Mengganti karakter atau kata (replace)

```
String teks = "Java Programming";  
System.out.println(teks.replace("Java", "Python")); //  
Output: Python Programming
```

- 3) Memisahkan String (split)

```
String data = "apel,mangga,jeruk";  
String[] buah = data.split(",");  
System.out.println(buah[1]); // Output: mangga
```

#### Latihan

Kerjakan soal dibawah ini!

Latihan	
Durasi Pengerjaan	30 menit
Start	16:00
End	16:30
Soal	Buat program yang meminta pengguna memasukkan sebuah kalimat. Program harus:

	<ul style="list-style-type: none"> <li>• Menghitung jumlah huruf vokal (a, i, u, e, o) dalam kalimat tersebut menggunakan if-else.</li> <li>• Menggunakan switch-case untuk menentukan apakah jumlah huruf vokal tersebut ganjil atau genap.</li> <li>• Jika jumlahnya genap, tampilkan "Jumlah vokal genap"; jika ganjil, tampilkan "Jumlah vokal ganjil".</li> <li>• Gunakan foreach untuk menampilkan setiap karakter dalam string secara terbalik.</li> </ul>
Link Pengumpulan	

### **Posttest**

Kerjakan soal dibawah ini!

<b>Pengerjaan Posttest</b>	
Durasi Pengerjaan	30 menit
Start	16:10
End	16:40
Soal	<p>Sebuah toko online memiliki sistem sederhana untuk menghitung diskon berdasarkan total belanja. Toko ini memiliki aturan sebagai berikut:</p> <ul style="list-style-type: none"> <li>• Jika total belanja lebih dari 500.000, pelanggan mendapatkan diskon 20%.</li> <li>• Jika total belanja antara 250.000 hingga 500.000, pelanggan mendapatkan diskon 10%.</li> <li>• Jika total belanja kurang dari 250.000, pelanggan tidak mendapatkan diskon.</li> </ul> <p>Toko ini juga memiliki sistem member dengan kategori berikut:</p> <ul style="list-style-type: none"> <li>• "Platinum" mendapatkan tambahan diskon 5% dari total belanja setelah diskon utama.</li> <li>• "Gold" mendapatkan tambahan diskon 3% dari total belanja setelah diskon utama.</li> <li>• "Silver" mendapatkan tambahan diskon 2% dari total belanja setelah diskon utama.</li> <li>• Jika bukan member, tidak ada tambahan diskon.</li> </ul> <p>Tugas Anda:</p> <ul style="list-style-type: none"> <li>• Buat program yang meminta input total belanja dan tipe member pelanggan.</li> <li>• Gunakan if-else atau switch case untuk menentukan diskon utama dan tambahan.</li> <li>• Gunakan ternary operator untuk menampilkan apakah pelanggan mendapatkan diskon atau tidak.</li> <li>• Pastikan program bisa menangani input yang tidak valid dengan memberikan pesan error.</li> </ul>
Link Pengumpulan	