

## Modul 5 Selection

### Tugas Pendahuluan :

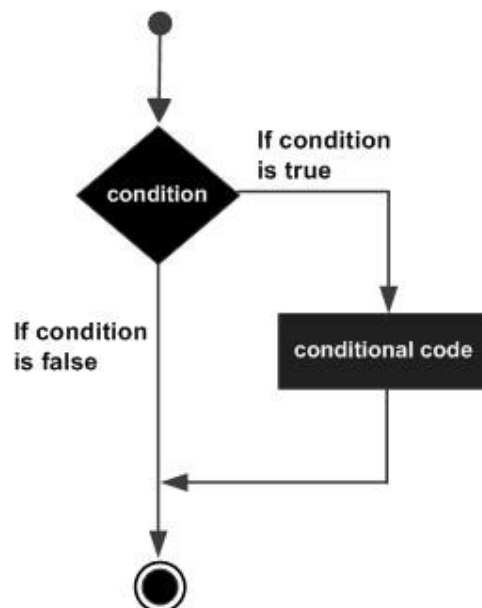
1. Jelaskan apa yang dimaksud dengan struktur seleksi (selection structure) dalam pemrograman.
2. Apa perbedaan antara pernyataan if dan operator ternary ? :? Berikan contoh penggunaan keduanya.
3. Jelaskan cara kerja pernyataan if-else dalam pemrograman. Berikan contoh penggunaan pernyataan if-else dalam program Java.
4. Jelaskan penggunaan operator ternary ? : sebagai alternatif untuk pernyataan if-else. Kapan sebaiknya kita menggunakan operator ternary?
5. Apa keuntungan menggunakan operator ternary ? : dibandingkan dengan pernyataan if-else? Berikan alasan mengapa kita harus menggunakan operator ternary dalam beberapa situasi.
6. Bagaimana cara kerja pernyataan switch dalam pemrograman? Berikan contoh penggunaan pernyataan switch dalam program Java.

### Materi

Keputusan dalam Java dinyatakan melalui pernyataan kondisional atau pernyataan percabangan. Dengan menggunakan pernyataan-pernyataan ini, program dapat mengambil keputusan berdasarkan kondisi yang diberikan. Berikut adalah beberapa konsep dasar dalam pengambilan keputusan dalam Java:

#### 1. Pernyataan if:

Pernyataan if digunakan untuk menjalankan blok kode tertentu jika kondisi yang diberikan benar (true). Jika kondisi bernilai salah (false), blok kode tersebut akan diabaikan. Perhatikan flowchart berikut :



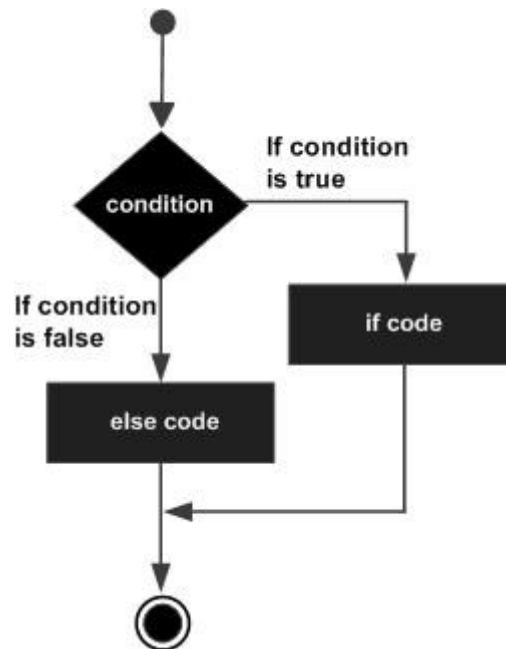
Gambar 1. Flowchart pernyataan if

Contoh penggunaan pernyataan if:

```
int x = 10;
if (x > 5) {
    System.out.println("Nilai x lebih besar dari 5");
}
```

## 2. Pernyataan if-else:

Pernyataan if-else digunakan untuk menjalankan blok kode tertentu jika kondisi yang diberikan benar (true), dan menjalankan blok kode lain jika kondisi tersebut salah (false). Flowchart untuk if-else sebagai berikut :



Gambar 2. Pernyataan if..else

Contoh penggunaan pernyataan if-else:

```
int x = 10;
if (x > 5) {
    System.out.println("Nilai x lebih besar dari 5");
} else {
    System.out.println("Nilai x kurang dari atau sama dengan 5");
}
```

## 3. Pernyataan if-else if-else:

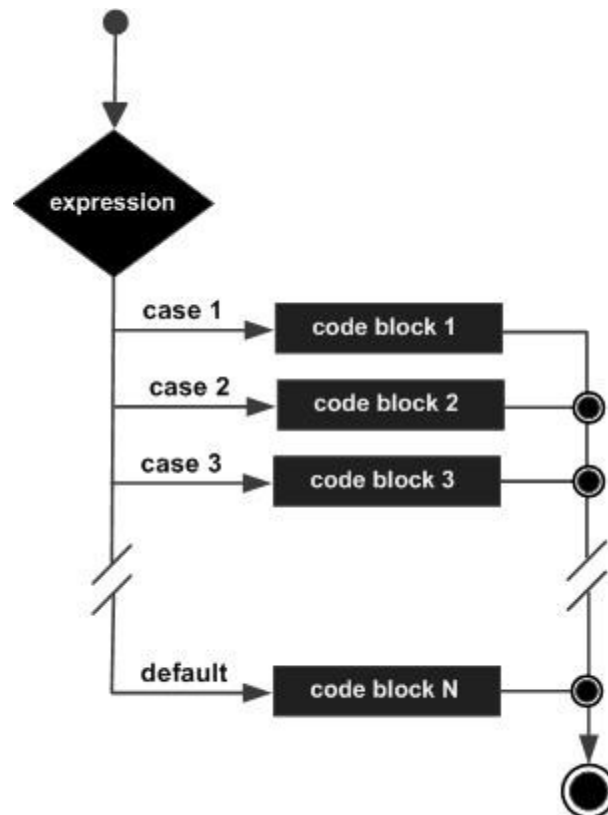
Pernyataan if-else if-else digunakan ketika ada beberapa kondisi yang harus diperiksa secara berurutan. Program akan mengevaluasi kondisi satu per satu dan menjalankan blok kode yang sesuai dengan kondisi pertama yang bernilai benar.

Contoh penggunaan pernyataan if-else if-else:

```
int x = 10;
if (x > 10) {
    System.out.println("Nilai x lebih besar dari 10");
} else if (x < 10) {
    System.out.println("Nilai x kurang dari 10");
} else {
    System.out.println("Nilai x sama dengan 10");
}
```

#### 4. Pernyataan switch:

Pernyataan switch digunakan ketika terdapat banyak kemungkinan nilai yang harus dibandingkan dengan satu variabel. Program akan mengevaluasi nilai variabel dan menjalankan blok kode yang sesuai dengan nilai tersebut. Berikut adalah flowchart untuk pernyataan Switch.



Gambar 3. Pernyataan Switch

Contoh penggunaan pernyataan switch:

```
int x = 2;
switch (x) {
    case 1:
        System.out.println("Nilai x sama dengan 1");
        break;
    case 2:
```

```

        System.out.println("Nilai x sama dengan 2");
        break;
    case 3:
        System.out.println("Nilai x sama dengan 3");
        break;
    default:
        System.out.println("Nilai x tidak cocok dengan kondisi yang ada");
        break;
}

```

Dalam pengambilan keputusan, penting untuk memperhatikan struktur logika yang benar dan menyediakan semua kemungkinan kondisi yang mungkin terjadi. Pastikan juga menggunakan operator logika dan operator perbandingan dengan tepat untuk menghasilkan hasil yang diharapkan.

Berikut adalah contoh program Java yang menggunakan beberapa pernyataan if dalam satu program:

```

1. import java.util.Scanner;
2.
3. public class KeputusanIf {
4.     public static void main(String[] args) {
5.         Scanner input = new Scanner(System.in);
6.
7.         System.out.print("Masukkan angka: ");
8.         int angka = input.nextInt();
9.
10.        // Pernyataan if pertama
11.        if (angka > 0) {
12.            System.out.println("Angka yang dimasukkan positif");
13.        }
14.
15.        // Pernyataan if kedua
16.        if (angka % 2 == 0) {
17.            System.out.println("Angka yang dimasukkan adalah bilangan genap");
18.        } else {
19.            System.out.println("Angka yang dimasukkan adalah bilangan ganjil");
20.        }
21.
22.        // Pernyataan if ketiga
23.        if (angka > 10) {
24.            System.out.println("Angka yang dimasukkan lebih dari 10");
25.        } else if (angka < 10) {
26.            System.out.println("Angka yang dimasukkan kurang dari 10");
27.        } else {
28.            System.out.println("Angka yang dimasukkan adalah 10");
29.        }
30.    }
31. }
32.

```

Pada contoh program di atas, terdapat beberapa pernyataan if yang digunakan untuk mengambil keputusan berdasarkan angka yang dimasukkan oleh pengguna. Pernyataan if pertama digunakan untuk

memeriksa apakah angka yang dimasukkan lebih besar dari 0. Pernyataan if kedua digunakan untuk memeriksa apakah angka yang dimasukkan merupakan bilangan genap atau ganjil menggunakan operator modulo %. Pernyataan if ketiga digunakan untuk membandingkan angka yang dimasukkan dengan angka 10 menggunakan operator perbandingan.

Berdasarkan angka yang dimasukkan oleh pengguna, program akan mencetak hasil sesuai dengan kondisi yang terpenuhi dalam setiap pernyataan if.

Output jalannya program:

```
Masukkan angka: 15
Angka yang dimasukkan positif
Angka yang dimasukkan adalah bilangan ganjil
Angka yang dimasukkan lebih dari 10
```

Anda dapat mencoba menjalankan program tersebut dengan memasukkan angka yang berbeda untuk melihat hasilnya sesuai dengan kondisi yang terpenuhi dalam pernyataan if.

## 5. Perbedaan if dan switch

Perbedaan antara pernyataan if-else dan pernyataan switch terletak pada cara mereka mengevaluasi kondisi dan mengambil keputusan. Berikut adalah perbedaan utama antara keduanya:

1. Pengevaluasian Kondisi:
  - a. if-else: Pernyataan if-else mengevaluasi kondisi yang bersifat boolean (true atau false). Kondisi bisa melibatkan perbandingan, operasi logika, atau hasil dari fungsi boolean.
  - b. switch: Pernyataan switch mengevaluasi ekspresi tunggal dan mencocokkannya dengan nilai-nilai yang terkait dalam setiap kasus (case).
2. Jumlah Kasus:
  - a. if-else: Pernyataan if-else lebih fleksibel dalam menangani kondisi yang kompleks dan dapat memiliki jumlah kasus yang tidak terbatas.
  - b. switch: Pernyataan switch ideal digunakan ketika ada banyak kasus yang terkait dengan nilai spesifik. Kasus-kasus dalam switch harus diskrit dan terbatas.
3. Perbandingan Nilai:
  - a. if-else: Pernyataan if-else mendukung perbandingan nilai dengan operator perbandingan seperti ==, !=, <, >, <=, dan >=.
  - b. switch: Pernyataan switch hanya mendukung perbandingan nilai kesamaan (==) dengan kasus-kasus yang sudah ditentukan.

Kapan sebaiknya kita menggunakan pernyataan if-else dan kapan kita menggunakan pernyataan switch?

1. Gunakan if-else ketika:
  - a. Terdapat kondisi yang kompleks dan memerlukan perbandingan nilai yang rumit.
  - b. Ada banyak kondisi yang mungkin dan kasus-kasus yang tidak terbatas.
  - c. Diperlukan tindakan berbeda untuk setiap kondisi yang ada.

## 2. Gunakan switch ketika:

- a. Terdapat sejumlah besar kasus yang diskrit dan terbatas.
- b. Diperlukan pencocokan nilai spesifik dengan tindakan yang sesuai.
- c. Kode perlu lebih terstruktur dan terlihat lebih rapi.

Pilihan antara if-else dan switch tergantung pada kebutuhan dan kompleksitas kondisi yang ingin dievaluasi. Pastikan untuk memilih pernyataan yang paling sesuai dengan situasi yang dihadapi dan memberikan kejelasan serta pemeliharaan kode yang baik.

## 6. Operator ?

Operator `?`, juga dikenal sebagai operator ternary, adalah operator kondisional dalam bahasa pemrograman Java. Operator ini memberikan cara singkat untuk menulis ekspresi kondisional sederhana. Sintaks dari operator `?` adalah sebagai berikut:

```
kondisi ? ekspresi1 : ekspresi2  
Berikut adalah cara kerjanya:
```

- 1) Kondisi dievaluasi menjadi true atau false.
- 2) Jika kondisi adalah true, maka ekspresi1 dievaluasi dan menjadi hasil dari seluruh ekspresi.
- 3) Jika kondisi adalah false, maka ekspresi2 dievaluasi dan menjadi hasil dari seluruh ekspresi.
- 4) Operator `?` : dapat digunakan sebagai alternatif untuk pernyataan if-else ketika Anda memiliki ekspresi kondisional sederhana dengan dua kemungkinan hasil.

Berikut adalah contoh untuk mendemonstrasikan penggunaan operator `?`:

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

Pada contoh ini, kondisi `angka % 2 == 0` memeriksa apakah angka adalah bilangan genap. Jika kondisi tersebut true, nilai "Genap" akan diberikan ke variabel hasil. Jika kondisi tersebut false, nilai "Ganjil" akan diberikan ke variabel hasil. Nilai dari hasil kemudian dicetak, yang dalam contoh ini akan menjadi "Genap" karena angka dapat dibagi oleh 2.

Operator `?` : dapat berguna dalam menyederhanakan ekspresi kondisional dan membuat kode lebih ringkas. Namun, penting untuk menggunakan operator ini dengan bijaksana dan memastikan bahwa kode tetap mudah dibaca dan dipahami.

## 7. Kapan Sebaiknya kita Menggunakan operator ternary

Operator ternary `?` : dapat digunakan sebagai alternatif untuk pernyataan if-else dalam situasi-situasi di mana kita memiliki kondisi yang sederhana dan ingin mengevaluasi sebuah ekspresi berdasarkan kondisi tersebut. Penggunaan operator ternary dapat membuat kode lebih ringkas dan mudah dibaca.

Perbedaan utama antara operator ternary dan pernyataan if-else adalah sintaksnya yang lebih singkat. Operator ternary terdiri dari tiga bagian: kondisi, ekspresi yang dievaluasi jika kondisi benar, dan ekspresi yang dievaluasi jika kondisi salah.

Kondisi yang sederhana: Operator ternary cocok digunakan ketika kita memiliki kondisi yang sederhana dan ingin menghasilkan nilai berdasarkan kondisi tersebut. Jika kondisi hanya melibatkan satu perbandingan atau operator logika yang sederhana, operator ternary bisa menjadi pilihan yang baik.

Kode yang singkat dan mudah dibaca: Operator ternary dapat membuat kode lebih ringkas dan mudah dibaca, terutama dalam situasi di mana ada keputusan sederhana yang harus dibuat. Ini dapat membantu mengurangi kelebihan penulisan kode yang tidak perlu.

Penggunaan yang tidak berlebihan: Meskipun operator ternary dapat berguna dalam beberapa situasi, kita perlu berhati-hati agar tidak menggunakan operator ternary secara berlebihan. Jika kondisi atau ekspresi yang dievaluasi rumit, sulit untuk dibaca, atau melibatkan banyak perbandingan atau logika, lebih baik menggunakan pernyataan if-else yang lebih ekspresif.

contoh penggunaan operator ternary:

```
int x = 10;
String hasil = (x > 5) ? "Lebih besar dari 5" : "Kurang dari atau sama dengan 5";
System.out.println(hasil);
```

Pada contoh di atas, kita menggunakan operator ternary untuk mengevaluasi kondisi  $x > 5$ . Jika kondisi tersebut benar, nilai "Lebih besar dari 5" akan diberikan ke variabel hasil. Jika kondisi tersebut salah, nilai "Kurang dari atau sama dengan 5" akan diberikan ke variabel hasil. Hasilnya akan dicetak sesuai dengan nilai kondisi.

Penting untuk menggunakan operator ternary dengan bijaksana dan mempertimbangkan faktor-faktor seperti kejelasan, kesederhanaan, dan kebutuhan konteks dalam pengambilan keputusan apakah menggunakan operator ternary atau pernyataan if-else.

## Contoh Program

### Contoh 1 :

Berikut adalah contoh program Java yang menggunakan operator  $?$  : dalam sebuah program lengkap:

```
1. import java.util.Scanner;
2.
3. public class OperatorTernary {
4.     public static void main(String[] args) {
5.         Scanner input = new Scanner(System.in);
6.
7.         System.out.print("Masukkan angka: ");
8.         int angka = input.nextInt();
9.
10.        String hasil = (angka % 2 == 0) ? "Genap" : "Ganjil";
11.        System.out.println("Angka yang dimasukkan adalah " + hasil);
12.    }
13. }
14.
```

Pada program di atas, pengguna diminta untuk memasukkan sebuah angka. Kemudian, menggunakan operator  $?$  :, program menentukan apakah angka tersebut genap atau ganjil. Hasilnya disimpan dalam variabel hasil yang kemudian dicetak.

Output jalannya program:

```
Masukkan angka: 7
Angka yang dimasukkan adalah Ganjil
```

Program ini memanfaatkan operator `?` : untuk mempersingkat kondisi angka `% 2 == 0` dan memberikan nilai "Genap" jika kondisi tersebut true, atau "Ganjil" jika kondisi tersebut false. Anda dapat mencoba menjalankan program ini dengan memasukkan angka-angka berbeda dan melihat hasilnya sesuai dengan kondisi genap atau ganjil dari angka yang dimasukkan.

## Contoh 2:

Berikut adalah contoh program Java yang menggabungkan berbagai pernyataan seleksi (selection statements) dan operator dalam sebuah program lengkap:

```
1. import java.util.Scanner;
2.
3. public class GabunganSelectionOperator {
4.     public static void main(String[] args) {
5.         Scanner input = new Scanner(System.in);
6.
7.         // Selection statement (if-else)
8.         System.out.print("Masukkan angka: ");
9.         int angka = input.nextInt();
10.
11.        if (angka > 0) {
12.            System.out.println("Angka yang dimasukkan adalah bilangan positif");
13.        } else if (angka < 0) {
14.            System.out.println("Angka yang dimasukkan adalah bilangan negatif");
15.        } else {
16.            System.out.println("Angka yang dimasukkan adalah nol");
17.        }
18.
19.        // Operator aritmatika
20.        int x = 5;
21.        int y = 3;
22.        int hasilPenjumlahan = x + y;
23.        int hasilPengurangan = x - y;
24.        int hasilPerkalian = x * y;
25.        double hasilPembagian = (double) x / y;
26.
27.        System.out.println("Hasil Penjumlahan: " + hasilPenjumlahan);
28.        System.out.println("Hasil Pengurangan: " + hasilPengurangan);
29.        System.out.println("Hasil Perkalian: " + hasilPerkalian);
30.        System.out.println("Hasil Pembagian: " + hasilPembagian);
31.
32.        // Selection statement (switch)
33.        System.out.print("Masukkan pilihan (A, B, atau C): ");
34.        char pilihan = input.next().charAt(0);
35.
36.        switch (pilihan) {
37.            case 'A':
38.                System.out.println("Anda memilih pilihan A");
39.                break;
```



```

40.         case 'B':
41.             System.out.println("Anda memilih pilihan B");
42.             break;
43.         case 'C':
44.             System.out.println("Anda memilih pilihan C");
45.             break;
46.         default:
47.             System.out.println("Pilihan yang dimasukkan tidak valid");
48.             break;
49.     }
50. }
51. }
52.

```

Pada program di atas, terdapat beberapa pernyataan seleksi seperti if-else dan switch, serta penggunaan operator aritmatika seperti penjumlahan, pengurangan, perkalian, dan pembagian. Program ini mengambil masukan dari pengguna dan melakukan seleksi serta operasi yang sesuai dengan masukan tersebut. Anda dapat mencoba menjalankan program ini dan memberikan input yang berbeda untuk melihat hasilnya sesuai dengan kondisi seleksi dan operasi yang dilakukan. Pastikan untuk memahami cara kerja pernyataan seleksi dan operator dalam program ini.

### Contoh 3:

Berikut adalah contoh program Java yang menggabungkan penggunaan pernyataan seleksi (selection) dan operator dalam satu program:

```

1. import java.util.Scanner;
2.
3. public class GabunganSelectionOperator {
4.     public static void main(String[] args) {
5.         Scanner input = new Scanner(System.in);
6.
7.         System.out.print("Masukkan usia: ");
8.         int usia = input.nextInt();
9.
10.        System.out.print("Apakah Anda memiliki SIM? (ya/tidak): ");
11.        String sim = input.next();
12.
13.        if (usia >= 18 && sim.equalsIgnoreCase("ya")) {
14.            System.out.println("Anda dapat mengemudikan mobil.");
15.        } else if (usia >= 16 && sim.equalsIgnoreCase("ya")) {
16.            System.out.println("Anda dapat mengemudikan sepeda motor.");
17.        } else {
18.            System.out.println("Anda tidak dapat mengemudikan kendaraan.");
19.        }
20.    }
21. }
22.

```

Pada contoh program di atas, pengguna diminta untuk memasukkan usia dan keberadaan SIM (Surat Izin Mengemudi). Program menggunakan pernyataan seleksi if-else if-else untuk memeriksa kondisi dan memberikan hasil berdasarkan nilai usia dan SIM yang dimasukkan.

- Jika usia 18 tahun atau lebih dan memiliki SIM, maka program akan mencetak "Anda dapat mengemudikan mobil."
- Jika usia 16 tahun atau lebih dan memiliki SIM, maka program akan mencetak "Anda dapat mengemudikan sepeda motor."
- Jika tidak memenuhi kedua kondisi di atas, maka program akan mencetak "Anda tidak dapat mengemudikan kendaraan."

Output jalannya program:

```
Masukkan usia: 20
Apakah Anda memiliki SIM? (ya/tidak): ya
Anda dapat mengemudikan mobil.
```

Anda dapat mencoba menjalankan program ini dengan memasukkan usia dan keberadaan SIM yang berbeda untuk melihat hasil sesuai dengan kondisi yang terpenuhi. Program ini menggabungkan penggunaan pernyataan seleksi (if-else if-else) dengan operator logika (&&) dan operator perbandingan (==, equalsIgnoreCase()) untuk mengambil keputusan berdasarkan kondisi yang diberikan.

#### Contoh 4:

Berikut adalah contoh program Java yang menggabungkan beberapa pernyataan seleksi (if-else, switch) dan operator dalam sebuah program lengkap:

```
1. import java.util.Scanner;
2.
3. public class ProgramGabungan {
4.     public static void main(String[] args) {
5.         Scanner input = new Scanner(System.in);
6.
7.         System.out.print("Masukkan nilai: ");
8.         int nilai = input.nextInt();
9.
10.        if (nilai >= 0 && nilai <= 100) {
11.            String keterangan;
12.
13.            if (nilai >= 90) {
14.                keterangan = "A";
15.            } else if (nilai >= 80) {
16.                keterangan = "B";
17.            } else if (nilai >= 70) {
18.                keterangan = "C";
19.            } else if (nilai >= 60) {
20.                keterangan = "D";
21.            } else {
22.                keterangan = "E";
23.            }
24.
25.            System.out.println("Nilai: " + nilai);
26.            System.out.println("Keterangan: " + keterangan);
27.
28.            switch (keterangan) {
```

```

29.         case "A":
30.             System.out.println("Pertahankan prestasimu!");
31.             break;
32.         case "B":
33.             System.out.println("Bisa lebih baik lagi!");
34.             break;
35.         case "C":
36.             System.out.println("Perlu lebih banyak belajar.");
37.             break;
38.         case "D":
39.             System.out.println("Perlu meningkatkan usaha.");
40.             break;
41.         case "E":
42.             System.out.println("Belajar lebih giat lagi!");
43.             break;
44.         default:
45.             System.out.println("Keterangan tidak valid.");
46.             break;
47.     }
48. } else {
49.     System.out.println("Nilai tidak valid.");
50. }
51. }
52. }
53.

```

Pada program di atas, pengguna diminta untuk memasukkan sebuah nilai. Program akan memeriksa apakah nilai tersebut valid (0 hingga 100). Jika nilai valid, program akan menentukan keterangan berdasarkan rentang nilai menggunakan pernyataan if-else. Kemudian, program menggunakan pernyataan switch untuk memberikan pesan tambahan berdasarkan keterangan yang diperoleh.

Output jalannya program:

```

Masukkan nilai: 85
Nilai: 85
Keterangan: B
Bisa lebih baik lagi!

```

Program ini menggabungkan beberapa pernyataan seleksi (if-else, switch) untuk mengevaluasi nilai dan memberikan keterangan serta pesan tambahan berdasarkan rentang nilai. Anda dapat mencoba menjalankan program ini dengan memasukkan nilai yang berbeda untuk melihat hasilnya sesuai dengan rentang nilai dan pesan keterangan yang diberikan.

### Contoh 5:

Berikut adalah contoh program Java yang menggabungkan penggunaan operator dan konstruksi pemilihan (selection) dalam satu program:

```

1. import java.util.Scanner;

```

```

2.
3. public class GabunganOperatorSelection {
4.     public static void main(String[] args) {
5.         Scanner input = new Scanner(System.in);
6.
7.         System.out.print("Masukkan nilai ujian: ");
8.         int nilai = input.nextInt();
9.
10.        // Penggunaan operator dan konstruksi pemilihan
11.        String hasil;
12.        if (nilai >= 80) {
13.            hasil = "A";
14.        } else if (nilai >= 70) {
15.            hasil = "B";
16.        } else if (nilai >= 60) {
17.            hasil = "C";
18.        } else if (nilai >= 50) {
19.            hasil = "D";
20.        } else {
21.            hasil = "E";
22.        }
23.
24.        // Cetak hasil
25.        System.out.println("Hasil ujian: " + hasil);
26.    }
27. }
28.

```

Pada program di atas, pengguna diminta untuk memasukkan nilai ujian. Program menggunakan operator perbandingan ( $\geq$ ) dan konstruksi pemilihan if-else if-else untuk menentukan nilai huruf yang sesuai dengan rentang nilai yang diberikan.

Output jalannya program:

```

Masukkan nilai ujian: 75
Hasil ujian: B

```

Program ini mengevaluasi kondisi secara berurutan dan menetapkan nilai huruf yang sesuai dengan rentang nilai yang diberikan. Hasil akhir kemudian dicetak. Anda dapat mencoba menjalankan program ini dengan memasukkan nilai ujian yang berbeda dan melihat hasil nilai huruf yang sesuai dengan rentang nilai yang ditentukan dalam program.

#### Contoh 6:

Berikut adalah contoh program Java yang menggabungkan pernyataan seleksi (selection statements) dan operator `?:` dalam sebuah program lengkap:

```

1. import java.util.Scanner;
2.
3. public class GabunganSelectionOperator {
4.     public static void main(String[] args) {
5.         Scanner input = new Scanner(System.in);
6.

```

```

7.         System.out.print("Masukkan angka: ");
8.         int angka = input.nextInt();
9.
10.        // Selection statement (if-else) dengan operator ?
11.        String hasil = (angka > 0) ? "Bilangan positif" : (angka < 0) ? "Bilangan negatif" :
"No1";
12.        System.out.println("Angka yang dimasukkan adalah " + hasil);
13.
14.        // Operator aritmatika dengan operator ?
15.        int x = 5;
16.        int y = 3;
17.        int hasilPenjumlahan = x + y;
18.        int hasilPengurangan = x - y;
19.        int hasilPerkalian = x * y;
20.        double hasilPembagian = (y != 0) ? ((double) x / y) : 0;
21.
22.        System.out.println("Hasil Penjumlahan: " + hasilPenjumlahan);
23.        System.out.println("Hasil Pengurangan: " + hasilPengurangan);
24.        System.out.println("Hasil Perkalian: " + hasilPerkalian);
25.        System.out.println("Hasil Pembagian: " + hasilPembagian);
26.
27.        // Selection statement (switch) dengan operator ?
28.        System.out.print("Masukkan pilihan (A, B, atau C): ");
29.        char pilihan = input.next().charAt(0);
30.
31.        String teks = (pilihan == 'A') ? "Anda memilih pilihan A" :
32.                        (pilihan == 'B') ? "Anda memilih pilihan B" :
33.                        (pilihan == 'C') ? "Anda memilih pilihan C" :
34.                        "Pilihan yang dimasukkan tidak valid";
35.        System.out.println(teks);
36.    }
37. }
38.

```

Pada program di atas, terdapat penggunaan operator ? : bersama pernyataan seleksi seperti if-else dan switch. Operator ? : digunakan untuk menggantikan blok pernyataan seleksi dengan ekspresi yang lebih ringkas.

Anda dapat mencoba menjalankan program ini dan memberikan input yang berbeda untuk melihat hasilnya sesuai dengan kondisi seleksi dan operasi yang dilakukan. Pastikan untuk memahami cara kerja pernyataan seleksi dan operator ? : dalam program ini.

#### Contoh 7:

Berikut adalah contoh program Java yang menggabungkan pernyataan seleksi (if-else dan switch) dengan operator ternary (? :) dalam sebuah program lengkap:

```

1. import java.util.Scanner;
2.
3. public class GabunganSelectionOperator {
4.     public static void main(String[] args) {
5.         Scanner input = new Scanner(System.in);
6.
7.         // Selection statement (if-else) dengan operator ternary

```

```

8.         System.out.print("Masukkan angka: ");
9.         int angka = input.nextInt();
10.
11.         String jenisAngka = (angka > 0) ? "positif" : (angka < 0) ? "negatif" : "nol";
12.         System.out.println("Angka yang dimasukkan adalah bilangan " + jenisAngka);
13.
14.         // Selection statement (switch) dengan operator ternary
15.         System.out.print("Masukkan pilihan (A, B, atau C): ");
16.         char pilihan = input.next().charAt(0);
17.
18.         String hasilPilihan = (pilihan == 'A') ? "Anda memilih pilihan A"
19.                                : (pilihan == 'B') ? "Anda memilih pilihan B"
20.                                : (pilihan == 'C') ? "Anda memilih pilihan C"
21.                                : "Pilihan yang dimasukkan tidak valid";
22.         System.out.println(hasilPilihan);
23.     }
24. }
25.

```

Pada program di atas, operator ternary (? :) digunakan dalam kombinasi dengan pernyataan seleksi (if-else dan switch) untuk menyederhanakan logika pemilihan dan memberikan nilai ke variabel hasil.

Output jalannya program:

```

Masukkan angka: 7
Angka yang dimasukkan adalah bilangan positif
Masukkan pilihan (A, B, atau C): B
Anda memilih pilihan B

```

Anda dapat mencoba menjalankan program ini dengan memberikan input yang berbeda untuk melihat hasilnya sesuai dengan kondisi seleksi yang ada. Perhatikan cara penggunaan operator ternary dalam program ini untuk menggantikan pernyataan if-else dan switch.

## Latihan

### Latihan 1:

Berikut adalah contoh program yang memiliki kesalahan dan perlu diperbaiki :

```

1. public class ProgramSalah {
2.     public static void main(String[] args) {
3.         int x = 5;
4.         int y = 10;
5.
6.         int z = x + y;
7.
8.         System.out.println("Hasil penjumlahan: " + z);
9.
10.        if (z > 10)
11.            System.out.println("Nilai z lebih besar dari 10");
12.        else

```

```

13.         System.out.println("Nilai z kurang dari atau sama dengan 10");
14.
15.     for (int i = 0, i < 5, i++) {
16.         System.out.println("Iterasi ke-" + i);
17.     }
18. }
19. }
20.

```

Program di atas memiliki beberapa kesalahan yang perlu diperbaiki. Tugas adalah memperbaiki kesalahan-kesalahan tersebut sehingga program dapat dikompilasi dan berjalan dengan semestinya.

Instruksi:

- 1) Identifikasi dan perbaiki kesalahan-kesalahan yang ada dalam program.
- 2) Perhatikan kesalahan pada sintaks, penempatan tanda kurung, dan penggunaan operator.
- 3) Tulislah kode program yang telah diperbaiki dalam file dengan ekstensi .java.
- 4) Jalankan program yang telah diperbaiki untuk memastikan tidak ada lagi kesalahan.
- 5) Lampirkan file program yang telah diperbaiki dalam pengumpulan tugas Anda.
- 6) Kirimkan tugas sesuai dengan petunjuk pengumpulan yang telah ditentukan.

## Latihan 2:

```

1. public class ProgramSalah {
2.     public static void main(String[] args) {
3.         int x = 5
4.         if (x > 0) {
5.             System.out.println("Angka positif");
6.         }
7.         else {
8.             System.out.println("Angka negatif");
9.         }
10.    }
11. }
12.

```

Program di atas mengandung beberapa kesalahan sintaksis yang harus diperbaiki agar program dapat dikompilasi dan dijalankan dengan benar. Tugas adalah memperbaiki kesalahan-kesalahan tersebut sehingga program dapat berfungsi sebagaimana mestinya.

Instruksi:

- 1) Identifikasi kesalahan-kesalahan sintaksis dalam program tersebut.
- 2) Perbaiki kesalahan-kesalahan tersebut dan tulis ulang program agar dapat dikompilasi dan dijalankan tanpa kesalahan.
- 3) Jalankan program setelah diperbaiki untuk memastikan tidak ada kesalahan.
- 4) Simpan program yang telah diperbaiki dalam file dengan ekstensi .java.
- 5) Sertakan file program yang telah diperbaiki dalam pengumpulan tugas Anda.

### Latihan 3 :

Berikut adalah contoh program yang salah dan perlu diperbaiki:

```
1. public class ProgramSalah {
2.     public static void main(String[] args) {
3.         int x = 5;
4.         int y = 10;
5.
6.         if (x < y)
7.             System.out.println("x lebih kecil dari y");
8.         else
9.             System.out.println("x lebih besar dari y");
10.
11.        int z = x + y;
12.
13.        for (int i = 0; i <= z; i++) {
14.            System.out.print(i + " ");
15.        }
16.    }
17. }
18.
```

Program di atas memiliki kesalahan yang mengakibatkan tidak dapat dikompilasi dengan benar. Tugas adalah memperbaiki kesalahan tersebut sehingga program dapat dijalankan dan menghasilkan output yang sesuai.

Instruksi:

- 1) Identifikasi kesalahan dalam program tersebut.
- 2) Lakukan perbaikan yang diperlukan untuk mengatasi kesalahan.
- 3) Jalankan program setelah melakukan perbaikan untuk memastikan program berjalan dengan benar.
- 4) Tulislah kode program yang telah diperbaiki dalam file dengan ekstensi .java.
- 5) Lampirkan file program yang telah diperbaiki dalam pengumpulan tugas Anda.
- 6) Tugas siswa adalah untuk menganalisis dan memperbaiki kesalahan dalam program tersebut agar dapat dijalankan dengan benar.

### Tugas Rumah

1. Buatlah sebuah program Java yang mengambil input bilangan bulat dari pengguna dan menentukan apakah bilangan tersebut adalah bilangan prima. Jika bilangan tersebut prima, program harus mencetak "Bilangan tersebut adalah bilangan prima", jika tidak, program harus mencetak "Bilangan tersebut bukan bilangan prima".
2. Buatlah sebuah program Java yang meminta pengguna memasukkan angka bulan (1-12) dan mencetak nama bulan yang sesuai. Jika pengguna memasukkan angka di luar rentang 1-12, program harus mencetak "Angka bulan tidak valid".
3. Buatlah sebuah program Java yang meminta pengguna memasukkan suhu dalam derajat Celsius dan mengonversinya ke Fahrenheit. Program harus mencetak hasil konversi dengan presisi dua desimal.



4. Buatlah sebuah program Java yang meminta pengguna memasukkan sebuah kata atau kalimat. Program harus menghitung dan mencetak jumlah huruf vokal (a, e, i, o, u) dalam kata atau kalimat tersebut.
5. Buatlah sebuah program Java yang meminta pengguna memasukkan sebuah angka bulat positif. Program harus mencetak deret angka dari 1 hingga angka yang dimasukkan, dengan setiap angka diikuti tanda bintang "\*". Contohnya, jika pengguna memasukkan angka 5, program harus mencetak:

```
1 *
2 **
3 ***
4 ****
5 *****
```

Instruksi:

- 1) Tulislah kode program untuk setiap soal praktikum dalam file dengan ekstensi .java.
- 2) Implementasikan logika program dengan benar sesuai dengan deskripsi soal.
- 3) Jalankan program untuk memastikan tidak ada kesalahan.
- 4) Lampirkan file-file program dalam pengumpulan tugas Anda.
- 5) Kirimkan tugas sesuai dengan petunjuk pengumpulan yang telah ditentukan.