

**KODE PROGRAM**

Tugas 4 PBO

Makassar, 12 Maret 2024

**TUGAS PEMROGRAMAN  
BEROIENTASI OBJEK**



Nama : Dzulfadly  
Stambuk : 13020220084  
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MUSLIM INDONESIA  
MAKASSAR  
2024**

## KODE PROGRAM

1. Apakah perbedaan antara struktur kontrol percabangan if-else dan switch-case?  
Dalam Java, perbedaan antara struktur kontrol percabangan `if-else` dan `switch-case` adalah:

### Penggunaan

- `if-else`: Digunakan ketika terdapat kondisi yang kompleks atau beragam yang harus dievaluasi.
- `switch-case`: Digunakan ketika terdapat satu variabel yang nilai-nilainya akan dibandingkan dengan beberapa nilai konstan.

### Kondisi

- `if-else`: Kondisi yang dievaluasi dapat berupa ekspresi boolean atau hasil dari perbandingan nilai.
- `switch-case`: Kondisi harus berupa nilai yang dapat diperbandingkan, seperti integer, karakter, atau enum.

### Kelebihan

- `if-else`: Lebih fleksibel dalam menangani kondisi yang kompleks dan tidak terbatas pada tipe data tertentu.
- `switch-case`: Lebih cocok digunakan saat membandingkan satu variabel dengan beberapa nilai konstan karena lebih mudah dibaca dan lebih ringkas.

### Break Statement

- `if-else`: Tidak memerlukan break statement karena setiap kondisi dievaluasi secara independen.
- `switch-case`: Biasanya memerlukan break statement setelah setiap kasus (case) untuk mencegah eksekusi kode pada kasus berikutnya.

2. Kapan digunakan struktur kontrol if-else dan switch-case

Struktur kontrol `if-else` dan `switch-case` digunakan dalam Java tergantung pada kebutuhan dan situasi tertentu:

## KODE PROGRAM

### Penggunaan if-else

- Ketika terdapat kondisi yang kompleks dan bervariasi yang harus dievaluasi.
- Ketika setiap kondisi memerlukan perlakuan atau tindakan yang berbeda.
- Saat ingin mengevaluasi ekspresi boolean.
- Untuk menangani kondisi yang tidak bisa diwakili dengan cara yang sederhana atau linear.

### Penggunaan switch-case

- Ketika hanya ada satu variabel yang nilai-nilainya akan dibandingkan dengan beberapa nilai konstan.
- Saat ingin menghindari penulisan kode yang berulang-ulang untuk evaluasi yang sama.
- Saat ingin kode terlihat lebih terstruktur dan mudah dibaca, terutama jika ada banyak pilihan.

3. Pada program 2, tambahkan perintah untuk memilih 2 opsi menggunakan kontrol switch..case.

opsi pilihah 1=inputNilai()

Pilihan 2=inputNilaiBaru()

```
Masukkan Jumlah Data : 2
Pilihan:
1. Input Nilai
2. Input Nilai Baru
Masukkan Pilihan : █
```

```
import java.util.Scanner;
```

```
public class TestNilai {
```

```
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("Masukkan Jumlah Data : ");
```

```
        int banyakData = input.nextInt();
```

```
        int nilai[] = new int[banyakData];
```

```
        // Menampilkan menu pilihan
```

## KODE PROGRAM

```
System.out.println("Pilihan: ");
System.out.println("1. Input Nilai");
System.out.println("2. Input Nilai Baru");
System.out.print("Masukkan Pilihan : ");
int pilihan = input.nextInt();

HitungRata hitung = new HitungRata(input);

// Memilih opsi dengan switch-case
switch (pilihan) {
    case 1:
        hitung.inputNilai(nilai);
        System.out.println("Daftar Nilai : ");
        hitung.cetakNilai(nilai);
        System.out.println("Rata Nilai : " + hitung.rataNilai(banyakData));
        break;
    case 2:
        hitung.inputNilaiBaru(banyakData);
        System.out.println("Daftar Nilai Baru : ");
        hitung.cetakNilaiBaru();
        break;
    default:
        System.out.println("Pilihan tidak valid!");
}
}
}

class HitungRata {

    private Scanner input;
```

## KODE PROGRAM

```
private int[] nilai;

public HitungRata(Scanner input) {
    this.input = input;
}

public void inputNilai(int[] nilai) {
    this.nilai = nilai;
    for (int i = 0; i < nilai.length; i++) {
        System.out.print("Masukkan Nilai : ");
        nilai[i] = input.nextInt();
    }
}

public void cetakNilai(int[] nilai) {
    for (int i = 0; i < nilai.length; i++) {
        System.out.print(nilai[i] + " ");
    }
    System.out.println();
}

public double rataNilai(int banyakData) {
    double total = 0;
    for (int i = 0; i < nilai.length; i++) {
        total += nilai[i];
    }
    return total / banyakData;
}

public void inputNilaiBaru(int banyakData) {
```

## KODE PROGRAM

```
for (int i = 0; i < nilai.length; i++) {  
    System.out.print("Masukkan Nilai Baru : ");  
    nilai[i] = input.nextInt();  
}  
}  
  
public void cetakNilaiBaru() {  
    for (int i = 0; i < nilai.length; i++) {  
        System.out.print(nilai[i] + " ");  
    }  
    System.out.println();  
}  
}
```

### 4. Apakah perbedaan antara struktur kontrol perulangan while dan do-while?

Di Java, keduanya while dan do-while digunakan untuk mengeksekusi blok kode berulang kali selama kondisi tertentu benar. Namun, perbedaan utama di antara keduanya terletak pada cara mereka menangani evaluasi awal terhadap kondisi tersebut.

Sebuah while loop mengevaluasi kondisi sebelum mengeksekusi blok kode di dalam loop. Jika kondisinya benar, maka kode di dalam loop akan dieksekusi. Jika kondisinya salah, maka perulangan akan dilewati seluruhnya.

Di sisi lain, sebuah do-while loop mengevaluasi kondisi setelah mengeksekusi blok kode di dalam loop. Artinya, kode di dalam perulangan dijamin akan dieksekusi setidaknya satu kali, terlepas dari apakah kondisinya benar atau salah.

### 5. Kapan digunakan struktur kontrol for?

Struktur kontrol for digunakan untuk melakukan perulangan yang telah ditentukan sebelumnya.

### 6. Apakah perbedaan antara Array dan ArrayList? berilah contoh masing-masing

## KODE PROGRAM

### Array :

- Struktur data statis yang menyimpan elemen dengan tipe data yang sama.
- Ukuran array harus ditentukan terlebih dahulu dan tidak dapat diubah setelahnya.
- Akses elemen array lebih cepat dibandingkan ArrayList.
- Membutuhkan lebih banyak memori dibandingkan ArrayList untuk menyimpan elemen dengan tipe data yang sama.

### Contoh array :

```
public class Array {  
    public static void main(String[] args) {  
        int[] angka = {1, 2, 3, 4, 5};  
        for (int i = 0; i < angka.length; i++) {  
            System.out.println(angka[i]);  
        }  
    }  
}
```

### ArrayList :

- Struktur data dinamis yang menyimpan elemen dengan tipe data yang sama.
- Ukuran ArrayList dapat diubah secara dinamis saat program berjalan.
- Akses elemen ArrayList lebih lambat dibandingkan array.
- Membutuhkan memori lebih sedikit dibandingkan array untuk menyimpan elemen dengan tipe data yang sama.

### Contoh ArrayList

```
import java.util.ArrayList;  
  
public class ArrayList {  
    public static void main(String[] args) {  
        ArrayList<String> nama = new ArrayList<>();  
        nama.add("Andi");  
    }  
}
```

## KODE PROGRAM

```
nama.add("Budi");  
nama.add("Cindy");  
for (String namaOrang : nama) {  
    System.out.println(namaOrang);  
}  
}  
}
```

7. Buatlah contoh program yang mengimplementasikan HashMap dengan memasukkan nilai dan key melalui keyboard!

```
Masukkan pasangan kunci-nilai ke dalam HashMap (ketik 'selesai' untuk mengakhiri):  
Masukkan kunci: Dzul  
Masukkan nilai: 10  
Masukkan kunci: fadly  
Masukkan nilai: 90  
Masukkan kunci: selesai  
  
Isi HashMap:  
Kunci: Dzul , Nilai: 10  
Kunci: fadly, Nilai: 90  
  
[Program finished]
```



## KODE PROGRAM

### Tugas Evaluasi 4

1. Berdasarkan ke tiga program di Modul Class utama, Class Orang dan Class Mahasiswa, manakah yang menunjukkan konsep pewarisan dan polimorfisme! Jelaskan konsep tersebut

sesuai program tersebut!

- Yang dimaksud pewarisan dari program tersebut adalah class Mahasiswa, di mana kelas Mahasiswa mewarisi properti dan metode dari class Orang, karena memiliki kata kunci Extends pada deklarasi Mahasiswa.
- Terlihat pada method info().Method info() di kelas Mahasiswa menggantikan method info() di kelas Orang ketika objek Mahasiswa dipanggil.

2. Tambahkan static pada method info() Class Orang dan Class Mahasiswa kemudian lakukan pemanggilan method info() pada program utama (Class utama (Class utama)

```
public class Mahasiswa extends Orang {  
    2 usages  
    private String stb;  
    no usages  
    public Mahasiswa() {  
        super();  
        this.stb = "13020220084";  
    }  
    no usages  
    public Mahasiswa(String stb, String nama) {  
        this.nama = nama;  
        this.stb = stb;  
    }  
    1 usage  
    public static void info() {  
        System.out.println("Class Mahasiswa");  
    }  
}
```

```
public class Orang {  
    3 usages  
    public String nama;  
    2 usages  
    public Orang() {  
        this.nama = "Aminah";  
    }  
    no usages  
    public Orang(String nama) {  
        this.nama = nama;  
    }  
    1 usage  
    public static void info() {  
        System.out.println("Class Orang");  
    }  
}
```

```
public class Utama {  
    no usages  
    public static void main(String[] args) {  
        Orang.info(); // Memanggil method info() dari Class Orang  
        Mahasiswa.info(); // Memanggil method info() dari Class Mah  
    }  
}
```

## KODE PROGRAM

3. Buatlah sebuah project dengan nama project stambuk anda dan buatlah pengorganisasian package Dan class seperti berikut.

