

## LAPORAN TUGAS 4

Nama : Nur Bakti Widjaya

Kelas : A4

NIM : 13020220134

### EVALUASI PRAKTIKUM

1. Apakah perbedaan antara struktur kontrol percabangan if-else dan switch-case?

:

**1. Penggunaan Kondisi:**

- **if-else:** Cocok untuk kondisi kompleks.
- **switch-case:** Hanya cocok untuk nilai-nilai yang telah ditentukan sebelumnya.

**2. Jenis Kondisi:**

- **if-else:** Menggunakan ekspresi boolean atau kondisi perbandingan.
- **switch-case:** Hanya menerima nilai yang bisa dibandingkan.

**3. Ketepatan Nilai:**

- **if-else:** Fleksibel untuk kondisi tidak ketat.
- **switch-case:** Memerlukan nilai yang sama persis dengan nilai yang ditentukan.

**4. Struktur:**

- **if-else:** Berurutan atau bersarang.
- **switch-case:** Berdasarkan ekspresi yang dibandingkan dengan nilai-nilai tertentu.

**5. Eksekusi:**

- **if-else:** Kondisi dievaluasi berurutan, blok pertama yang sesuai dieksekusi.
- **switch-case:** Ekspresi dibandingkan dengan nilai-nilai, eksekusi dimulai dari nilai yang cocok dan berlanjut sampai ditemukan pernyataan **break**.

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

:

**Penggunaan struktur kontrol if-else:**

1. Kondisi kompleks.
2. Kondisi tidak eksklusif.

3. Kondisi tidak memiliki nilai tetap.

**Penggunaan struktur kontrol switch-case:**

1. Banyak nilai tetap.
2. Banyak pilihan eksekusi.
3. Kode sederhana.

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  
Menu:  
1. Input Nilai  
2. Input Nilai Baru  
Pilih menu: 1  
Masukkan Nilai :  
1  
2  
  
Daftar Nilai :  
1  
2  
  
Rata Nilai : 1.5  
BUILD SUCCESSFUL (total time: 8 seconds)  
  
RUN:  
Masukkan Jumlah Data : 3  
Menu:  
1. Input Nilai  
2. Input Nilai Baru  
Pilih menu: 2  
Masukkan Nilai Baru :  
8  
7  
6  
Daftar Nilai Baru :  
8  
7  
6  
BUILD SUCCESSFUL (total time: 11 seconds)
```

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

:

**1. Evaluasi kondisi awal:**

- **while:** Kondisi dievaluasi sebelum menjalankan blok pernyataan. Jika kondisi awalnya salah, blok pernyataan tidak akan pernah dijalankan.
- **do-while:** Blok pernyataan dieksekusi sekali sebelum kondisi dievaluasi. Bahkan jika kondisi awalnya salah, blok pernyataan akan dijalankan sekali.

**2. Urutan penulisan kondisi:**

- **while:** Kondisi ditulis di awal loop sebelum blok pernyataan.
- **do-while:** Kondisi ditulis di akhir loop setelah blok pernyataan.

**3. Eksekusi blok pernyataan:**

- **while:** Blok pernyataan hanya dieksekusi jika kondisi awalnya benar.
- **do-while:** Blok pernyataan selalu dieksekusi setidaknya sekali sebelum pengecekan kondisi.

5. Kapan digunakan struktur kontrol for?

:

1. **Iterasi dengan jumlah yang diketahui:** Ketika Anda tahu berapa kali Anda ingin menjalankan blok pernyataan.
2. **Kondisi awal, kondisi perulangan, dan langkah iterasi yang terperinci:** Memungkinkan Anda menentukan semua ini dalam satu baris pernyataan.
3. **Iterasi melalui rentang nilai:** Berguna untuk iterasi numerik dari nilai awal ke nilai akhir dengan langkah iterasi yang jelas.
4. **Perulangan bersarang:** Dapat digunakan untuk membuat perulangan dalam perulangan, yang sering digunakan dalam logika pemrograman yang kompleks.

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

:

**Array:**

- Array adalah struktur data yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama.
- Jumlah elemen dalam array ditentukan saat deklarasi dan tidak dapat diubah setelahnya.
- Array menggunakan sintaksis yang sederhana dan merupakan bagian dari bahasa pemrograman itu sendiri.
- Array memiliki ukuran yang tetap dan alokasi memori statis.
- Contoh:

```
int[] angka = new int[5]; // Deklarasi array integer dengan panjang 5
```

```
String[] nama = {"tio", "cakra", "pian"}; // Deklarasi array string dengan nilai awal
```

**ArrayList:**

- ArrayList adalah kelas dalam Java yang menyediakan implementasi dari struktur data daftar (list) yang dapat diperluas.
- ArrayList dapat menyimpan elemen dengan tipe data yang berbeda dan dinamis menyesuaikan ukurannya saat elemen ditambah atau dihapus.

- ArrayList memungkinkan penambahan atau penghapusan elemen tanpa harus menentukan ukuran awalnya.
- ArrayList menggunakan metode dan properti yang disediakan oleh kelas ArrayList di dalam bahasa Java.
- ArrayList menggunakan alokasi memori dinamis.
- Contoh:

```
import java.util.ArrayList;
```

```
ArrayList<Integer> angka = new ArrayList<Integer>(); // Deklarasi ArrayList untuk menyimpan integer
```

```
ArrayList<String> nama = new ArrayList<String>(); // Deklarasi ArrayList untuk menyimpan string
```

```
nama.add("anis"); // Menambahkan elemen ke ArrayList nama
```

```
nama.add("prabowo");
```

```
nama.add("ganjar");
```

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

:

```
run:
Masukkan jumlah pasangan nilai dan kunci: 2
Masukkan kunci: 1
Masukkan nilai: 2
Masukkan kunci: 3
Masukkan nilai: 4
Daftar pasangan nilai dan kunci:
Kunci: 1, Nilai: 2
Kunci: 3, Nilai: 4
BUILD SUCCESSFUL (total time: 11 seconds)
```

## EVALUASI PRAKTIKUM

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

:

1. Pewarisan (Inheritance): Pewarisan terjadi ketika sebuah kelas (Mahasiswa) mewarisi properti dan metode dari kelas lain (Orang). Dalam program ini, Mahasiswa adalah kelas anak dari kelas Orang, yang berarti kelas Mahasiswa akan mewarisi semua properti dan metode yang dideklarasikan dalam kelas Orang. Konsep pewarisan memungkinkan kita untuk memanfaatkan kembali kode yang sudah ada dan memperluas fungsionalitasnya. Dalam contoh ini, kelas Mahasiswa mewarisi properti nama dari kelas Orang.
2. Polimorfisme: Polimorfisme adalah konsep di mana suatu objek dapat memiliki banyak bentuk. Dalam konteks pewarisan, polimorfisme mengacu pada kemampuan suatu objek (Mahasiswa) untuk berperilaku secara berbeda tergantung pada konteks di mana ia digunakan. Dalam program ini, kita dapat melihat polimorfisme ketika konstruktor kelas Mahasiswa di-overload. Di satu sisi, kita memiliki konstruktor tanpa parameter yang memanggil `super()` untuk memanggil konstruktor kelas Orang, dan di sisi lain, kita memiliki konstruktor yang menerima parameter nama dan stb. Polimorfisme memungkinkan kita untuk menggunakan objek Mahasiswa dengan cara yang berbeda, tergantung pada kebutuhan program.

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

:

```
Ini adalah kelas Orang
Ini adalah kelas Mahasiswa
```

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

:



