

LAPORAN TUGAS 4

Nama : Sahrul Adiyanto

Kelas : A4

Nim : 13020220143

Matkul : pemrograman berorientasi objek

EVALUASI PRAKTIKUM [Modul 3]

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

:

1. **Ekspresi yang diuji:**

- **if-else:** Bisa menggunakan ekspresi boolean atau nilai yang dievaluasi menjadi boolean.
- **switch-case:** Hanya dapat membandingkan nilai yang dapat dibandingkan, seperti nilai numerik atau string.

2. **Jumlah Kondisi:**

- **if-else:** Tidak terbatas, dapat memiliki banyak kondisi.
- **switch-case:** Terbatas, harus ditentukan sebelumnya dan hanya satu kasus yang sesuai untuk setiap nilai.

3. **Fleksibilitas:**

- **if-else:** Lebih fleksibel karena dapat menangani kondisi yang rumit dengan ekspresi boolean yang kompleks.
- **switch-case:** Cenderung lebih terbatas karena hanya cocok untuk situasi dengan jumlah kasus yang terbatas.

4. **Penggunaan:**

- **if-else:** Umum digunakan untuk situasi dengan banyak kondisi atau kondisi yang rumit.
- **switch-case:** Lebih sering digunakan untuk sejumlah kecil kasus atau kondisi yang hanya memerlukan pengujian nilai tunggal.

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

:

- **if-else**
 1. Kondisi rumit: Ketika kondisi yang harus diperiksa cukup rumit atau memiliki banyak perbandingan yang berbeda.
 2. Kondisi boolean: Ketika kondisi yang diuji adalah ekspresi boolean atau dapat dievaluasi menjadi boolean.

3. Banyak kondisi: Ketika ada banyak kondisi yang perlu diperiksa dan mungkin ada beberapa jalur yang berbeda tergantung pada kondisi tersebut.

4. Nilai tidak terbatas: Ketika kondisi yang diuji dapat berupa nilai yang tidak terbatas, seperti rentang numerik atau string yang bervariasi.

- Switch-case

1. Pemilihan kasus: Ketika hanya perlu memilih di antara beberapa kasus yang mungkin berbeda.

2. Kasus konstan: Ketika kasus-kasus yang diuji memiliki nilai yang sudah diketahui sebelumnya dan stabil.

3. Kondisi tunggal: Ketika kondisi yang diuji adalah nilai tunggal yang harus dibandingkan dengan beberapa nilai konstan.

4. Kode yang bersih: Ketika struktur switch-case dapat membuat kode menjadi lebih bersih dan mudah dibaca daripada rangkaian pernyataan if-else.

3. Pada program 2, tambahkan perintah untuk memilih 2 opsi menggunakan kontrol switch..case. opsi pilihah1=inputNilai()

Pilihan 2=inputNilaiBaru()

:

```
run:
Masukkan Jumlah Data : 2
Menu:
1. Input Nilai
2. Input Nilai Baru
Pilih menu: 1
Masukkan Nilai :
2
3

Daftar Nilai :
2
3

Rata Nilai      : 2.5
BUILD SUCCESSFUL (total time: 7 seconds)
```

```
run:
Masukkan Jumlah Data : 3
Menu:
1. Input Nilai
2. Input Nilai Baru
Pilih menu: 2
Masukkan Nilai Baru :
3
1
3
Daftar Nilai Baru :
3
1
3
BUILD SUCCESSFUL (total time: 12 seconds)
```

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

1. : **Evaluasi Kondisi:**

- **while:** Kondisi pengulangan dievaluasi sebelum blok pernyataan dijalankan. Jika kondisi awalnya false, blok pernyataan tidak pernah dijalankan.

- **do-while:** Blok pernyataan dijalankan sekali sebelum kondisi pengulangan dievaluasi. Ini berarti bahwa setidaknya satu iterasi akan dilakukan, bahkan jika kondisi awalnya false.

2. Keberadaan Kondisi di Awal atau Akhir:

- **while:** Kondisi pengulangan ditentukan di awal, sebelum blok pernyataan dijalankan.
- **do-while:** Kondisi pengulangan ditentukan di akhir, setelah blok pernyataan dijalankan setidaknya sekali.

3. Keberadaan Blok Pernyataan Minimal Satu Kali:

- **while:** Blok pernyataan hanya akan dieksekusi jika kondisi awalnya true.
- **do-while:** Blok pernyataan akan dieksekusi setidaknya sekali, bahkan jika kondisi awalnya false.

5. Kapan digunakan struktur kontrol for?

:

1. **iterasi Melalui Rentang Angka:** Ketika Anda perlu melakukan iterasi melalui rentang angka tertentu, seperti dari 1 hingga N, di mana N sudah diketahui sebelumnya.
2. **Iterasi Melalui Elemen Kumpulan Data:** Ketika Anda perlu melakukan iterasi melalui setiap elemen dalam sebuah array, daftar, atau struktur data lainnya.
3. **Iterasi dengan Langkah Tertentu:** Ketika Anda perlu melakukan iterasi dengan langkah yang telah ditentukan, misalnya, menghitung mundur dari N hingga 1 dengan langkah -1.
4. **Iterasi dengan Inisialisasi, Kondisi, dan Perubahan:** Ketika Anda perlu menyertakan semua langkah-langkah iterasi (inisialisasi, kondisi, dan perubahan) dalam satu baris kode untuk kejelasan dan keterbacaan.
5. **Iterasi dengan Penanganan Indeks:** Ketika Anda perlu menggunakan variabel iterasi (biasanya disebut indeks) untuk mengakses elemen tertentu dalam sebuah kumpulan data.

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

:

1. Fleksibilitas Ukuran:	<ul style="list-style-type: none"> • Array: Ukuran tetap dan tidak bisa diubah. • ArrayList: Ukuran dapat disesuaikan secara dinamis.
2. Tipe Data:	<ul style="list-style-type: none"> • Array: Dapat menampung tipe data primitif dan objek. • ArrayList: Hanya dapat menampung objek, bukan tipe data primitif.
3. Manipulasi Elemen:	<ul style="list-style-type: none"> • Array: Manipulasi langsung dengan indeks. • ArrayList: Manipulasi menggunakan metode kelas ArrayList.

contoh penggunaan array dalam java

```
int[] angka = new int[5];

// Mengisi array dengan nilai
angka [0] = 10;
angka [1] = 20;
angka [2] = 30;
angka [3] = 40;
angka [4] = 50;

// Mengakses nilai array
System.out.println(angka [2])
```

Contoh penggunaan ArrayList dalam java

```
import java.util.ArrayList;

// Deklarasi dan inisialisasi ArrayList integer
ArrayList<Integer> numbersList = new ArrayList<>();

// Menambah elemen ke ArrayList
numbersList.add(10);
numbersList.add(20);
numbersList.add(30);
numbersList.add(40);
numbersList.add(50);

// Mengakses nilai ArrayList
System.out.println(numbersList.get(2)); // Output: 30
```

```
// Menghapus elemen dari ArrayList  
numbersList.remove(3); // Menghapus elemen ke-4  
System.out.println(numbersList);
```

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

EVALUASI PRAKTIKUM [Modul 4]

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!

: **Pewarisan (Inheritance):** Dalam paradigma pemrograman berorientasi objek, pewarisan memungkinkan sebuah kelas untuk mewarisi atribut dan metode dari kelas lain. Misalnya, dalam program, kelas Mahasiswa bisa mewarisi sifat-sifat dari kelas Orang. Dalam pewarisan, penggunaan kata kunci "extends" menunjukkan bahwa kelas Mahasiswa mewarisi semua atribut dan metode yang didefinisikan dalam kelas Orang. Dengan demikian, kelas Mahasiswa akan memiliki atribut nama yang telah didefinisikan di kelas Orang.

Polimorfisme: Polimorfisme adalah konsep di mana suatu objek dapat mengambil banyak bentuk atau perilaku. Dalam konteks pewarisan, polimorfisme terjadi ketika kelas turunan (subclass) dapat menggunakan metode yang sama dengan kelas dasar (superclass), tetapi juga bisa memiliki perilaku yang berbeda. Dalam program, polimorfisme terjadi karena kelas Mahasiswa memiliki metode konstruktor yang sama dengan kelas Orang, yaitu "public Mahasiswa()". Ini berarti bahwa kelas Mahasiswa menggunakan metode konstruktor kelas Orang saat membuat objek Mahasiswa tanpa perlu mendefinisikan ulang konstruktor tersebut.

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.

:



