TUGAS 4

PEMROGRAMAN BERORIENTASI OBJEK

Nama : Rahman Ramadhan

Kelas ; A4

Nim : 13020220128

Evaluasi Praktikum (Modul 3 Struktur control dan array)

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

Jawab:

1. Penulisan Kode:
   - if-else: Menjalankan blok kode berdasarkan kondisi yang benar atau salah.
   - switch-case: Memilih blok kode berdasarkan nilai tertentu dari ekspresi.

2. Kondisi:
   - if-else: Menggunakan kondisi boolean atau ekspresi yang menghasilkan nilai boolean.
   - switch-case: Menggunakan nilai yang dibandingkan dengan beberapa kasus yang telah ditentukan.
3. Penggunaan:
   - if-else: Cocok untuk kondisi dengan logika yang berbeda.
   - switch-case: Lebih cocok untuk memeriksa beberapa nilai yang mungkin dihasilkan oleh satu variabel.
4. Kemungkinan Percabangan:
   - if-else: Tidak terbatas.
   - switch-case: Hanya untuk beberapa nilai diskrit.

5. Fleksibilitas:
    - if-else: Fleksibel, mendukung ekspresi boolean kompleks.
    - switch-case: Lebih terbatas, hanya membandingkan nilai diskrit.

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

Jawab :

Gunakan if-else ketika:

1. Anda memiliki beberapa kondisi yang berbeda dan setiap kondisi memerlukan logika yang berbeda.
2. Kondisi Anda kompleks dan memerlukan pengecekan boolean yang lebih rumit.
3. Anda perlu mengecek kondisi berurutan dan berdasarkan pada hasilnya, menjalankan blok kode yang sesuai.
4. Tidak ada pola yang jelas dalam nilai yang dievaluasi.

Gunakan switch-case ketika:

1. Anda memiliki satu variabel atau ekspresi dan ingin mengevaluasi beberapa nilai yang mungkin dihasilkan oleh variabel tersebut.
2. Anda memiliki pola nilai yang jelas yang ingin Anda bandingkan.
3. Anda ingin menghindari penulisan banyak pernyataan if-else berturut-turut untuk kondisi yang terpisah.
4. Anda perlu meningkatkan kejelasan dan membaca kodenya dengan lebih mudah ketika membandingkan beberapa nilai tertentu.

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

opsi pilihah 1=inputNilai() dan Pilihan 2=inputNilaiBaru() jawab :

```
package pertemuan2.modul3; import
java.util.Scanner;
```

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

        Scanner input = new Scanner(System.in);
        HitungRata hitung = new HitungRata();

        boolean isRunning = true;
while (isRunning) {
            System.out.println("Menu:");
            System.out.println("1. Input Nilai");
            System.out.println("2. Input Nilai Baru");
            System.out.println("3. Keluar");
System.out.print("Pilih opsi: ");
int pilihan = input.nextInt();

            switch (pilihan) {
case 1:
                    System.out.print("Masukkan Jumlah Data :
");                     int banyakData = input.nextInt();
int nilai[] = new int[banyakData];
System.out.print("Masukkan Nilai : ");
hitung.inputNilai(nilai);
                    System.out.print("Daftar Nilai : ");
hitung.cetakNilai(nilai);
                    System.out.println("Rata Nilai : " + hitung.rataNilai(banyakData));
                    break;
case 2:
                    System.out.print("Masukkan Nilai Baru: ");                     hitung.inputNilaiBaru(banyakData);
// Pastikan banyakData sudah dideklarasikan sebelumnya                     System.out.print("Daftar Nilai Baru
: ");                     hitung.cetakNilaiBaru();
```

```
                break;
case 3:
                isRunning = false;
                System.out.println("Program berhenti.");
                break;
default:
                System.out.println("Pilihan tidak valid. Silakan pilih opsi 1, 2, atau 3.");
                break;
            }
        }
        input.close();
    }
}
```

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

Jawab :

1. **Evaluasi Kondisi:**
    - **while**: Pertama-tama, kondisi di evaluasi. Jika kondisi benar, blok kode dalam pernyataan **while** akan dieksekusi.
    - **do-while**: Pertama-tama, blok kode dalam pernyataan **do** akan dieksekusi, kemudian kondisi di evaluasi. Jika kondisi benar, perulangan akan dilanjutkan.
2. **Eksekusi Blok Kode:**
    - **while**: Blok kode hanya akan dieksekusi jika kondisi awalnya benar.
    - **do-while**: Blok kode akan dieksekusi setidaknya satu kali, bahkan jika kondisi awalnya salah.
3. **Kemungkinan untuk Tidak Dieksekusi:**
    - **while**: Ada kemungkinan bahwa blok kode dalam pernyataan **while** tidak akan pernah dieksekusi jika kondisinya salah dari awal.
    - **do-while**: Blok kode dalam pernyataan **do** akan selalu dieksekusi setidaknya satu kali, bahkan jika kondisi awalnya salah.

5. Kapan digunakan struktur kontrol for?

Jawab :

Struktur kontrol **for** digunakan saat ingin melakukan iterasi atau perulangan sejumlah tertentu kali atau untuk setiap elemen dalam sebuah koleksi. Penggunaan yang umum dari struktur **for** meliputi:

1. Iterasi Melalui Sebuah Rentang Nilai:
    - Contoh: Menghitung jumlah total dalam rentang nilai tertentu.
2. Iterasi Melalui Koleksi Objek:
    - Contoh: Mengakses setiap elemen dalam sebuah array atau list.
3. Iterasi dengan Langkah Tertentu:
    - Contoh: Melangkahi setiap dua elemen dalam sebuah list atau array.
4. Iterasi dengan Menggunakan Enumerate:
    - Contoh: Mengakses indeks dan nilai dalam sebuah list secara bersamaan.
5. Iterasi Melalui Struktur Data Lainnya:
    - for key, value in dictionary.items(): Untuk mengakses kunci dan nilai dalam sebuah dictionary.
    - for element in set: Untuk mengakses setiap elemen dalam sebuah set.

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

Jawab :

Array:

- Array adalah struktur data yang menyimpan sejumlah elemen dengan tipe data yang sama secara terus-menerus dalam memori.
- Ukuran array didefinisikan pada saat pembuatannya dan tidak bisa diubah setelahnya.
- Array dapat menyimpan elemen primitif maupun objek.
- Array memiliki sintaksis khusus dalam deklarasinya

Contoh penggunaan array dalam java

```
int[] Arraysaya = new int[5]; // Mendeklarasikan array dengan panjang 5
Arraysaya [0] = 10;
Arraysaya [1] = 20;
```

Arraysaya [2] = 30;

Arraysaya [3] = 40;

Arraysaya [4] = 50;

ArrayList:

- ArrayList adalah kelas dalam Java yang menyediakan implementasi dari daftar dinamis. Ini berarti ukuran ArrayList dapat berubah-ubah.
- Anda tidak perlu menentukan ukuran ArrayList pada saat pembuatan, dan ArrayList akan secara otomatis mengelola ukurannya sesuai kebutuhan.
- ArrayList hanya dapat menyimpan objek, tidak dapat menyimpan tipe data primitif. Jika Anda perlu menyimpan tipe data primitif, Anda harus menggunakan kelas pembungkus (wrapper class) seperti Integer, Double, dll.
- Untuk menggunakan ArrayList, Anda perlu mengimpor paket **java.util.ArrayList** dan menginisialisasi ArrayList menggunakan konstruktor

import java.util.ArrayList;

ArrayList<Integer> ArrayListSy = new ArrayList<>(); // Mendeklarasikan ArrayList tanpa menentukan ukuran

ArrayListSy.add(10);

ArrayListSy.add(20);

ArrayListSy.add(30);

ArrayListSy.add(40);

ArrayListSy.add(50);

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

Jawab :

```
=== Masukkan kunci-nilai ke dalam HashMap (ketik 'keluar' untuk mengakhiri) ===
Masukkan kunci(Karakter)  : Rahman
Masukkan nilai(Bil.Bulat) : 10
Masukkan kunci(Karakter)  : Ali
Masukkan nilai(Bil.Bulat) : 20
Masukkan kunci(Karakter)  : Bambang
Masukkan nilai(Bil.Bulat) : 50
Masukkan kunci(Karakter)  : keluar

Isi HashMap :
Kunci : Rahman, Nilai : 10
Kunci : Bambang, Nilai : 50
Kunci : Ali, Nilai : 20
```

Evaluasi Praktikum (Modul 4 Konsep Pemrograman Berorientasi Objek)

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!

Jawab :

Pewarisan (Inheritance): Pewarisan adalah konsep dalam pemrograman berorientasi objek di mana sebuah kelas dapat mewarisi sifat-sifat (atribut dan metode) dari kelas lain. Dalam program Anda, kelas Mahasiswa mewarisi sifat-sifat dari kelas Orang.

Dalam kelas Mahasiswa, penggunaan kata kunci extends Orang menunjukkan bahwa kelas Mahasiswa mewarisi semua atribut dan metode yang didefinisikan dalam kelas Orang. Dengan demikian, kelas Mahasiswa akan memiliki atribut nama yang didefinisikan di kelas Orang.

Polimorfisme: Polimorfisme adalah konsep di mana suatu objek dapat memiliki banyak bentuk atau perilaku. Dalam konteks pewarisan, polimorfisme terjadi ketika kelas turunan (subclass) dapat menggunakan metode yang sama dengan kelas dasar (superclass) tetapi juga dapat memiliki perilaku yang berbeda.

Dalam program Anda, polimorfisme terjadi karena kelas Mahasiswa memiliki metode konstruktor yang sama seperti kelas Orang, yaitu public Mahasiswa(). Ini berarti bahwa kelas Mahasiswa menggunakan metode konstruktor kelas Orang saat membuat objek Mahasiswa tanpa harus 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)!

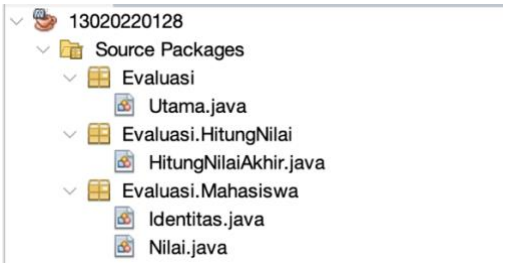
Jawab :

```
D:\Tugas 3>java Utama
Stb : ALI
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.


Source Packages
Evaluasi
Utama.java
Evaluasi.HitungNilai
HitungNilaiAkhir.java
Evaluasi.Mahasiswa
Identitas.java
Nilai.java

Jawab :


13020220128
Source Packages
Evaluasi
Utama.java
Evaluasi.HitungNilai
HitungNilaiAkhir.java
Evaluasi.Mahasiswa
Identitas.java
Nilai.java