

NAMA : Wahyu Kadri Rahmat Suat

NIM : 13020220081

KELAS : A2

TUGAS : PBO 4

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

- if-else digunakan ketika terdapat beberapa kondisi yang saling terkait atau berbeda satu sama lain. Ini memungkinkan untuk mengevaluasi kondisi yang kompleks dan memilih jalur eksekusi berdasarkan kebenaran atau kebenaran kondisi tersebut.
- switch-case digunakan ketika ada satu variabel atau ekspresi yang membutuhkan evaluasi nilai terhadap beberapa kasus atau kemungkinan. Ini lebih berguna ketika ada banyak kondisi yang mungkin dievaluasi dengan nilai yang sama.

Pemilihan antara if-else dan switch-case tergantung pada kebutuhan spesifik dari algoritma atau logika yang ingin diimplementasikan, serta fitur-fitur yang disediakan oleh bahasa pemrograman yang digunakan.

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

➤ gunakan struktur kontrol if-else ketika:

- Anda memiliki kondisi yang kompleks dan tidak terbatas pada perbandingan nilai tetap.
- Anda perlu mengevaluasi ekspresi boolean atau kondisi yang terkait dengan logika yang kompleks.
- Anda memerlukan kontrol alur program yang sangat fleksibel dan dapat menangani banyak kasus yang berbeda dengan logika yang berbeda.

➤ Sementara itu, gunakan struktur kontrol switch-case ketika:

- Anda perlu membandingkan satu nilai dengan beberapa nilai konstan atau tetap.
- Kode yang mudah dibaca dan dipahami, terutama jika ada banyak kondisi yang bergantung pada nilai yang sama.

- Anda ingin mengoptimalkan kinerja, karena dalam beberapa bahasa pemrograman, switch-case dapat diimplementasikan lebih efisien daripada serangkaian pernyataan if-else.
3. Pada program 2, tambahkan perintah untuk memilih 2 opsi menggunakan kontrol switch..case. opsi pilihah 1=inputNilai() Pilihan 2=inputNilaiBa

```

1  import java.util.*;
2
3  public class HitungRata {
4      private double total = 0.0;
5      private ArrayList<Integer> nilaiBaru = new ArrayList<>();
6      Scanner input = new Scanner(System.in);
7
8      public void inputNilai(int nilai[]) {
9          for (int i = 0; i < nilai.length; i++) {
10             nilai[i] = input.nextInt();
11             total += nilai[i];
12         }
13     }
14
15     public double rataNilai(int Ndata) {
16         return total / ((double) Ndata);
17     }
18
19     public void cetakNilai(int nilai[]) {
20         for (int angka : nilai) {
21             System.out.print(angka + "\t");
22         }
23         System.out.println();
24     }
25
26     public void inputNilaiBaru(int jumlah) {
27         while (jumlah > 0) {
28             nilaiBaru.add(input.nextInt());
29             jumlah--;
30         }
31     }
32
33     public void cetakNilaiBaru() {
34         ListIterator<Integer> i = nilaiBaru.listIterator(index:0);
35         while (i.hasNext()) {
36             Integer data = i.next();
37             if (data == null) {
38                 System.out.println(x:"null");
39             } else {

```

```

33     public void cetakNilaiBaru() {
34         ListIterator<Integer> i = nilaiBaru.listIterator(index:0);
35         while (i.hasNext()) {
36             Integer data = i.next();
37             if (data == null) {
38                 System.out.println(x:"null");
39             } else {
40                 System.out.println(data.toString());
41             }
42         }
43     }
44 }

```

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

Perbedaan utama antara while dan do while adalah pada do while, perulangan akan dijalankan satu kali bahkan sebelum kondisi diperiksa, sedangkan pada perulangan while, perulangan akan dijalankan jika kondisinya benar

5. Kapan digunakan struktur kontrol for?

struktur kontrol for cocok digunakan ketika Anda memiliki ide tentang berapa kali iterasi harus dilakukan atau ketika Anda perlu melakukan iterasi melalui kumpulan data dalam rangkaian numerik atau terstruktur.

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

- Array adalah struktur data yang digunakan untuk menyimpan sekelompok elemen dengan tipe data yang sama. Saat deklarasi, ukuran array ditentukan dan tidak dapat diubah setelahnya. Akses atau manipulasi elemen dalam array dilakukan dengan menggunakan indeks. Array dapat berisi tipe data primitif seperti int, double, char, atau objek seperti String.

```

int[] angka = new int[5];
angka[0] = 10;

```

- ArrayList merupakan kelas dalam Java yang terdapat di paket java.util dan mewakili implementasi dari antarmuka List. Ini digunakan untuk menyimpan koleksi elemen dengan tipe data yang sama atau berbeda. Ukuran ArrayList bersifat fleksibel dan dapat berubah secara dinamis saat program berjalan. Untuk mengakses atau memanipulasi elemen dalam ArrayList, berbagai metode disediakan oleh kelas ArrayList, seperti add(), get(), set(), dan lainnya. Namun, perlu diingat bahwa ArrayList hanya dapat menyimpan objek, sehingga jika ingin menyimpan tipe primitif, perlu menggunakan kelas pembungkus (wrapper class) seperti Integer, Double, Character, dan sebagainya.

```
import java.util.ArrayList;

ArrayList<Integer> angka = new ArrayList<Integer>();
angka.add(10);
```

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

JAWABAN EVALUASI 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!

Class mahasiswa yang menjadi class pewarisan dari class utama yaitu class orang, bisa kita lihat pada class mahasiswa yang di mana dia mendelarasikan 'extend' yang dimana dari kelas mahasiswa akan mewariskan semua variabel dan metode yang didefenisikan didalam class orang

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

```

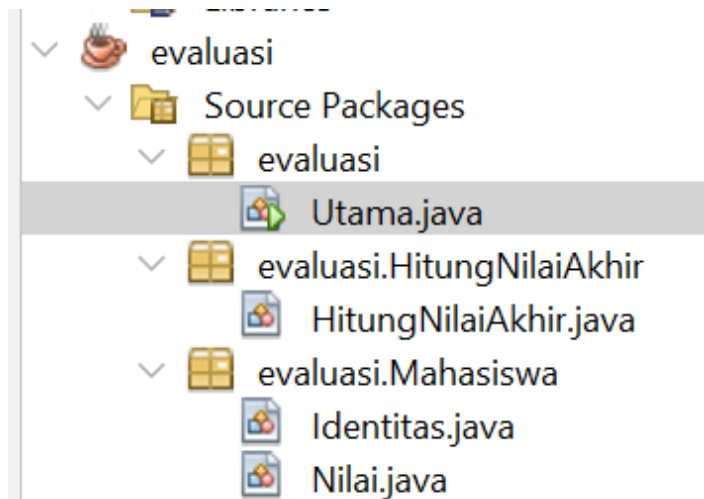
5 package javaapplication4;
6
7 public class Orang {
8     public String nama;
9     public Orang() { this.nama= "Aminah";}
10    public Orang(String nama) {
11        this.nama = nama;
12    }
13    public void info(){
14        System.out.println("Nama " + this.nama);
15    }
16 }

5 package javaapplication4;
6
7 public class Utama {
8     public static void main(String[] args) {
9         Orang orang = new Orang();
10        orang.nama = "Wahyu"; // isi sesuai nama anda
11        orang.info();
12    }
13 }

5 package javaapplication4;
6
7 public class mahasiswa extends Orang{
8     private String stb;
9     public mahasiswa() {
10        super();
11        this.stb="1302002134"; //stambuk anda
12    }
13    public mahasiswa(String stb, String nama) {
14        this.nama = nama;
15        this.stb = stb;
16    }
17 }

```

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



```
1 package evaluasi;
2 import java.io.IOException;
3 import java.util.Scanner;
4 import javax.swing.JOptionPane;
5
6 import evaluasi.HitungNilaiAkhir.HitungNilaiAkhir;
7 import evaluasi.mahasiswa.Identitas;
8 import evaluasi.mahasiswa.Nilai;
9
10 public class Utama {
11     public static void main(String[] args) throws IOException {
12         Identitas identitas = new Identitas();
13         Nilai nilai = new Nilai();
14         HitungNilaiAkhir hitungNilai = new HitungNilaiAkhir();
15
16         Scanner scanner = new Scanner(System.in);
17         System.out.print("Masukkan Nama      : ");
18         identitas.setNama(scanner.nextLine());
19         System.out.print("Masukkan Stambuk    : ");
20         identitas.setStambuk(scanner.nextLine());
21         System.out.println("");
22         System.out.print("Masukkan Nilai Tugas 1: ");
23         nilai.setTugas1(scanner.nextInt());
24         System.out.print("Masukkan Nilai Tugas 2: ");
25         nilai.setTugas2(scanner.nextInt());
26
27         nilai.setTugas2(scanner.nextInt());
28         System.out.println("");
29         System.out.print("Masukkan Nilai MID    : ");
30         nilai.setMid(scanner.nextInt());
31         System.out.print("Masukkan Nilai UAS    : ");
32         nilai.setUas(scanner.nextInt());
33
34         double tugas = hitungNilai.NilaiTugas(nilai.getTugas1(), nilai.getTugas2());
35         double na = hitungNilai.nilaiAkhir(tugas, nilai.getMid(), nilai.getUas());
36
37         JOptionPane.showMessageDialog(null, "Nama: " + identitas.getNama() + "\nStambuk: " + identitas.getStambuk());
38         JOptionPane.showMessageDialog(null, "Nilai Tugas: " + tugas + "\n\nNilai MID: " + nilai.getMid() + "\n\nNilai UAS: "
39     }
40 }
```

```
Stambuk());  
    "\n\nNilai UAS: " + nilai.getUas() + "\n\nNilai Akhir: " + na);
```

```
5     package evaluasi.HitungNilaiAkhir;  
6  
7     public class HitungNilaiAkhir {  
8         public double NilaiTugas(int tugas1, int tugas2) {  
9             return (tugas1 + tugas2) * 2.0;  
10        }  
11  
12        public double nilaiAkhir(double tugas, int mid, int uas) {  
13            return (tugas * 0.4) + (mid * 0.3) + (uas * 0.3);  
14        }  
15    }
```

```
1     package evaluasi.mahasiswa;  
2  
3     public class Identitas {  
4         private String nama;  
5         private String stambuk;  
6  
7         public void setNama(String nama){  
8             this.nama = nama;  
9         }  
10        public String getNama(){  
11            return nama;  
12        }  
13        public void setStambuk(String stambuk){  
14            this.stambuk = stambuk;  
15        }  
16        public String getStambuk(){  
17            return stambuk;  
18        }  
19  
20  
21    }
```

```
1 package evaluasi.mahasiswa;
2
3 public class Nilai {
4     private int tugas1;
5     private int tugas2;
6     private int mid;
7     private int uas;
8
9     public void setTugas1(int tugas1) {
10         this.tugas1 = tugas1;
11     }
12
13     public int getTugas1() {
14         return tugas1;
15     }
16
17     public void setTugas2(int tugas2) {
18         this.tugas2 = tugas2;
19     }
20
21     public int getTugas2() {
22         return tugas2;
23     }
24
25     public void setMid(int mid) {
26         this.mid = mid;
27     }
28
29     public int getMid() {
30         return mid;
31     }
32
33     // Setter dan Getter untuk variabel uas
34     public void setUas(int uas) {
35         this.uas = uas;
36     }
37
38     public int getUas() {
39         return uas;
40     }
41 }
```