

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : M. Akbar Amiruddin
Nim : 13020220023
Kelas : A1

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR

2024

MODUL 3 STRUKTUR KONTROL DAN ARRAY

EVALUASI PRAKTIKUM

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

=

- If-else:
 - If-else digunakan untuk memeriksa kondisi tunggal atau serangkaian kondisi secara berurutan.
 - Bisa menangani kondisi yang kompleks dengan menggunakan kondisi bersarang (nested conditions).
 - Cocok digunakan ketika terdapat kondisi yang beraneka ragam dan tidak terbatas pada nilai tertentu.
- Switch-case:
 - Switch-case digunakan ketika ada banyak pilihan yang harus dievaluasi berdasarkan nilai ekspresi tunggal.
 - Lebih efisien daripada if-else dalam beberapa kasus, terutama ketika ada banyak kemungkinan nilai yang berbeda.
 - Dibatasi oleh fakta bahwa hanya bisa mengevaluasi ekspresi yang sesuai dengan tipe data konstan (biasanya integer atau karakter).

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

=

- Struktur kontrol if-else digunakan ketika kita perlu mengevaluasi suatu kondisi dan melakukan tindakan berdasarkan hasil evaluasi tersebut. Jika kondisi yang dievaluasi benar, blok kode di dalam if akan dieksekusi; jika tidak, blok kode di dalam else (opsional) akan dieksekusi.
- Switch-case digunakan ketika kita memiliki serangkaian nilai yang mungkin untuk sebuah ekspresi dan ingin melakukan tindakan berbeda berdasarkan nilai yang diberikan. Ini adalah alternatif yang lebih bersih daripada menggunakan serangkaian if-else berturut-turut. Switch-case cocok ketika kita memiliki banyak kasus yang spesifik dan ingin menghindari berulangnya pengecekan kondisi yang sama.

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

- opsi pilihan 1=inputNilai()
- Pilihan 2=inputNilaiBaru()

=

```
import java.util.Scanner;  
public class TestNilai {
```

```

public static void main(String[] args) {
    HitungRata hitung = new HitungRata();
    Scanner input = new Scanner(System.in);

    System.out.println("Menu:");
    System.out.println("1. Input Nilai");
    System.out.println("2. Input Nilai Baru");
    System.out.print("Pilih menu: ");
    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 Jumlah Nilai Baru: ");
            int jumlahNilaiBaru = input.nextInt();
            hitung.inputNilaiBaru(jumlahNilaiBaru);
            System.out.print("Daftar Nilai Baru: ");
            hitung.cetakNilaiBaru();
            break;
        default:
            System.out.println("Pilihan tidak valid.");
            break;
    }
}
}

```

4. Apakah perbedaan antara struktur kontrol perulangan while dan do-while?
 - While adalah jenis pengulangan yang digunakan ketika ingin melakukan iterasi selama sebuah kondisi tertentu terpenuhi.
 - Do-while loop adalah jenis pengulangan yang setidaknya satu iterasi akan dieksekusi sebelum kondisi dievaluasi, bahkan jika kondisi di awalnya tidak terpenuhi.

5. Kapan digunakan struktur kontrol for?

=

Struktur kontrol for digunakan ketika Anda perlu melakukan iterasi atau perulangan melalui serangkaian instruksi dengan jumlah iterasi yang diketahui sebelumnya. Ini sangat berguna ketika Anda ingin menjalankan blok kode sejumlah tertentu atau ketika Anda perlu mengulangi sebuah blok kode dengan langkah iterasi yang teratur.

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

=

Perbedaan utama antara Array dan ArrayList terletak pada fleksibilitas dan kemampuan mereka untuk menangani perubahan ukuran data.

1. Array:

- Ukuran array ditentukan pada saat pembuatan dan tidak dapat diubah setelahnya.
- Elemen-elemen array harus memiliki tipe data yang sama.
- Penanganan data statis, sehingga tidak dapat menambah atau mengurangi elemen secara dinamis setelah array dibuat.

Contoh:

```
// Deklarasi dan inisialisasi array
int[] numbers = new int[5];

// Mengakses elemen array
numbers[0] = 1;
numbers[1] = 2;
```

2. ArrayList:

- ArrayList adalah bagian dari Java Collections Framework dan merupakan implementasi dari antarmuka List.
- Ukuran ArrayList dapat berubah secara dinamis saat program berjalan.
- Dapat menyimpan elemen dengan tipe data yang berbeda (jika menggunakan tipe data Object atau menggunakan generics).

Contoh:

```
import java.util.ArrayList;

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

// Menambahkan elemen ke ArrayList
numbers.add(1);
numbers.add(2);

// Mengakses elemen ArrayList
int firstElement = numbers.get(0);
```

Dengan ArrayList, Anda dapat menambahkan atau menghapus elemen secara dinamis sesuai kebutuhan aplikasi, sementara Array lebih cocok untuk situasi di mana ukuran data tetap atau tidak berubah setelah pembuatan.

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

=

```
import java.util.HashMap;
```

```

import java.util.Scanner;

public class ContohHashMap {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        HashMap<String, Integer> hashMap = new HashMap<>();

        System.out.print("Masukkan jumlah data yang ingin
dimasukkan: ");
        int jumlahData = scanner.nextInt();
        scanner.nextLine(); // Membuang newline character

        for (int i = 0; i < jumlahData; i++) {
            System.out.print("Masukkan key: ");
            String key = scanner.nextLine();

            System.out.print("Masukkan nilai: ");
            int nilai = scanner.nextInt();
            scanner.nextLine(); // Membuang newline character

            hashMap.put(key, nilai);
        }

        System.out.println("\nHashMap setelah dimasukkan:");
        for (String key : hashMap.keySet()) {
            System.out.println("Key: " + key + ", Nilai: " +
hashMap.get(key));
        }
    }
}

```

MODUL 4 KONSEP PEMROGRAMAN BERORIENTASI OBJEK

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!

=

- Pewarisan (Inheritance):

Pewarisan adalah konsep dalam pemrograman berorientasi objek di mana sebuah class dapat mewarisi atribut dan metode dari class lainnya. Ini memungkinkan class turunan (subclass) untuk memperluas atau memodifikasi perilaku class induk (superclass). Konsep ini ditemukan dalam program 3, di mana class `Mahasiswa` merupakan subclass dari class `Orang`. Dengan demikian, `Mahasiswa` mewarisi atribut `nama` dan konstruktor dari class `Orang`.

- Polimorfisme:

Polimorfisme adalah konsep di mana suatu objek dapat memiliki banyak bentuk. Ini memungkinkan sebuah objek dapat berperilaku berbeda-beda tergantung pada konteksnya. Konsep ini juga ditemukan dalam program 3, di mana method `info()` pada class `Orang` dan class `Mahasiswa` memiliki nama yang sama tetapi perilaku yang berbeda. Ketika method `info()` dipanggil untuk objek `Mahasiswa`, itu mencetak nama dan stambuk mahasiswa, sedangkan ketika dipanggil untuk objek `Orang`, itu hanya mencetak nama.

Jadi, sementara program 3 menunjukkan kedua konsep pewarisan dan polimorfisme, program 1 dan 2 lebih fokus pada konsep pewarisan karena tidak ada penambahan perilaku yang berbeda untuk objek subclass.

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

- Utama.java

```
7 Nama      : M. Akbar Amiruddin
8 NIM       : 13020220023
9 Hari/Tgl  : Rabu/27-03-2024
10 Pengerjaan : 09.00 - 10.50
11 Materi    : Evaluasi Praktikum Modul 4
12 */
13
14 package pertemuan2.modul4.user;
15
16 public class Utama {
17     public static void main(String[] args) {
18         Orang orang = new Orang();
19         orang.nama = "ALI"; // isi sesuai nama anda
20         System.out.println("Stb : " + orang.nama);
21
22         // Memanggil method info() dari Class Orang
23         Orang.info();
24
25         // Menambahkan objek mhs1 dengan konstruktor default Mahasiswa()
26         Mahasiswa mhs1 = new Mahasiswa();
27         System.out.println("Nama Mahasiswa: " + mhs1.getNama());
28         System.out.println("Stambuk Mahasiswa: " + mhs1.getStb());
29
30         // Memanggil method info() dari Class Mahasiswa
31         Mahasiswa.info();
32     }
33 }
```

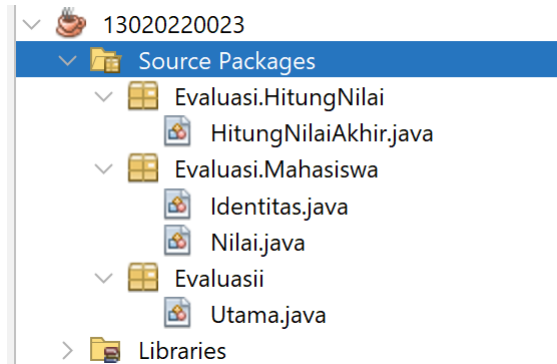
- Utama.java

```
7  Nama      : M. Akbar Amiruddin
8  NIM       : 13020220023
9  Hari/Tgl  : Rabu/27-03-2024
10 Pengerjaan : 09.00 - 10.50
11 Materi    : Evaluasi Praktikum Modul 4
12 */
13
14 package pertemuan2.modul4.userr;
15
16 public class Orang {
17     protected String nama;
18
19     public Orang() {
20         this.nama = "Aminah";
21     }
22
23     public Orang(String nama) {
24         this.nama = nama;
25     }
26
27     public String getNama() {
28         return nama;
29     }
30
31     public void setNama(String nama) {
32         this.nama = nama;
33     }
34
35     public static void info() {
36         System.out.println("Ini adalah info dari Class Orang");
37     }
38 }
```

- Mahasiswa.java

```
7  Nama      : M. Akbar Amiruddin
8  NIM       : 13020220023
9  Hari/Tgl  : Rabu/27-03-2024
10 Pengerjaan : 09.00 - 10.50
11 Materi    : Evaluasi Praktikum Modul 4
12 */
13
14 package pertemuan2.modul4.userr;
15
16 public class Mahasiswa extends Orang {
17     private String stb;
18
19     public Mahasiswa() {
20         super();
21         this.stb = "1302002134"; // stambuk anda
22     }
23
24     public Mahasiswa(String stb, String nama) {
25         super(nama);
26         this.stb = stb;
27     }
28
29     // Getter untuk mendapatkan nilai stb
30     public String getStb() {
31         return stb;
32     }
33
34     // Setter untuk mengatur nilai stb
35     public void setStb(String stb) {
36         this.stb = stb;
37     }
38
39     // Method info untuk mencetak nama dan stambuk
40     public static void info() {
41         System.out.println("Ini adalah info dari Class Mahasiswa");
42     }
43 }
```

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



Lengkapi Program berikut :

- Identitas.java

```
6      /*
7      Nama          : M. Akbar Amiruddin
8      NIM           : 13020220023
9      Hari/Tgl      : Rabu/27-03-2024
10     Pengerjaan    : 09.00 - 10.50
11     Materi        : Evaluasi Praktikum Modul 4
12     */
13
14     package Evaluasi.Mahasiswa;
15
16     public class Identitas {
17         public String nama;
18         public String stambuk;
19
20         public void setNama (String nama){
21             this.nama = nama;
22         }
23
24         public String getNama () {
25             return nama;
26         }
27
28         public void setStambuk (String stambuk){
29             this.stambuk = stambuk;
30         }
31
32         public String getstambuk () {
33             return stambuk;
34         }
35     }
```


- Nilai.java

```

8      NIM      : 13020220023
9      Hari/Tgl   : Rabu/27-03-2024
10     Pengerjaan : 09.00 - 11.00
11     Materi      : Evaluasi Praktikum Modul 4
12     */
13     package Evaluasi.Mahasiswa;
14
15     public class Nilai {
16         private int tugas1;
17         private int tugas2;
18         private int mid;
19         private int uas;
20
21         public void setTugas1(int tugas1){
22             this.tugas1 = tugas1;
23         }
24         public int getTugas1(){
25             return tugas1;
26         }
27         public void setTugas2(int tugas2){
28             this.tugas2 = tugas2;
29         }
30         public int getTugas2(){
31             return tugas2;
32         }
33         public void setMid(int mid){
34             this.mid = mid;
35         }
36         public int getMid(){
37             return mid;
38         }
39         public void setuas (int uas){
40             this.uas = uas;
41         }
42         public int getuas(){
43             return uas;
44         }
45     }

```

- HitungNilaiAkhir.java

```

6      /*
7      Nama      : M. Akbar Amiruddin
8      NIM       : 13020220023
9      Hari/Tgl  : Rabu/27-03-2024
10     Pengerjaan : 09.00 - 11.00
11     Materi     : Evaluasi Praktikum Modul 4
12     */
13     package Evaluasi.HitungNilai;
14
15     public class HitungNilaiAkhir{
16         public double nilaiTugas(int tugas1, int tugas2){
17             double rataTugas = (tugas1 + tugas2) / 2.0;
18             return rataTugas;
19         }
20
21         public double nilaiAkhir(double tugas, int mid, int uas){
22             double nilaiAkhir = (tugas * 0.4) + (mid * 0.3) + (uas * 0.3);
23             return nilaiAkhir;
24         }
25     }

```

- Utama.java

```
7 Nama      : M. Akbar Amiruddin
8 NIM       : 13020220023
9 Hari/Tgl  : Rabu/27-03-2024
10 Pengerjaan : 09.00 - 10.50
11 Materi    : Evaluasi Praktikum Modul 4
12 */
13 package Evaluasii;
14
15 import Evaluasi.HitungNilai.HitungNilaiAkhir;
16 import Evaluasi.Mahasiswa.*;
17
18 import javax.swing.JOptionPane;
19
20 public class Utama{
21
22     public static void main(String[] args) {
23         String nama = JOptionPane.showInputDialog("Masukkan Nama:");
24         String stambuk = JOptionPane.showInputDialog("Masukkan Stambuk:");
25
26         int tugas1 = Integer.parseInt(JOptionPane.showInputDialog("Masukkan Tugas 1:"));
27         int tugas2 = Integer.parseInt(JOptionPane.showInputDialog("Masukkan Tugas 2:"));
28         int mid = Integer.parseInt(JOptionPane.showInputDialog("Masukkan Nilai MID:"));
29         int uas = Integer.parseInt(JOptionPane.showInputDialog("Masukkan Nilai UAS:"));
30
31         Identitas mhs = new Identitas();
32         mhs.setNama(nama);
33         mhs.setStambuk(stambuk);
34
35         HitungNilaiAkhir hitungNilai = new HitungNilaiAkhir();
36         double tugas = hitungNilai.nilaiTugas(tugas1, tugas2);
37
38         double na = hitungNilai.nilaiAkhir(tugas, mid, uas);
39
40         JOptionPane.showMessageDialog(null, "Nama: " + mhs.getNama() + "\nStambuk: " + mhs.getstambuk());
41         JOptionPane.showMessageDialog(null, "Tugas: " + tugas + "\nMid: " + mid + "\nUAS: " + uas + "\nNilai Akhir: " + na);
42     }
43 }
```