

TUGAS 4 PEMROGRAMAN BERORIENTASI OBJEK

EVALUASI PRAKTIKUM

Nama: Tizar Fitrah Anugerah

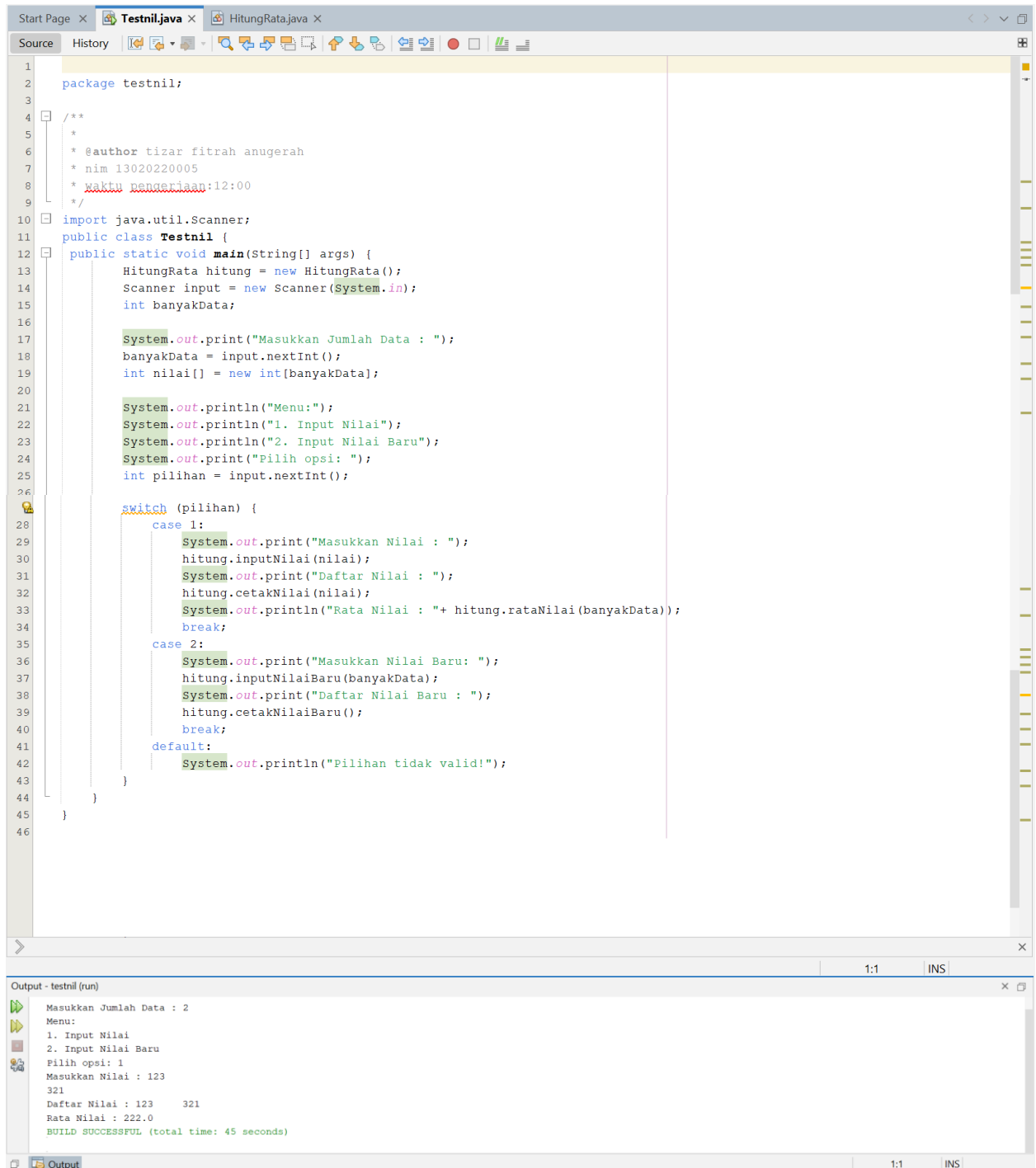
Kelas :A1

Nim :13020220005

1. Apakah perbedaan antara struktur kontrol percabangan if-else dan switch-case?
 - If-else adalah struktur kontrol percabangan yang memeriksa satu kondisi pada suatu waktu. Jika kondisi yang diperiksa adalah benar (true), blok kode dalam if statement dieksekusi. Jika tidak, blok kode dalam else statement (jika ada) akan dieksekusi. If-else cocok digunakan ketika terdapat beberapa kondisi yang tidak bisa diprediksi dengan pasti, atau ketika kondisi yang diperiksa tidak sesuai dengan struktur tertentu.
 - Switch-case adalah struktur kontrol percabangan yang memeriksa nilai dari ekspresi tunggal dan membandingkannya dengan serangkaian nilai yang mungkin. Jika nilai ekspresi cocok dengan salah satu nilai yang ditentukan, blok kode yang sesuai dengan nilai tersebut dieksekusi. Switch-case biasanya digunakan ketika terdapat banyak pilihan yang mungkin dan perlu dievaluasi berdasarkan nilai tunggal.
2. Kapan digunakan struktur kontrol if-else dan switch-case?
 - If-else cocok digunakan ketika kondisi yang ingin diperiksa tidak hanya berdasarkan nilai tunggal, tetapi juga mungkin melibatkan ekspresi yang lebih kompleks. beberapa kondisi harus dievaluasi secara berurutan, if-else lebih baik digunakan karena ia mengevaluasi kondisi satu per satu, berhenti pada kondisi yang benar.
 - Switch-case berguna ketika ingin memeriksa nilai dari satu ekspresi tunggal dan melakukan berbagai tindakan berdasarkan nilai tersebut. Switch-case efektif ketika nilai-nilai pilihan adalah diskrit dan teratur atau terbatas pada kumpulan nilai tertentu.

3. Pada program 2, tambahkan perintah untuk memilih 2 opsi menggunakan kontrol switch..case.
opsi pilihan 1=inputNilai()
Pilihan 2=inputNilaiBaru()

Kode:



```
1 package testnil;
2
3
4 /**
5  *
6  * @author tizar fitrah anugerah
7  * nim 13020220005
8  * Waktu Pembuatan:12:00
9  */
10 import java.util.Scanner;
11 public class Testnil {
12     public static void main(String[] args) {
13         HitungRata hitung = new HitungRata();
14         Scanner input = new Scanner(System.in);
15         int banyakData;
16
17         System.out.print("Masukkan Jumlah Data : ");
18         banyakData = input.nextInt();
19         int nilai[] = new int[banyakData];
20
21         System.out.println("Menu:");
22         System.out.println("1. Input Nilai");
23         System.out.println("2. Input Nilai Baru");
24         System.out.print("Pilih opsi: ");
25         int pilihan = input.nextInt();
26
27         switch (pilihan) {
28             case 1:
29                 System.out.print("Masukkan Nilai : ");
30                 hitung.inputNilai(nilai);
31                 System.out.print("Daftar Nilai : ");
32                 hitung.cetakNilai(nilai);
33                 System.out.println("Rata Nilai : " + hitung.rataNilai(banyakData));
34                 break;
35             case 2:
36                 System.out.print("Masukkan Nilai Baru : ");
37                 hitung.inputNilaiBaru(banyakData);
38                 System.out.print("Daftar Nilai Baru : ");
39                 hitung.cetakNilaiBaru();
40                 break;
41             default:
42                 System.out.println("Pilihan tidak valid!");
43         }
44     }
45 }
46
```

Output - testnil (run)

```
Masukkan Jumlah Data : 2
Menu:
1. Input Nilai
2. Input Nilai Baru
Pilih opsi: 1
Masukkan Nilai : 123
321
Daftar Nilai : 123    321
Rata Nilai : 222.0
BUILD SUCCESSFUL (total time: 45 seconds)
```

- HitungRata Class

```
Start Page X TestNil.java X HitungRata.java X
Source History
1 package testnil;
2 import java.util.*;
3 /**
4  *
5  * @author tizar
6  * 13020220005
7  * 13020220005 :12:30
8  */
9 public class HitungRata {
10     private double total = 0.0;
11     private ArrayList<Integer> nilaiBaru = new ArrayList<>();
12     private Scanner input = new Scanner(System.in);
13
14     public void inputNilai(int[] nilai) {
15         for (int i = 0; i < nilai.length; i++) {
16             nilai[i] = input.nextInt();
17             total += nilai[i];
18         }
19     }
20
21     public double rataNilai(int Ndata) {
22         return total / ((double) Ndata);
23     }
24
25     public void cetakNilai(int[] nilai) {
26         for (int angka : nilai) {
27             System.out.print(angka + "\t");
28         }
29         System.out.println();
30     }
31
32     public void inputNilaiBaru(int jumlah) {
33         while (jumlah > 0) {
34             nilaiBaru.add(input.nextInt());
35             jumlah--;
36         }
37     }
38
39     public void cetakNilaiBaru() {
40         ListIterator<Integer> iterator = nilaiBaru.listIterator();
41         while (iterator.hasNext()) {
42             Integer data = iterator.next();
43             if (data == null) {
44                 System.out.println("null");
45             } else {
46                 System.out.println(data.toString());
47             }
48         }
49     }
50 }
51
```

7:27 INS

4. Apakah perbedaan antara struktur kontrol perulangan while dan do-while?
- while: Struktur perulangan while mengevaluasi kondisi sebelum menjalankan blok kode di dalamnya. Jika kondisi terpenuhi (true), maka blok kode akan dieksekusi. Jika kondisi awalnya salah (false), maka blok kode tidak akan dieksekusi sama sekali.
 - do-while: Struktur perulangan do-while mengevaluasi kondisi setelah menjalankan blok kode di dalamnya. Artinya, blok kode akan dieksekusi sekali sebelum kondisi diuji. Jika kondisi terpenuhi (true) setelah eksekusi pertama, maka blok kode akan terus dieksekusi. Ini memastikan bahwa setidaknya satu iterasi akan dilakukan bahkan jika kondisi awalnya salah.
5. Kapan digunakan struktur kontrol for?
- Struktur kontrol percabangan digunakan dalam pemrograman untuk membuat kondisi atau syarat tertentu. Dua jenis syntax yang umum digunakan untuk mengimplementasikan struktur kontrol percabangan adalah If-else dan Switch-case. If-else digunakan untuk membuat kondisi sederhana, sedangkan Switch-case digunakan untuk membuat kondisi yang lebih kompleks. Struktur kontrol percabangan memungkinkan program untuk mengambil keputusan berdasarkan kondisi tertentu, seperti memvalidasi input pengguna, menjalankan kode berdasarkan pilihan pengguna, atau mengatur alur program berdasarkan kondisi tertentu
6. Apakah perbedaan antara Array dan ArrayList?berilah contoh masing-masing!
- Array adalah data structure yang menggunakan tipe data yang sama dan didefinisikan pada waktu pembuatan. Array dapat didefinisikan dengan tipe data yang sama, seperti int, float, dan string. Array tidak dapat menyimpan data yang berbeda tipe data. Array dapat didefinisikan dengan ukuran tertentu yang tidak dapat diubah setelah pembuatan. Contoh:

```
int[] arr = new int[5];
arr[0] = 1;
arr[1] = 2;
arr[2] = 3;
arr[3] = 4;
arr[4] = 5;
```

- ArrayList adalah data structure yang dapat menyimpan data berbeda tipe data. ArrayList dapat menyimpan data dengan tipe data yang berbeda, seperti int, float, dan string. ArrayList dapat menambah atau mengurangi ukuran dengan metode add() dan remove(). Contoh:

```
import java.util.ArrayList;
ArrayList<Integer> list = new ArrayList<>();
list.add(1);
list.add(2);
list.add(3);
list.add(4);
list.add(5);
```

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

Contoh:

```
1 package contohhash;
2 import java.util.HashMap;
3 import java.util.Map;
4 import java.util.Scanner;
5 /**
6  *
7  * @author Tizar Fitrah
8  * waktu pengerjaan = 14:00
9  */
10 public class ContoHash {
11     public static void main(String[] args) {
12         Scanner scanner = new Scanner(System.in);
13         Map<String, Integer> hashMap = new HashMap<>();
14
15         System.out.println("Masukkan jumlah data yang ingin dimasukkan:");
16         int jumlahData = scanner.nextInt();
17         scanner.nextLine(); // Membersihkan buffer
18
19         // Memasukkan nilai dan kunci (key) ke dalam HashMap
20         for (int i = 0; i < jumlahData; i++) {
21             System.out.println("Masukkan kunci (key) ke-" + (i + 1) + ":");
22             String key = scanner.nextLine();
23             System.out.println("Masukkan nilai ke-" + (i + 1) + ":");
24             int value = scanner.nextInt();
27             hashMap.put(key, value);
28         }
29
30         // Menampilkan isi HashMap
31         System.out.println("Isi HashMap:");
32         for (Map.Entry<String, Integer> entry : hashMap.entrySet()) {
33             System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());
34         }
35     }
36 }
```

Output:

```
run:
Masukkan jumlah data yang ingin dimasukkan:
3
Masukkan kunci (key) ke-1:
23
Masukkan nilai ke-1:
21
Masukkan kunci (key) ke-2:
32
Masukkan nilai ke-2:
21
Masukkan kunci (key) ke-3:
45
Masukkan nilai ke-3:
56
Isi HashMap:
Key: 23, Value: 21
Key: 45, Value: 56
Key: 32, Value: 21
BUILD SUCCESSFUL (total time: 16 seconds)
```

EVALUASI PRAKTIKUM 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)

Pewarisan adalah proses pembuatan subkelas dari kelas induk, yang memungkinkan subkelas mengambil semua karakteristik yang dimiliki oleh kelas induk dan menambahkan atribut atau metode yang lebih spesifik. Dalam kasus kelas Mahasiswa, ia merupakan subkelas dari kelas Orang, yang berarti Mahasiswa mewarisi semua atribut dan metode dari kelas Orang. Pada konstruktor tanpa parameter di kelas Mahasiswa, terdapat pemanggilan `super()` yang mengacu pada konstruktor kelas induk (Orang), sehingga konstruktor kelas induk akan dieksekusi terlebih dahulu sebelum konstruktor kelas Mahasiswa. Dengan kata lain, kelas Mahasiswa mengambil semua karakteristik yang dimiliki oleh kelas Orang dan menambahkan atribut tambahan.

Polimorfisme

adalah konsep yang memungkinkan objek dari kelas yang berbeda menggunakan metode dengan nama yang sama. Dalam kasus kelas Mahasiswa, terdapat dua versi dari konstruktor kelas Mahasiswa: satu tanpa parameter dan satu dengan dua parameter (nama dan stb). Konsep polimorfisme memungkinkan penggunaan metode dengan nama yang sama di kelas yang berbeda. Dalam hal ini, metode konstruktor Mahasiswa() tanpa parameter adalah contoh polimorfisme, di mana kelas Mahasiswa memiliki versi konstruktor yang berbeda dari kelas induknya (Orang). Meskipun kelas Mahasiswa mewarisi konstruktor tanpa parameter dari kelas Orang, kelas Mahasiswa juga memiliki versi konstruktor tambahan yang memungkinkan inisialisasi atribut nama dan stb sekaligus.

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

• Utama

```
1 package modul4;
2
3 /**
4  *
5  * @author tizar fitrah
6  */
7 public class Modul4 {
8     public static void main(String[] args) {
9         Orang orang = new Orang();
10        orang.nama = "Tizar";
11        System.out.println("Stb : " + orang.nama);
12
13        Orang.info();
14
15        Mahasiswa mahasiswa = new Mahasiswa();
16        mahasiswa.nama = "Fitrah";
17        System.out.println("Stb Mahasiswa : " + mahasiswa.nama);
18
19        Mahasiswa.info();
20    }
21 }
22
23
```

- Orang

```

1 package modul4;
2
3 /**
4  *
5  * @author tizar fitrah anugerah
6  */
7
8 public class Orang {
9     public String nama;
10
11     public Orang() {
12         this.nama = "Aminah";
13     }
14
15     public Orang(String nama) {
16         this.nama = nama;
17     }
18
19     public static void info() {
20         System.out.println("Ini adalah class Orang");
21     }
22 }

```

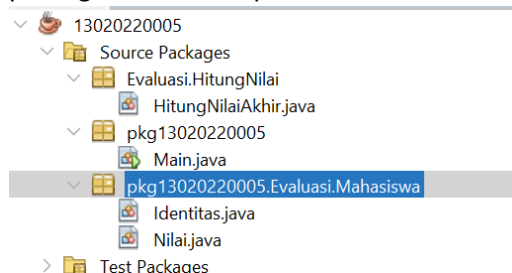
- Mahasiswa

```

7 /**
8  *
9  * @author Tizar fitrah
10 */
11
12 public class Mahasiswa extends Orang {
13     private String stb;
14
15     public Mahasiswa() {
16         super();
17         this.stb = "13020220005";
18     }
19
20     public Mahasiswa(String stb, String nama) {
21         this.nama = nama;
22         this.stb = stb;
23     }
24
25     public static void info() {
26         System.out.println("Ini adalah class Mahasiswa");
27     }
28 }

```

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



Setelah Itu Lengkapi Program pada soal yang telah diberikan

- Identitas

```
1 package pkg13020220005.Evaluasi.Mahasiswa;
2
3 /**
4  *
5  * @author tizar fitrah
6  */
7 public class Identitas {
8     public String nama;
9     public String stambuk;
10
11     public void setNama(String nama) {
12         this.nama = nama;
13     }
14
15     public String getNama() {
16         return nama;
17     }
18
19     public void setStambuk(String stambuk) {
20         this.stambuk = stambuk;
21     }
22
23     public String getStambuk() {
24         return stambuk;
25     }
26 }
27
```

- Nilai

```
1 package pkg13020220005.Evaluasi.Mahasiswa;
2
3 /**
4  *
5  * @author tizar fitrah
6  */
7 public class Nilai {
8     private int tugas1;
9     private int tugas2;
10    private int mid;
11    private int uas;
12
13
14    public void setTugas1(int tugas1) {
15        this.tugas1 = tugas1;
16    }
17
18    public int getTugas1() {
19        return tugas1;
20    }
21
22    public void setTugas2(int tugas2) {
23        this.tugas2 = tugas2;
24    }
25
26    public int getTugas2() {
27        return tugas2;
28    }
29
30    public void setMid(int mid) {
31        this.mid = mid;
32    }
33
34    public int getMid() {
35        return mid;
36    }
37
38
39    public void setUas(int uas) {
40        this.uas = uas;
41    }
42
43    public int getUas() {
44        return uas;
45    }
46 }
47
```


- Hitung nilai akhir

```

1 package Evaluasi.HitungNilai;
2
3 /**
4  *
5  * @author tizar fitrah
6  */
7
8 public class HitungNilaiAkhir {
9     public double nilaiTugas(int tugas1, int tugas2) {
10         double rataTugas = (tugas1 + tugas2) / 2.0;
11         return rataTugas;
12     }
13
14     public double nilaiAkhir(double tugas, int mid, int uas) {
15         double nilaiAkhir = (tugas * 0.4) + (mid * 0.3) + (uas * 0.3);
16         return nilaiAkhir;
17     }
18 }

```

- Main

```

1 package pkg13020220005;
2 import Evaluasi.HitungNilai.HitungNilaiAkhir;
3 import pkg13020220005.Evaluasi.Mahasiswa.*;
4
5 import javax.swing.JOptionPane;
6
7 /**
8  *
9  * @author tizar fitrah
10  */
11
12 public class Main {
13
14     public static void main(String[] args) {
15         String nama = JOptionPane.showInputDialog("Masukkan Nama:");
16         String stambuk = JOptionPane.showInputDialog("Masukkan Stambuk:");
17
18         int tugas1 = Integer.parseInt(JOptionPane.showInputDialog("Masukkan Tugas 1:"));
19         int tugas2 = Integer.parseInt(JOptionPane.showInputDialog("Masukkan Tugas 2:"));
20         int mid = Integer.parseInt(JOptionPane.showInputDialog("Masukkan Nilai Mid:"));
21         int uas = Integer.parseInt(JOptionPane.showInputDialog("Masukkan Nilai UAS:"));
22
23         Identitas mhs = new Identitas();
24         mhs.setNama(nama);
25         mhs.setStambuk(stambuk);
26
27         HitungNilaiAkhir hitungNilai = new HitungNilaiAkhir();
28         double tugas = hitungNilai.nilaiTugas(tugas1, tugas2);
29
30         double na = hitungNilai.nilaiAkhir(tugas, mid, uas);
31
32         JOptionPane.showMessageDialog(null, "Nama: " + mhs.getNama() + "\nStambuk: " + mhs.getStambuk());
33         JOptionPane.showMessageDialog(null, "Tugas: " + tugas + "\nMid: " + mid + "\nUAS: " + uas + "\nNilai Akhir: " + na);
34     }
35 }

```

- Output

Input

Masukkan Nilai Mid:

87

OK Cancel

Input

Masukkan Nilai UAS:

92

OK Cancel

Message

Nama: Tizar Fitrah
Stambuk: 13020220005

OK

Message

Tugas: 90.5
Mid: 87
UAS: 92
Nilai Akhir: 89.89999999999999

OK