LAPORAN RESMI

PRAKTIKUM 12

PRAKTIKUM PEMROGRAMAN BERBASIS OBYEK



Nama : Muhammad Zaid A

Kelas : 2 D4 Teknik Informatika A

NRP : 2110191013

PRAKTIKUM 12 – OVERRIDING DAN OVERLOADING

**PENDAHULUAN**

1. Memahami tentang overloading

Overloading adalah suatu keadaan dimana beberapa method sekaligus dapat mempunyai nama yang sama, akan tetapi mempunyai fungsionalitas yang berbeda.

1. Memahami tentang overriding

Overriding adalah suatu keadaan dimana method pada subclass menolak method pada parent classnya.

1. Memahami aturan tentang overridden

Method yang terkena override (overriden method) diharuskan tidak boleh mempunyai modifier yang lebih luas aksesnya dari method yang meng-override (overriding method).

**PERCOBAAN**

Percobaan 1 – Melakukan overloading pada method

1. Listing Program

import java.awt.Point;

public class Segiempat {

int x1 = 0;

int y1 = 0;

int x2 = 0;

int y2 = 0;

public void buatSegiempat(int x1, int y1, int x2, int y2){

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

}

public void buatSegiempat(Point topleft, Point bottomright){

x1 = topleft.x;

y1 = topleft.y;

x2 = bottomright.x;

y2 = bottomright.y;

}

void buatSegiempat(Point topleft, int w, int h){

x1 = topleft.x;

y1 = topleft.y;

x2 = (x1 + w);

y2 = (y1 + h);

}

void cetakSegiempat(){

System.out.println("Segiempat: <" + x1 + ", " + y1);

System.out.println(", " + x2 + ", " + y2 + ">");

}

public static void main(String[] args){

Segiempat rect = new Segiempat();

System.out.println("Buat segiempat dengan koordinat(25,25) dan (50,50)");

rect.buatSegiempat(25,25,50,50);

rect.cetakSegiempat();

System.out.println();

System.out.println("Buat segiempat dengan point (10,10)dan point (20,20)");

rect.buatSegiempat(new Point(10,10), new Point(20,20));

rect.cetakSegiempat();

System.out.println();

System.out.println("Buat segiempat dengan 1 point (10,10),koordinat (50,50)");

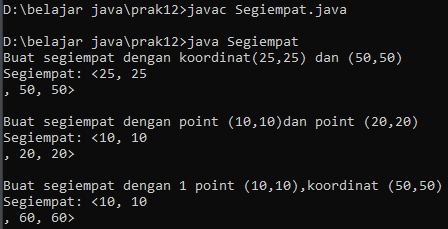
rect.buatSegiempat(new Point(10,10), 50, 50);

rect.cetakSegiempat();

}

}

1. Output



1. Analisis

**LATIHAN SOAL**

Latihan 1 - Overriding

Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !

1. Listing Program

class Base {

private void amethod(int iBase) {

System.out.println("Base.amethod");

}

}

class Over extends Base {

public static void main(String argv[]) {

Over o = new Over();

int iBase=0;

o.amethod(iBase);

}

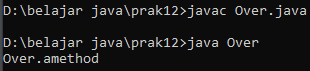
public void amethod(int iOver) {

System.out.println("Over.amethod");

}

}

1. Output



1. Analisis

Latihan 2 - Overloading

Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !

1. Listing Program

class MyParent {

int x, y;

MyParent(int x, int y) {

this.x = x;

this.y = y;

}

public int addMe(int x, int y) {

return this.x + x + y + this.y;

}

public int addMe(MyParent myPar) {

return addMe(myPar.x, myPar.y);

}

}

class MyChild extends MyParent {

int z;

MyChild (int x, int y, int z) {

super(x,y);

this.z = z;

}

public int addMe(int x, int y, int z) {

return this.x + x + this.y + y + this.z + z;

}

public int addMe(MyChild myChi) {

return addMe(myChi.x, myChi.y, myChi.z);

}

public int addMe(int x, int y) {

return this.x + x + this.y + y;

}

}

public class MySomeOne {

public static void main(String args[]) {

MyChild myChi = new MyChild(10, 20, 30);

MyParent myPar = new MyParent(10, 20);

int x = myChi.addMe(10, 20, 30);

int y = myChi.addMe(myChi);

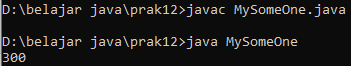
int z = myPar.addMe(myPar);

System.out.println(x + y + z);

}

}

1. Output



1. Analisis

Latihan 3 - Overloading

Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !

1. Listing Program

class MyClass {

void myMethod(int i) {

System.out.println("int version");

}

void myMethod(String s) {

System.out.println("String version");

}

public static void main(String args[]){

MyClass obj = new MyClass();

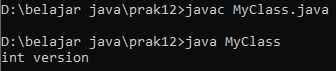
char ch = 'c';

obj.myMethod(ch);

}

}

1. Output



1. Analisis

Latihan 4 - Mengimplementasikan UML class diagram dalam program

1. Listing Program

class Orang {

protected String nama;

protected int umur;

public Orang (String nama){

this.nama=nama;

}

public Orang(String nama,int umur){

this.nama=nama;

this.umur=umur;

}

}

class Orang{

protected String nama;

protected int umur;

public Orang(String nama){

this.nama=nama;

}

public Orang (String nama,int umur){

this.nama=nama;

this.umur=umur;

}

}

class Dosen extends Orang{

private int nip;

public Dosen(String nama){

super(nama);

}

public Dosen(String nama, int nip){

super(nama);

this.nip = nip;

}

public Dosen(String nama, int nip, int umur){

super(nama,umur);

this.nip = nip;

}

public void Info(){

System.out.println("Nama : " + super.nama);

System.out.println("NIP : " + nip);

System.out.println("Umur : " + super.umur);

}

}

public class TesLatihan{

public static void main(String[] args){

System.out.println("Memasukkan identitas dosen 1 :Agus");

Dosen dosen1 = new Dosen("Agus");

System.out.println("Memasukkan identitas dosen 2 : Budi,NIP. 1458");

Dosen dosen2 = new Dosen("Budi", 1458);

System.out.println("Memasukkan identitas dosen 3 : Iwan,NIP. 1215, umur 47");

Dosen dosen3 = new Dosen("Iwan", 1215, 47);

System.out.println();

dosen1.Info();

System.out.println();

dosen2.Info();

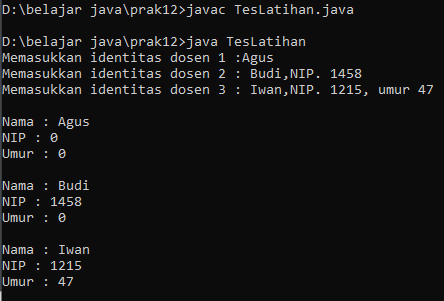
System.out.println();

dosen3.Info();

}

}

1. Output



1. Analisis

**TUGAS**

Tugas 1 - Mengimplementasikan UML class diagram dalam program

1. Listing Program

class RerataNilai{

public int average(int a1, int a2){

return (a1+a2)/2;

}

public double average(double b1, double b2){

return (b1+b2)/2.0;

}

public int average(int c1, int c2, int c3){

return (c1+c2+c3)/3;

}

}

public class TesTugas{

public static void main(String[] args){

RerataNilai rn = new RerataNilai();

System.out.println("Rerata nilai 21 dan 13 adalah : " +rn.average(21,13));

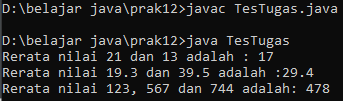
System.out.println("Rerata nilai 19.3 dan 39.5 adalah :" + rn.average(19.3,39.5));

System.out.println("Rerata nilai 123, 567 dan 744 adalah: " + rn.average(123,567,744));

}

}

1. Output



1. Analisis

Tugas 2 - Mengimplementasikan UML class diagram dalam program

1. Listing Program

class Katak{

protected int umur;

protected String nama;

public Katak(int umur, String nama){

this.umur = umur; this.nama = nama;

}

public String caraBergerak(){

if (umur > 2)

return ("Melompat");

else return ("Berenang");

}

}

class Kecebong extends Katak{

protected double panjangEkor;

public Kecebong(int umur, String nama, double panjangEkor){

super(umur,nama);

this.panjangEkor = panjangEkor;

}

}

public class TesTugas2{

public static void main(String[] args){

Katak O1 = new Katak(5,"Froggy");

Kecebong O2 = new Kecebong(2,"Junior Frog",10.0);

System.out.println("Obyek\tumur\tnama\t\tpanjangEkor\tcaraBergerak ");

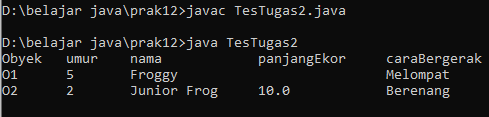
System.out.println("O1\t" + O1.umur + "\t" + O1.nama + "\t\t\t\t" + O1.caraBergerak());

System.out.println("O2\t" + O2.umur + "\t" + O2.nama + "\t" + O2.panjangEkor + "\t\t" + O2.caraBergerak());

}

}

1. Output



1. Analisis