LAPORAN PRAKTIKUM

PEMROGRAMAN 1

Nama : Mochammad Fahrudin

Npm : 4513210024

1. Pilihan penjelasan definisi dan contoh :
2. Overloading
3. Overriding

Gunakan “Case”.

Jawaban :

1. Overloading adalah **method dengan nama yang sama** dengan method lain pada suatu *class*tetapi dengan **parameter yang berbeda**. Tujuan dibuatnya overloading yaitu memudahkan penggunaan method dengan fungsi yang hampir sama.

Contoh program :

Over.java

import java.util.Scanner;

public class over

{

public static void main(String[] args)

{

int Jumlah;

int Nilai1;

int Nilai2;

Scanner scan = new Scanner(System.in);

//menampilkan menu PILIHAN

System.out.println("MENU PILIHAN : ");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("1. Overloading");

System.out.println("2. Overriding");

System.out.println("");

System.out.print("Masukkan Nomor PILIHAN : ");

int Menu = scan.nextInt();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

switch(Menu){

case 1 :

System.out.println("Menu yang Anda Pilih Overloading ");

System.out.println("Overloading adalah method dengan nama yang sama dengan method lain pada suatu class tetapi dengan parameter yang berbeda. ");

System.out.println (" class Pertambahan{ ");

System.out.println ("public void tambah1(){");

System.out.println ("int a=10, b=35;");

System.out.println(" System.out.println(''" + "Hasil Pertambahann dari metod tambah1 ke-1 = ''" + "+(a+b));");

System.out.println ("}");

System.out.println ("//Metod tambah1 di overloading dengan 2 parameter (int x, int y)");

System.out.println ("public void tambah1(int x, int y){");

System.out.println(" System.out.println(''" + "Hasil Pertambahann dari metod tambah1 ke-2 = ''" + "+(x+y));");

System.out.println ("}");

System.out.println ("public static void main(String [] args){");

System.out.println ("Pertambahan pp;");

System.out.println ("pp = new Pertambahan();");

System.out.println ("pp.tambah1();//memanggil metod tambah1 ke-1");

System.out.println ("pp.tambah1(5,5);//memanggil metod tambah1 ke-2");

System.out.println("}");

System.out.println ("}");

break;

case 2:

System.out.println("Menu yang Anda Pilih Overriding ");

System.out.println("Definisi Overridding : ");

System.out.println("Overriding memiliki method yang namanya sama namun berbeda class. Method pertama adalah method yang berasal dari class induk dengan parameter yang sama dengan method pada class anak. Pada Overriding nama method dan parameternya pada kedua classnya harus sama.");

System.out.println("");

System.out.println("Contoh program Overridding : ");

System.out.println("class Nama { ");

System.out.println("public void Sapa() { ");

System.out.println("System.out.println(''" + "Mochammad Fahrudin''" + ");");

System.out.println(" } ");

System.out.println("public static void main(String [] args) { ");

System.out.println("Siapa ss;");

System.out.println("Nama nn;");

System.out.println("ss = new Siapa();");

System.out.println("nn = new Nama();");

System.out.println("");

System.out.println("ss.Sapa(); //memanggil method Sapa() pada class Siapa");

System.out.println("nn.Sapa(); //memanggil method Sapa() pada class Nama");

System.out.println("}");

System.out.println("}");

System.out.println("class Siapa extends Nama { ");

System.out.println("//Method Sapa() pada class Nama di override");

System.out.println("public void Sapa() {");

System.out.println("System.out.println(''" + "Puri Rahmatika''" + ");");

System.out.println("}");

System.out.println("}");

break;

default: System.out.print("Menu yang anda pilih tidak tersedia !!");

}

}

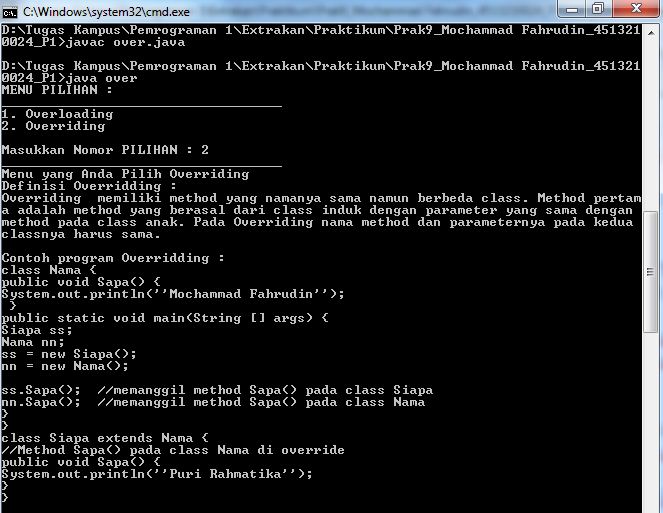
}

**Screenshot**



1. Overriding adalah method *subclass* sama dengan method *super class*, parameternya sama tetapi pernyataan atau implementasinya berbeda.

Contoh program :



1. Contoh program yang ada dimodul
2. Mahasiswa

//perintah Class & Object - Mahasiswa

// package Latihan01.sesi1.bin

import java.util.\*;

public class Mahasiswa{

String nim;

String nama;

static Integer jumlah;

//kalau tidak ada satupun constructor yang didefinisikan , maka akan dibuatkan

//by default, Java akan membuatkan constructor

//yang tidak menerima argument/tanpa parameter

//dan tidak melakukan apa-apa

//public Mahasiswa() //default Mahasiswa

}

1. main.java

//perintah Class & Object - DemoClassObject1

import java.util.\*;

public class Main

{

public static void main(String[]args)

{

Mahasiswa.jumlah = 0;

//Mahasiswa.nim = "999";

//error, instance variabel harus ada objectnya terlebih dahulu

System.out.println("Jumlah Mahasiswa : "+Mahasiswa.jumlah);

Mahasiswa m1 = new Mahasiswa();

m1.nim = "123";

m1.nama = "Mochammad ";

m1.jumlah = 1;

Mahasiswa m2 = new Mahasiswa ();

m2.nim = "456";

m2.nama = "Fahrudin";

m2.jumlah = 2;

tampilkanMahasiswa(m1);

tampilkanMahasiswa(m2);

System.out.println("Jumlah Mahasiswa : "+Mahasiswa.jumlah);

}

public static void tampilkanMahasiswa(Mahasiswa m)

{

System.out.println("NIM : "+m.nim);

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

System.out.println("Jumlah : "+m.jumlah);

System.out.println("");

System.out.println("");

}

}

1. Dosen.java

//perintah Class & Object - Dosen

import java.util.\*;

public class dosen

{

String nama;

String email;

public dosen (String nama, String email)

{

this.nama = nama;

this.email = email;

}

public dosen () {}

public void info()

{

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

System.out.println("Email : "+email);

System.out.println("");

System.out.println("");

}

}

1. DemoOverLoading.java

import java.util.\*;

public class DemoOverloading

{

public static void main (String [] args)

{

dosen d = new dosen();

d.info();

d.nama = "Chandra";

d.email = "Ichan@gmail.com";

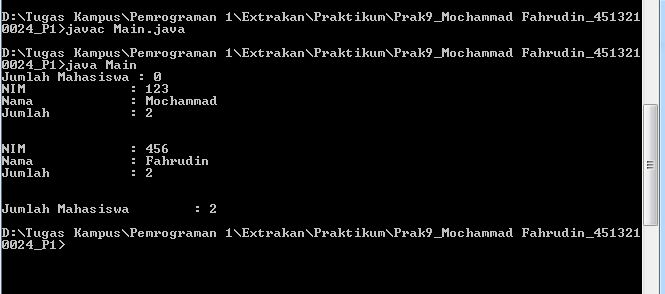
d.info();

}

}

**Screenshot**

Main.java



DemoOverloading.java

