LAPORAN RESMI

PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK (PBO)

MODUL 5

"INTERFACE"

Disusun Oleh :

TGL PRAKTIKUM	:	3 Desember 2012
NAMA	:	ACHMAD FAJAR NORULLAH
NRP	:	11.04.111.000.75
KELAS/KELOMPOK	:	B/3
DOSEN PENGAMPU	:	ERY SETIYAWAN JULLEV, S.KOM.
ASISTEN	:	SHOHIB

Disetujui : ...../...../...../Bangkalan
(SHOHIB) 10.04.111.00101

LABORATORIUM COMMON COMPUTING
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA
2012 / 2013

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada modul ini kita akan mempelajari tentang interface. Yang mana Interface merupakan sebuah method yang tidak memiliki body.

1.2 Tujuan

Tujuan mahasiswa mengikuti praktikum PBO Modul 4 ini adalah :

1. Mahasiswa dapat memahami tentang penggunaan Interface.
2. Mahasiswa dapat menggunakan Interface dalam suatu permasalahan.

BAB II

DASAR TEORI

2.1 Interface

Produktifitas dalam pemrograman dapat ditingkatkan antara lain dengan cara memaksimalkan kode yang dapat dimanfaatkan berulang kali (reusable code). Interface adalah struktur kode yang diawali kata kunci interface dan berisi bagian reusable code berupa abstract method atau konstanta. Semua method dalam interface secara otomatis memiliki visibilitas public.

Abstract method dalam interface dapat diimplementasikan dalam class dengan menggunakan kata kunci implements. Dalam java, satu class hanya dapat menjadi turunan dari satu class namun diperkenankan untuk dapat mengimplementasikan lebih dari satu interface.

Instance atau objek dapat dibentuk dengan memasangkan variabel bertipe class atau interface dengan referensi berupa pemanggilan constructor class. Casting dapat dilakukan untuk mengubah objek dari tipe superclass menjadi tipe subclass.

BAB III

TUGAS PENDAHULUAN

Pertanyaan :

1. Apa yang dimaksud dengan abstract method?
2. Apa yang dimaksud dengan overriding method?
3. Apa yang dimaksud dengan instansiasi?

Jawaban

1. Method yang tidak memiliki body yang biasanya diimplementasikan pada interface.

2. Overriding method adalah keadaan dimana terdapat 2 method dengan nama dan parameter yang sama sehingga akan terjadi override.

3. Intansiasi atau objek dapat dibentuk dengan memasangkan variabel bertipe class atau interface dengan referensi berupa pemanggilan constructor class. Casting dapat dilakukan untuk mengubah objek dari tipe superclass menjadi tipe subclass.

BAB IV

IMPLEMENTASI

4.1 Kegiatan Praktikum

Pada bagian kegiatan praktikum ini, kita akan belajar membuat interface dan mengimplementasikannya dalam sebuah class.

```
/*interface berisi abstract method atau method header*/
interface MyComparable {
boolean greaterThan(Object obj); boolean lessThan(Object obj);
boolean equal(Object obj);
}
/*interface berisi konstanta*/
interface Constants {
int min = 1000;
int max = 9999;
}
/*class mengimplementasikan dua interface*/
class FourDigitsNumber implements Constants, MyComparable {
private int value;
public FourDigitsNumber(int initValue) {
/*latihan 1: atur agar nilai yang diinputkan dalam constructor hanya
berada di antara min - max
*/
}
/*latihan 2: tambahkan method get untuk mengakses value*/
/*overriding method greaterThan*/
public boolean greaterThan(Object obj) {
/*casting from superclass to subclass*/ FourDigitsNumber number =
(FourDigitsNumber)obj; return ( value > number.getValue() );
}
/*latihan 3: lengkapi overriding method interface*/
}
/*Contoh penggunaan class*/
class ExInterface {
public static void main(String [] args) { FourDigitsNumber number1 =
new FourDigitsNumber(700); FourDigitsNumber number2 = new
FourDigitsNumber(1700); FourDigitsNumber number3 = new
FourDigitsNumber(70000); System.out.println(number1.getValue());
System.out.println(number2.getValue());
System.out.println(number3.getValue());
System.out.println(number1.greaterThan(number2));
System.out.println(number1.lessThan(number2));
System.out.println(number1.equal(number2));
}}
```

BAB V

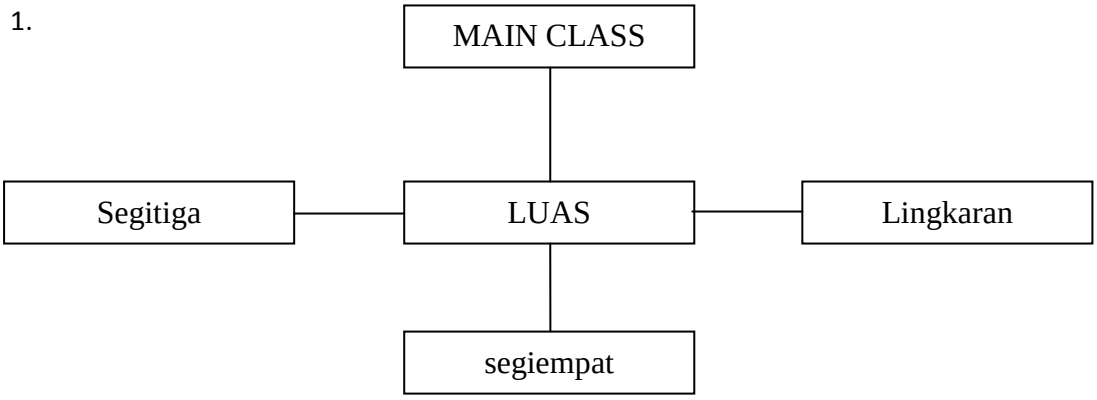
TUGAS AKHIR

5.1 Tugas Akhir

1. Segitiga, Lingkaran dan Segiempat adalah bentuk geometri yang dapat dihitung luasnya. Gambarkan diagram hirarki class untuk kasus ini. Kemudian tulis kode program untuk class dan buat program sederhana untuk keperluan pengujian: atur array dengan elemen berisi referensi ke setiap bentuk geometri di atas, masukkan data ke dalamnya dan gunakan untuk memanggil method guna menghitung luas dari setiap bentuk geometri tersebut.
2. Buatlah sebuah interface dengan nama MtkDasar yang berisi dua buah method head atau abstract
   String tambah(int a, int b);
   String kurang(int a, int b);
   Buatlah sebuah class dengan nama Pecahan yang terdiri dari: dua variabel bertipe int yaitu pembilang dan penyebut, sebuah constructor dengan dua parameter bertipe int, method set( ) / get( ), method toString( ). Pecahan harus mengimplementasikan interface MtkDasar.
   Buatlah sebuah class CobaPecahan yang berfungsi menguji class Pecahan.

   Jawaban:

1.

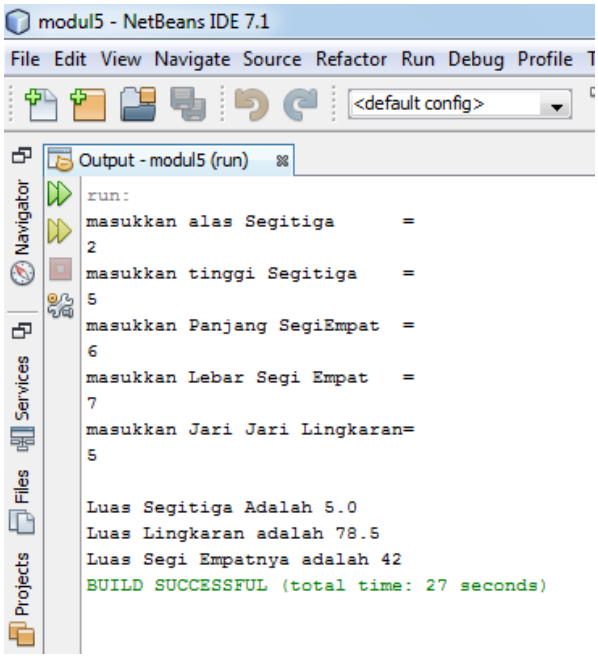



```
public interface Luas {
 double pi=3.14;
 double k=0.5;
 public void Lingkaran();
 public void Segitiga();
 public void Segiempat();

public class LuasSegitiga implements Luas{
double luas;
int a,t;

    @Override
    public void Lingkaran() {
```

```
        throw new UnsupportedOperationException("Not supported yet.");
    }

    @Override
    public void Segitiga() {
        luas=a*t*k;
        System.out.println("Luas Segitiga Adalah "+luas);
    }

    @Override
    public void Segiempat() {
        throw new UnsupportedOperationException("Not supported yet.");
    }

}

public class LuasLingkaran implements Luas{
double luas;
int r;
    @Override
    public void Lingkaran() {
      luas=pi*r*r;
        System.out.println("Luas Lingkaran adalah "+luas);
    }

    @Override
    public void Segitiga() {

    }

    @Override
    public void Segiempat() {

    }

}

public class LuasSegiempat implements Luas{
    int luas,p,l;
    @Override
    public void Lingkaran() {
    }

    @Override
    public void Segitiga() {
    }

    @Override
    public void Segiempat() {
        luas=p*l;
        System.out.println("Luas Segi Empatnya adalah "+luas);
    }
```

```
}
package TA1;

import java.util.Scanner;

/**
 *
 * @author Fajarranz
 */
public class main {
    public static void main(String[] args) {
        int a,t,p,l,r;
        Scanner input = new Scanner(System.in);
        System.out.println("masukkan alas Segitiga      = ");
        a=input.nextInt();
        System.out.println("masukkan tinggi Segitiga    = ");
        t=input.nextInt();
        System.out.println("masukkan Panjang SegiEmpat  = ");
        p=input.nextInt();
        System.out.println("masukkan Lebar Segi Empat   = ");
        l=input.nextInt();
        System.out.println("masukkan Jari Jari Lingkaran= ");
        r=input.nextInt();
        System.out.println("");

        LuasSegitiga x= new LuasSegitiga();
        x.a=a;
        x.t=t;
        x.Segitiga();
        LuasLingkaran y= new LuasLingkaran();
        y.r=r;
        y.Lingkaran();
        LuasSegiempat z= new LuasSegiempat();
        z.p=p;
        z.l=l;
        z.Segiempat();


    }

}
```

Run ning program:



2.

```
public interface MtkDasar {
 String tambah(int a, int b);
 String kurang(int a, int b);
}

public class Pecahan implements MtkDasar{
    int pembilang,penyebut;

    public void setpembilang(int a){
    this.pembilang=a;
    }
    public void setpenyebut(int b){
    this.penyebut=b;
    }
    public int getpembilang(){
    return pembilang;
    }
    public int getpenyebut(){
    return penyebut;
    }

    @Override
    public String toString(){
    return pembilang+"/"+penyebut;
    }
    @Override
    public String tambah(int a, int b) {
```

```
     return String.valueOf(a+b);
    }

    @Override
    public String kurang(int a, int b) {
    return String.valueOf(a-b);
    }
}

public class CobaPecahan {
    public static void main(String[] args) {
        Pecahan x= new Pecahan();
        System.out.println(x+"+1/3");
        x.tambah(1,6);
        System.out.println(x.toString());
        System.out.print(x+" + 1/3 = ");
        x.kurang(1, 6);
        System.out.println(x.toString());

    }
}
```

BAB VI
PENUTUP

6.1 Kesimpulan

Interface merupakan bab penting karena ini dikaitkan dengan abstract method dan overriding method.

6.2 Saran

- Diharap asisten untuk menjelaskan mengenai tugas praktikum dan kejelasan tentang tugas akhir supaya jelas dan dapat dimengerti.