

## **LAPORAN 7**

### **Pemrograman Berioorientasi Objek**

**“AntarMuka dan Implementasi”**



**Nama : Panji Akbar**  
**NIM : 60200112059**  
**Kelas : F**

**LABORATORIUM KOMPUTER TERPADU**  
**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS SAINS & TEKNOLOGI**  
**UNIVERSITAS ISLAM NEGERI ALAUDDIN**  
**MAKASSAR**  
**2015**

## LEMBAR PENILAIAN PRAKTIKUM

Nama : Panji Akbar .....  
NIM : 60200112059 .....  
Praktikum : Pemrograman berorientasi objek...  
Judul praktikum : Antarmuka Dan Implementasi.....

No	Komponen penilaian	Nilai
I.	Nilai Kehadiran (100)	
II.	Nilai Pelaksanaan praktikum (100)	
	A. Kedisiplinan (25)	
	B. Keterampilan menggunakan Komputer/software/jaringan (40)	
	C. Keaktifan (25)	
	D. Dan lain ... lain (10)	
III.	Nilai laporan praktikum (100)	
TOTAL (300)		

Tanggal praktikum : .....  
Dikumpulkan tanggal : .....  
Dikoreksi tanggal : .....

Asisten Praktikum,

.....

## LEMBAR PENGESAHAN/ACC PRAKTIKUM

Laporan praktikum pemrograman berorientasi objek dengan judul AntarMuka Dan Implementasi yang disusun oleh :

Nim : 60200112059

Nama : Panji Akbar

Kelas : C

Kelompok :

Telah diperiksa dan dikonsultasikan kepada Asisten I dan Asisten II  
maka dinyatakan diterima

Makassar, .....

Asisten I

Asisten II

Gunawan, S.Kom

Mudassir

Mengetahui,  
Dosen Penanggung Jawab

Nur Afif , S.T., M.T

## 1. TujuanPraktikum

- Mengetahui Penggunaan kelas Abstrak
- Mengetahui penggunaan interface
- Mengetahui penggunaan innerclass dan klas anonymous

## 2. DasarTeori

**Abstract class** adalah kelas yang terletak pada posisi tertinggi pada hierarki kelas. Kelas ini digunakan sebagai basis bagi penurunan kelas lainnya, sehingga abstract class tidak dapat diinstansiasi secara langsung menjadi object. Suatu abstract class bisa mengandung method kongkrit dan/atau method abstract. Abstract method merupakan suatu method yang tidak mempunyai badan method. Hanya berupa nama method dan parameter inputan method. Ketika suatu abstract class diturunkan/ di-**extends**, maka subclass harus mengimplementasikan semua method abstract pada parent/ base class. Jika tidak mengimplementasikan semua method abstract maka subclass harus dideklarasikan sebagai abstract class.

**Interface** adalah kumpulan method yang hanya memuat deklarasi dan struktur method, tanpa detail implementasinya. Sedangkan detail dari method berada pada class yang mengimplementasikan interface tersebut. Interface digunakan bila Anda ingin mengaplikasikan suatu method yang spesifik, yang tidak diperoleh dari proses inheritance. Tipe data yang boleh pada interface hanya tipe data **konstan**. Setelah mengetahui apa itu interface.

## Abstract class

```
package praktikum_7;

public abstract class Model {

    protected int a;
    protected int b;

    public void geser(int a, int b) {
        this.a = a;
        this.b = b;
        System.out.println("a = " + a + "b = " + b);
    }

    public abstract float hitungLuas();
    public abstract void tulis();

}

package praktikum_7;

public class BujurSangkar extends Model {

    private float sisi;

    public BujurSangkar() {
        sisi = 5;
    }

    public float hitungLuas() {
        return sisi*sisi;
    }

    public void tulis() {
        System.out.println("Bujur Sangkar");
    }

}
```

```
package praktikum_7;

public abstract class Segitiga extends Model {

    protected float alas;
    protected float tinggi;

    public Segitiga() {
        alas = 5;
        tinggi = 5;
    }

    public float hitungLuas() {
        return (alas * tinggi/2);
    }

}
```

```
package praktikum_7;

public class SegitigaSamaKaki extends Segitiga {

    public void tulis () {
        System.out.println("Segitiga sama kaki");
    }

}
```

```
package praktikum_7;

public class SegitigaSamaSisi extends Segitiga {

    public void tulis () {
        System.out.println("Segitiga sama sisi");
    }

}
```

```
CobaBentuk.java
1 package praktikum_7;
2
3 import praktikum_7.Model;
4
5 public class CobaBentuk {
6
7     public static void cetakBangun(Model b) {
8
9         System.out.println(b.hitungLuas());
10        b.tulis();
11        b.geser(5, 5);
12    }
13
14    public static void main(String args[]) {
15
16        cetakBangun(new BujurSangkar());
17        cetakBangun(new SegitigaSamaKaki());
18        cetakBangun(new SegitigaSamaSisi());
19    }
20 }
21
22
23
```

Problems @ Javadoc Declaration Search Console

<terminated> CobaBentuk (1) [Java Application] C:\Program Files\Java\jre1.8.0\_25\bin\javaw.exe (Jun 28, 2015, 7:48:51 AM)

25.0  
Bujur Sangkar  
a = 5b = 5  
12.5  
Segitiga sama kaki  
a = 5b = 5  
12.5  
Segitiga sama sisi  
a = 5b = 5

## Interface

```
package praktikum_7.Interface;

public interface BentukKeliling {

    public static final double PHI = 3.14;
    public double hitungKeliling();

}

package praktikum_7.Interface;

public interface BentukLuas {

    public static final double PHI = 3.14;
    public double hitungLuas();

}
```

```
BentukKeliling.java  BentukLuas.java  Lingkaran.java
laporan praktikum/src/praktikum_7/Interface/BentukKeliling.java
2
3 public class Lingkaran implements BentukLuas, BentukKeliling{
4
5     private double jariJari;
6
7     public Lingkaran(double jariJari) {
8         this.jariJari = jariJari;
9     }
10
11     public double hitungKeliling()
12     {
13         return 2*BentukKeliling.PHI * jariJari;
14     }
15
16     public double hitungLuas()
17     {
18         return 2*BentukLuas.PHI * Math.pow(jariJari, 2);
19     }
20 }
21
```

Problems @ Javadoc Declaration Search Console

<terminated> CobaBentuk (1) [Java Application] C:\Program Files\Java\jre1.8.0\_25\bin\javaw.exe (Jun 28, 2015, 7:51:38 AM)

25.0

Bujur Sangkar

a = 5b = 5

12.5

Segitiga sama kaki

a = 5b = 5

12.5

Segitiga sama sisi

a = 5b = 5



BentukKeliling.java BentukLuas.java Lingkaran.java \*Trapesium.java

```
1 package praktikum_7.Interface.Extend;
2
3 public class Trapesium {
4
5     public interface BentukInterface {
6
7         public static final double PHI = 3.14;
8         public void tulis();
9     }
10    public interface HitungKeliling extends BentukInterface {
11        public double hitungKeliling();
12    }
13    public interface HitungLuas extends BentukInterface {
14        public double hitungLuas();
15    }
16
17    public abstract class lingkaran implements HitungKeliling, HitungLuas {
18        private double jariJari;
19
20        public lingkaran(double jariJari) {
21            this.jariJari = jariJari;
22        }
23    }
24 }
```

Problems Javadoc Declaration Search Console

<terminated> CobaBentuk (1) [Java Application] C:\Program Files\Java\jre1.8.0\_25\bin\javaw.exe (Jun 28, 2015, 7:54:33 AM)

25.0  
Bujur Sangkar  
a = 5b = 5  
12.5  
Segitiga sama kaki  
a = 5b = 5  
12.5  
Segitiga sama sisi  
a = 5b = 5