

**LAPORAN PRAKTIKUM
PEMROGRAMAN II
MODUL 5**



Polimorfisme

Oleh:

Muhammad Ryan Rizky Rahmadi

NIM. 2210817310001

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
NOVEMBER 2023**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN II
MODUL 5

Laporan Praktikum Pemrograman II Modul 4 : Polimorfisme ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Muhammad Ryan Rizky Rahmadi
NIM : 2210817310001

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Bachrul Uluum
NIM. 2010817210025

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 1 011

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	1
B. Output Program.....	5
C. Pembahasan.....	5
a. Peliharaan.....	Error! Bookmark not defined.
b. Main	Error! Bookmark not defined.
LINK GIT HUB.....	8

DAFTAR GAMBAR

Gambar 1.1 Screenshoot Output Soal 1.....	5
---	---

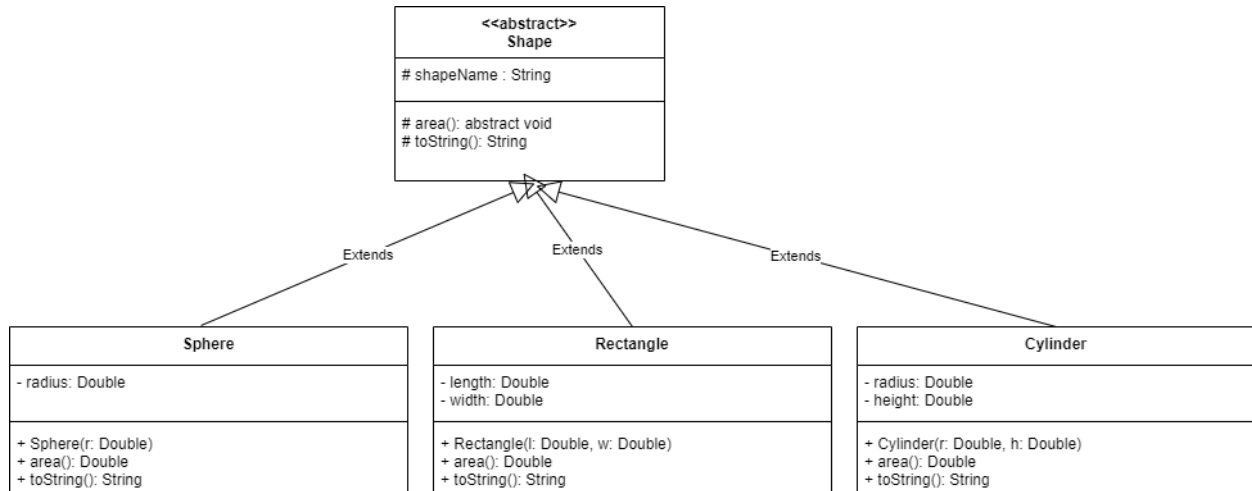
DAFTAR TABEL

Tabel 1.1 Soal 1	1
Tabel 2.1 Source Code Soal 1 Spehere.....	1
Tabel 3.1 Source Code Soal 1 Shape.....	2
Tabel 4.1 Source Code Soal 1 Rectangle	2
Tabel 5.1 Source Code Soal 1 Cylinder	3
Tabel 6.1 Source Code Soal 1 Paint	3
Tabel 7.1 Source Code Soal 1 PaintThings	4

SOAL 1

Pada praktikum kali ini anda akan diminta untuk membuat sebuah program yang dapat menghitung banyaknya liter cat yang digunakan untuk mewarnai bentuk ruang yang beragam.

Buatlah sebuah hierarki kelas abstrak Shape dimana memiliki 3 kelas anak yaitu Sphere, Rectangle, dan Cylinder seperti ditunjukkan oleh diagram kelas berikut.



Method `area()` digunakan untuk menghitung luas masing-masing objek. Berikut adalah formula yang digunakan untuk menghitung luas masing-masing bangun yang harus diimplementasikan.

Sphere: $4 \times \pi \times radius^2$

Rectangle: $length \times width$

Cylinder: $\pi \times radius^2 \times height$

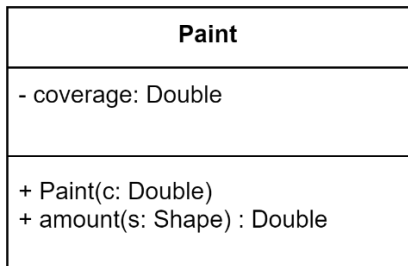
Method toString() digunakan untuk mengembalikan nilai String dari nama bangun

Berikut adalah ilustrasi dari kelas Sphere.java. Implementasikan kelas lainnya untuk Shape, Rectangle dan Cyclinder

Contoh Ilustrasi Sphere.java

```
public class Sphere extends Shape
{
private double radius; //radius in feet
// -----
// Constructor: Sets up the sphere.
// -----
public Sphere(double r)
{
super("Sphere"); radius = r;
}
// -----
// Returns the surface area of the sphere.
// -----
public double area()
{
return 4*Math.PI*(radius*radius);
}
// -----
// Returns the sphere as a String.
// -----
public String toString()
{
return super.toString() + " of radius " + radius;
}
}
```

Selanjutnya, Buatlah kelas Paint.java seperti ditunjukkan diagram kelas berikut.



Method amount digunakan untuk menghitung banyaknya liter cat yang digunakan dengan persamaan berikut:

$$\text{amount of paint} = \frac{\text{area of shape}}{\text{coverage}}$$

Lengkapi kode dibawah supaya menghasilkan keluaran yang diinginkan

Paint.java
<pre>public class Paint { private double coverage; //number of square feet per gallon // ----- // Constructor: Sets up the paint object. // ----- public Paint(double c) { coverage = c; } // ----- // Returns the amount of paint (number of gallons) // needed to paint the shape given as the parameter. // ----- public double amount(Shape s) { System.out.println ("Computing amount for " + s); return 0; } }</pre>

Terakhir, Buatlah kelas main bernama PaintThings.java. Tambahkan beberapa hal berikut agar program berjalan sesuai yang diinginkan.

1. Instansiasi 3 bentuk objek:
 - a. objek bernama deck berbentuk persegi panjang dengan ukuran Panjang 20cm dan lebar 30cm.
 - b. objek bernama bigBall berbentuk bola dengan ukuran radius 15cm.
 - c. objek bernama tank berbentuk silinder dengan ukuran radius 10cm dan tinggi 30cm.
2. Panggil fungsi yang tepat agar dapat menghitung jumlah cat yang diperlukan.

Petunjuk untuk kelas main PaintThings.java

```
import          java.text.DecimalFormat;
public class PaintThings
{
    -----
    //
    // Creates some shapes and a Paint object
    // and prints the amount of paint needed
    // to paint each shape. -----
    //
    public static void main (String[] args)
    {
        final double COVERAGE = 350;
        Paint paint = new Paint(COVERAGE);
        Rectangle deck;

        Sphere      bigBall;
        Cylinder tank;

        double deckAmt, ballAmt, tankAmt;

        // Instantiate the three shapes to paint
```

```

System.out.println ("Big Ball " + fmt.format(ballAmt)); System.out.println
("Tank " + fmt.format(tankAmt));

}

}

```

1. Jalankan program dan perhatikan hasil untuk ketiga bentuk yang berbeda, screenshot hasil yang didapatkan dan lampirkan di dalam source code.
2. Simpan coding anda dengan nama package: **soal1**
3. Pastikan terdapat screenshoot pada repositori github

Tabel 1.1 Soal 1

A. Source Code

```

1 package soal1;
2
3 import java.lang.Math;
4 public class sphere extends shape {
5     private double radius;
6
7     public sphere(Double r){
8         super("Sphere");
9         radius = r;
10    }
11
12    public Double area(){return
13 4*Math.PI*(radius*radius);}
14
15    public String toString(){return super.toString()
16 + " of radius "+radius;}
17 }

```

Tabel 2.1 Source Code Soal 1 Spehere

1	package soall;
2	
3	public abstract class shape {
4	protected String shapeName;
5	
6	public shape(String name) {
7	shapeName = name;
8	}
9	
10	protected abstract Double area();
11	
12	public String toString() {
13	return shapeName;
14	}
15	}
16	

Tabel 3.1 Source Code Soal 1 Shape

1	package soall;
2	
3	public class rectangle extends shape{
4	private Double length, width;
5	
6	public rectangle(Double l, Double w){
7	super("Rectangle");
8	length = l;
9	width = w;
10	}
11	
12	public Double area(){return length*width;}
13	
14	public String toString(){return super.toString()
15	+ " of length " + length + " and width "+width;}
16	}

Tabel 4.1 Source Code Soal 1 Rectangle

1	package soall;
2	
3	public class cylinder extends shape{
4	private Double radius, height;
5	
6	public cylinder(Double r, Double h){
7	super("Cylinder");
8	radius = r;
9	height = h;
10	}
11	
12	public Double area(){return
13	Math.PI*(radius*radius)*height;}
14	
15	public String toString(){return super.toString()
16	+ " of radius " + radius + " and height "+height;}
17	}

Tabel 5.1 Source Code Soal 1 Cylinder

1	package soall;
2	
3	public class paint {
4	private double coverage;
5	
6	public paint(double c){coverage = c;}
7	
8	public double amount(shape s){
9	System.out.println("Computing amount for "+
10	s);
11	return s.area()/coverage;
12	}
13	}

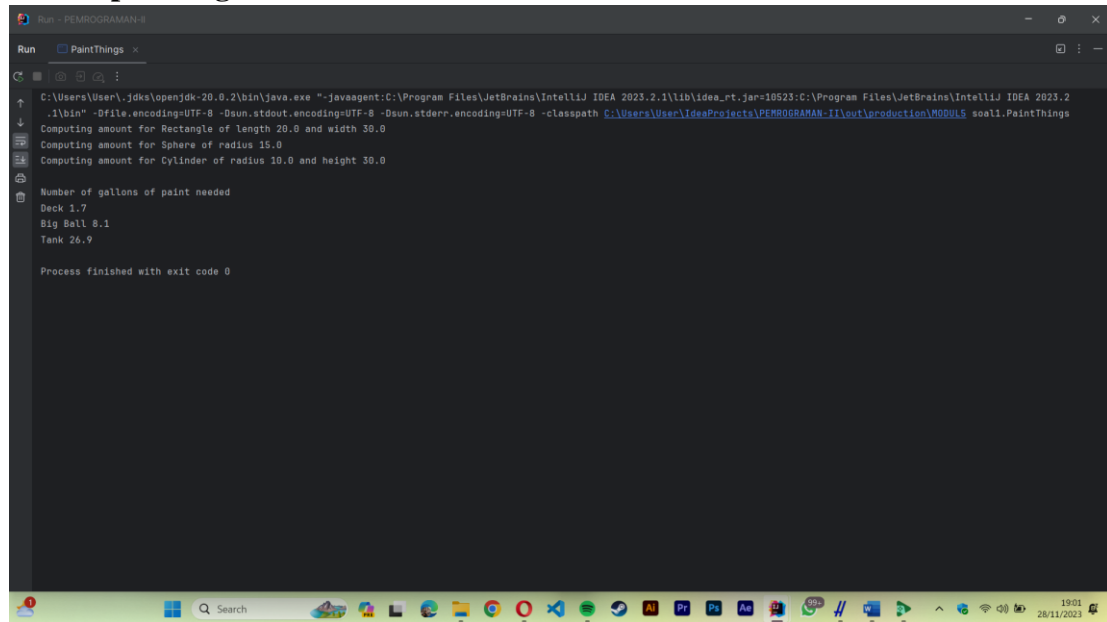
Tabel 6.1 Source Code Soal 1 Paint

1	package soall;
2	
3	import java.text.DecimalFormat;
4	
5	public class PaintThings {
6	public static void main(String[] args) {
7	final double COVERAGE = 350;
8	paint paint = new paint(COVERAGE);

9	rectangle deck;
10	sphere bigBall;
11	cylinder tank;
12	double deckAmt, ballAmt, tankAmt;
13	
14	deck = new rectangle(20.0,30.0);
15	bigBall = new sphere(15.0);
16	tank = new cylinder(10.0,30.0);
17	
18	deckAmt = paint.amount(deck);
20	ballAmt = paint.amount(bigBall);
21	tankAmt = paint.amount(tank);
22	
23	DecimalFormat fmt = new
24	DecimalFormat("0.0");
25	System.out.println("\nNumber of gallons of
26	paint needed");
27	System.out.println("Deck " +
28	fmt.format(deckAmt));
29	System.out.println("Big Ball " +
30	fmt.format(ballAmt));
31	System.out.println("Tank " +
32	fmt.format(tankAmt));
33	}
34	}

Tabel 7.1 Source Code Soal 1 PaintThings

B. Output Program



```
C:\Users\User\jdk\openjdk-20.0.2\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\lib\idea_rt.jar=10523:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath C:\Users\User\IdeaProjects\PEMROGRAMAN-11\out\production\MODUL5 soal1\PaintThings
Computing amount for Rectangle of length 20.0 and width 30.0
Computing amount for Sphere of radius 15.0
Computing amount for Cylinder of radius 10.0 and height 30.0

Number of gallons of paint needed
Deck 1.7
Big Ball 8.1
Tank 26.9

Process finished with exit code 0
```

Gambar 1.1 Screenshot Output Soal 1

C. Pembahasan

a. Sphere

Line 1 : *Package*, untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3: *Import*, suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 4, 7, 12, dan 15: *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 4 dan 17 : {}, penanda awal dan akhir *block code*..

b. Shape

Line 1 : *Package*, untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3, 6, dan 12 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 3 dan 15 : {}, penanda awal dan akhir *block code*.

Line 10 : *Abstract*, sebagai ekspresi konsep umum tempat kelas yang lebih spesifik dapat diturunkan.

Line 13 : *Return*, mengembalikan nilai dari sebuah function.

c. Rectangle

Line 1 : *Package*, untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3, 6, 12, dan 14: `Public`, kunci pada java yang dapat diakses kelas class lain.

Line 3 : *Extends* , kata kunci yang ditulis dengan kelas anak selama deklarasi kelas diikuti dengan nama kelas induk.

Line 3 dan 16: `{ }`, penanda awal dan akhir block code.

d. Cylinder

Line 1 : `Package`, untuk pengelompokan class atau interface menjadi satu unit.

Line 3 : *Extends* , kata kunci yang ditulis dengan kelas anak selama deklarasi kelas diikuti dengan nama kelas induk.

Line 3, 6, 12, dan 15: `Public`, kunci pada java yang dapat diakses kelas class lain.

Line 3 dan 17 : `{ }`, penanda awal dan akhir block code.

Line 15 : *Return*, mengembalikan nilai dari sebuah function.

e. Paint

Line 1 : `Package`, untuk pengelompokan class atau interface menjadi satu unit.

Line 3, 6, dan 8: `Public`, kunci pada java yang dapat diakses kelas class lain.

Line 3 dan 13 : `{ }`, penanda awal dan akhir block code.

Line 9 : `System.out.println` mencetak sesuatu dari program sebagai output.

Line 11 : *Return*, mengembalikan nilai dari sebuah function.

f. PaintThings

Line 1 : `Package`, untuk pengelompokan class atau interface menjadi satu unit.

Line 2: `Import`, suatu perintah untuk memasukkan suatu method atau perintah dalam Bahasa Java.

Line 4 dan 11: `Public`, kunci pada java yang dapat diakses kelas class lain.

Line 4 dan 34 : `{ }`, penanda awal dan akhir block code.

Line 24-31 : `System.out.println` mencetak sesuatu dari program sebagai output.

LINK GIT HUB

<https://github.com/RylenRawr/PEMROGRAMAN-II/tree/main/MODUL%205>