

# LAPORAN PRAKTIKUM

## PEMROGRAMAN II

### MODUL 5



## POLIMORFISME

Oleh:

# Rizki Adhitiya Maulana

NIM. 2410817110014

**PROGRAM STUDI TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS LAMBUNG MANGKURAT**  
**NOVEMBER 2025**

**LEMBAR PENGESAHAN**  
**LAPORAN PRAKTIKUM PEMROGRAMAN II**  
**MODUL 5**

Laporan Praktikum Pemrograman II Modul 5: Polimorfisme ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Rizki Adhitiya Maulana  
NIM : 2410817110014

Jovan Gilbert Natamasindah  
NIM. 2310817310002

Irham Maulani Abdul Gani, S.Kom.,  
M.Kom.

NIP. 199710312025061009

## **DAFTAR ISI**

LEMBAR PENGESAHAN .....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR .....	iv
DAFTAR TABEL.....	v
SOAL 1 .....	6
A.    Source Code .....	10
B.    Output Program.....	13
C.    Pembahasan.....	13
D.    GitHub.....	20
TAUTAN GIT.....	21

## **DAFTAR GAMBAR**

Gambar 1. Diagram Kelas UML Soal Praktikum .....	6
Gambar 2. Diagram Kelas UML Paint.....	7
Gambar 3. Rumus Perhitungan Jumlah Cat yang Dibutuhkan .....	7
Gambar 4. Screenshot Hasil Jawaban Soal 1 .....	13
Gambar 5. Screenshot pada Repositori GitHub .....	20

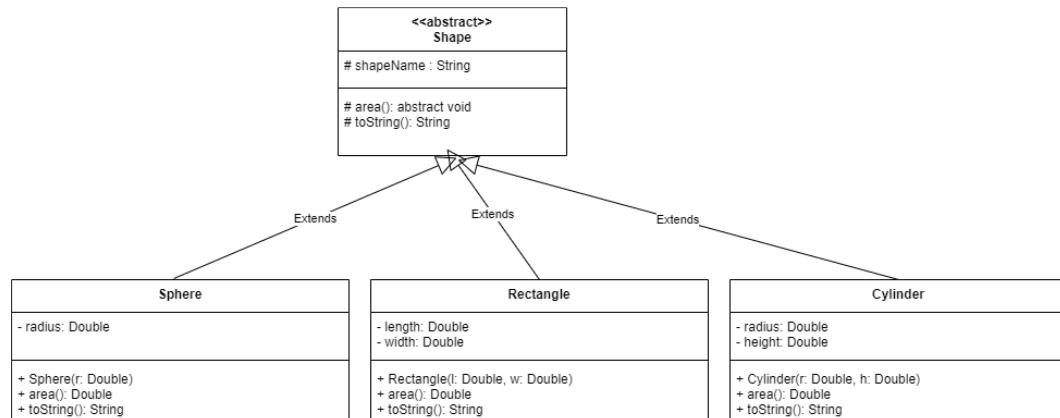
## **DAFTAR TABEL**

Tabel 1. Soal 1 Ilustrasi Sphere.java.....	7
Tabel 2. Soal 1 Ilustrasi Paint.java.....	8
Tabel 3. Soal 1 Ilustrasi PaintThings.java.....	8
Tabel 4. Source Code Soal 1 Shape.java.....	10
Tabel 5. Source Code Soal 1 Sphere.java .....	10
Tabel 6. Source Code Soal 1 Rectangle.java .....	11
Tabel 7. Source Code Soal 1 Cylinder.java .....	11
Tabel 8. Source Code Soal 1 Paint.java .....	12
Tabel 9. Source Code Soal 1 PaintThings.java.....	12

## 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.



Gambar 1. Diagram Kelas UML Soal Praktikum

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

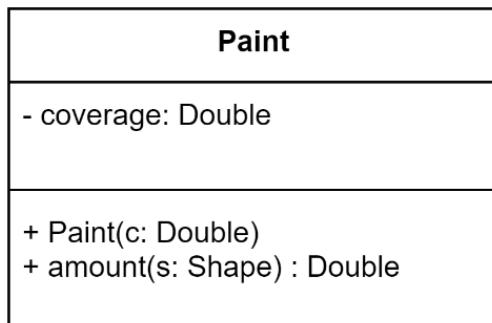
Method toString() digunakan untuk mengembalikan nilai String dari nama bangun. Berikut adalah ilustrasi dari kelas Sphere.java. Implementasikan kelas lainnya untuk Shape, Rectangle dan Cyclinder.

Tabel 1. Soal 1 Ilustrasi Sphere.java

**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.



Gambar 2. Diagram Kelas UML Paint

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

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

Gambar 3. Rumus Perhitungan Jumlah Cat yang Dibutuhkan

Lengkapi kode dibawah supaya menghasilkan keluaran yang diinginkan

Tabel 2. Soal 1 Ilustrasi Paint.java

**Paint.java**

```
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;
    }
}
```

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

Tabel 3. Soal 1 Ilustrasi 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
    // Compute the amount of paint needed for each shape
    // Print the amount of paint for each.
    DecimalFormat fmt = new DecimalFormat("0.#");
    System.out.println ("\nNumber of gallons of paint needed...");
    System.out.println ("Deck " + fmt.format(deckAmt));
    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: **soall**.
3. Pastikan terdapat screenshoot pada repositori github.

## A. Source Code

Tabel 4. Source Code Soal 1 Shape.java

Shape.java	
1	package soal1;
2	
3	public abstract class Shape
4	{
5	protected String shapeName;
6	
7	public Shape(String name)
8	{
9	shapeName = name;
10	}
11	
12	public abstract double area();
13	
14	public String toString()
15	{
16	return shapeName;
17	}
18	}

Tabel 5. Source Code Soal 1 Sphere.java

Sphere.java	
1	package soal1;
2	
3	public class Sphere extends Shape
4	{
5	private double radius;
6	
7	public Sphere(double r)
8	{
9	super("Sphere");
10	radius = r;
11	}
12	
13	public double area()
14	{
15	return 4 * Math.PI * (radius * radius);
16	}
17	
18	public String toString()
19	{
20	return super.toString() + " of radius " +
21	radius;
22	}

Tabel 6. Source Code Soal 1 Rectangle.java

**Rectangle.java**

```
1 package soall;
2
3 public class Rectangle extends Shape
4 {
5     private double length;
6     private double width;
7
8     public Rectangle(double l, double w)
9     {
10         super("Rectangle");
11         length = l;
12         width = w;
13     }
14
15     public double area()
16     {
17         return length * width;
18     }
19
20     public String toString()
21     {
22         return super.toString() + " of length " + length
23         + " and width " + width;
24     }
}
```

Tabel 7. Source Code Soal 1 Cylinder.java

**Cylinder.java**

```
1 package soall;
2
3 public class Cylinder extends Shape
4 {
5     private double radius;
6     private double height;
7
8     public Cylinder(double r, double h)
9     {
10         super("Cylinder");
11         radius = r;
12         height = h;
13     }
14
15     public double area()
16     {
17         return Math.PI * (radius * radius) * height;
18     }
}
```

18	}
19	
20	public String toString()
21	{
22	return super.toString() + " of radius " + radius
23	+ " and height " + height;
24	}

Tabel 8. Source Code Soal 1 Paint.java

<b>Paint.java</b>	
1	package soall;
2	
3	public class Paint
4	{
5	private double coverage;
6	
7	public Paint(double c)
8	{
9	coverage = c;
10	}
11	
12	public double amount(Shape s)
13	{
14	System.out.println("Computing amount for " + s);
15	return s.area() / coverage;
16	}
17	}

Tabel 9. Source Code Soal 1 PaintThings.java

<b>Paint.java</b>	
1	package soall;
2	
3	import java.text.DecimalFormat;
4	
5	public class PaintThings
6	{
7	public static void main(String[] args)
8	{
9	final double COVERAGE = 350;
10	Paint paint = new Paint(COVERAGE);
11	
12	Rectangle deck;
13	Sphere bigBall;
14	Cylinder tank;
15	
16	double deckAmt, ballAmt, tankAmt;
17	}

```

18     deck = new Rectangle(20, 30);
19     bigBall = new Sphere(15);
20     tank = new Cylinder(10, 30);
21
22     deckAmt = paint.amount(deck);
23     ballAmt = paint.amount(bigBall);
24     tankAmt = paint.amount(tank);
25
26     DecimalFormat fmt = new DecimalFormat("0.#");
27     System.out.println("\nNumber of gallons of paint
needed...");
28     System.out.println("Deck " +
fmt.format(deckAmt));
29     System.out.println("Big Ball " +
fmt.format(ballAmt));
30     System.out.println("Tank " +
fmt.format(tankAmt));
31 }
32 }
```

## B. Output Program

The screenshot shows an IDE interface with several tabs at the top: Shape.java, Sphere.java, Rectangle.java, Cylinder.java, Paint.java, and PaintThings.java. The PaintThings tab is active. In the center, there is a terminal-like window showing the following text:

```

C:\Users\Advan\jdks\openjdk-24.0.2+54\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2\lib\idea_rt.jar" -Dfile.encoding=UTF-8 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 4. Screenshot Hasil Jawaban Soal 1

## C. Pembahasan

- **Shape.java**

Pada baris [1] terdapat `package soall;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soall.

Pada baris [3] terdapat `public abstract class Shape` dimana berfungsi untuk membuat class abstrak publik bernama Shape. Di karenakan

bersifat abstrak, class ini tidak dapat dibuat objeknya secara langsung dan hanya digunakan sebagai cetakan untuk class turunannya.

Pada baris [5] terdapat `protected String shapeName;` dimana berfungsi untuk menyimpan nama bentuk dalam tipe data String, sehingga variabel ini dapat menyimpan teks. `protected` berarti variabel ini dapat diakses oleh class turunan maupun class lain yang berada dalam package yang sama.

Pada baris [7] sampai [10] terdapat constructor `public Shape (String name)` dimana berfungsi untuk memberikan nilai awal pada variabel `shapeName` ketika objek dari class turunan dibuat. `shapeName = name;` berfungsi untuk menyimpan nilai ke dalam variabel `shapeName`.

Pada baris [12] terdapat `public abstract double area();` dimana berfungsi untuk mendeklarasikan metode abstrak bernama `area()` yang akan digunakan untuk menghitung luas bentuk.

Pada baris [14] sampai [17] terdapat `public String toString()` dimana berfungsi untuk menampilkan informasi yang ada di dalamnya. `return shapeName;` berfungsi untuk mengembalikan nilai dari variabel `shapeName` untuk ditampilkan ke layar.

Pada baris [4] sampai [18] terdapat tanda `{}` dimana digunakan untuk membatasi blok kode yang ada pada program.

- **Sphere.java**

Pada baris [1] terdapat `package soal1;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama `soal1`.

Pada baris [3] terdapat `public class Sphere extends Shape` dimana berfungsi untuk membuat class publik bernama `Sphere` yang mewarisi atau mengambil sifat dari class `Shape`. `extends` digunakan untuk menyatakan bahwa class `Sphere` mewarisi seluruh atribut dan metode dari class `Shape`.

Pada baris [5] terdapat `private double radius;` dimana berfungsi untuk menyimpan nilai jari-jari bola dalam tipe data double, sehingga variabel ini dapat menyimpan desimal.

Pada baris [7] sampai [11] terdapat constructor `public Sphere(double r)` dimana berfungsi untuk memberikan nilai awal pada objek Sphere saat dibuat. `super("Sphere");` berfungsi untuk memanggil constructor dari class Shape. `radius = r;` berfungsi untuk menyimpan nilai r ke dalam variabel radius milik class Sphere.

Pada baris [13] sampai [16] terdapat `public double area()` dimana berfungsi untuk menghitung luas. `return 4 * Math.PI * (radius * radius);` berfungsi untuk menghitung luas dengan rumus  $4 \times \pi \times r^2$ .

Pada baris [18] sampai [21] terdapat `public String toString()` dimana berfungsi untuk menampilkan informasi mengenai objek Sphere. `return super.toString() + " of radius " + radius;` berfungsi untuk menggabungkan teks hasil dari `super.toString()` dari class induk dengan teks " of radius ", lalu nilai jari-jari dari variabel radius.

Pada baris [4] sampai [22] terdapat tanda `{}` dimana digunakan untuk membatasi blok kode yang ada pada program.

- **Rectangle.java**

Pada baris [1] terdapat `package soal1;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal1.

Pada baris [3] terdapat `public class Rectangle extends Shape` dimana berfungsi untuk membuat class publik bernama Rectangle yang mewarisi atau mengambil sifat dari class Shape. `extends` digunakan untuk menyatakan bahwa class Rectangle mewarisi seluruh atribut dan metode dari class Shape.

Pada baris [5] terdapat `private double length;` dimana berfungsi untuk menyimpan panjang persegi panjang dalam tipe data double, sehingga variabel ini dapat menyimpan desimal.

Pada baris [6] terdapat `private double width;` dimana berfungsi untuk menyimpan lebar persegi panjang dalam tipe data double, sehingga variabel ini dapat menyimpan desimal.

Pada baris [8] sampai [13] terdapat constructor `public Rectangle(double l, double w)` dimana berfungsi untuk memberikan nilai awal pada objek Rectangle saat dibuat. `super("Rectangle");` berfungsi untuk memanggil constructor dari class induk. `length = l;` berfungsi untuk menyimpan nilai ke dalam variabel length. `width = w;` berfungsi untuk menyimpan nilai ke dalam variabel width.

Pada baris [15] sampai [18] terdapat `public double area()` dimana berfungsi untuk menghitung luas. `return length * width;` berfungsi untuk mengembalikan hasil perkalian antara panjang dan lebar.

Pada baris [20] sampai [23] terdapat `public String toString()` dimana berfungsi untuk menampilkan informasi objek. `return super.toString() + " of length " + length + " and width " + width;` berfungsi untuk menggabungkan teks hasil dari `super.toString()` dari class induk dengan tambahan teks "of length" dan "and width", serta nilai dari variabel length dan width.

Pada baris [4] sampai [24] terdapat tanda `{}` dimana digunakan untuk membatasi blok kode yang ada pada program.

- **Cylinder.java**

Pada baris [1] terdapat `package soall;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soall.

Pada baris [3] terdapat `public class Cylinder extends Shape` dimana berfungsi untuk membuat class publik bernama Cylinder yang mewarisi atau mengambil sifat dari class Shape. `extends` digunakan untuk menyatakan bahwa class Cylinder mewarisi seluruh atribut dan metode dari class Shape.

Pada baris [5] terdapat `private double radius;` dimana berfungsi untuk menyimpan nilai jari-jari tabung dalam tipe data double, sehingga variabel ini dapat menyimpan desimal.

Pada baris [6] terdapat `private double height;` dimana berfungsi untuk menyimpan nilai tinggi tabung dalam tipe data double, sehingga variabel ini dapat menyimpan desimal.

Pada baris [8] sampai [13] terdapat constructor `public Cylinder(double r, double h)` dimana berfungsi untuk memberikan nilai awal pada objek Cylinder saat dibuat. `super("Cylinder");` berfungsi untuk memanggil constructor dari class induk. `radius = r;` berfungsi untuk menyimpan nilai ke dalam variabel `radius`. `height = h;` berfungsi untuk menyimpan nilai ke dalam variabel `height`.

Pada baris [15] sampai [18] terdapat `public double area()` dimana berfungsi untuk menghitung luas. `return Math.PI * (radius * radius) * height;` berfungsi untuk menghitung hasil perkalian  $\pi \times r^2 \times h$ .

Pada baris [20] sampai [23] terdapat `public String toString()` dimana berfungsi untuk menampilkan informasi objek Cylinder. `return super.toString() + " of radius " + radius + " and height " + height;` berfungsi untuk menggabungkan teks hasil dari `super.toString()` dari class induk dengan tambahan teks "`of radius`" dan "`and height`", serta menampilkan nilai dari variabel `radius` dan `height`.

Pada baris [4] sampai [24] terdapat tanda `{}` dimana digunakan untuk membatasi blok kode yang ada pada program.

- **Paint.java**

Pada baris [1] terdapat `package soal1;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal1.

Pada baris [3] terdapat `public class Paint` dimana berfungsi untuk membuat class publik bernama Paint.

Pada baris [5] terdapat `private double coverage;` dimana berfungsi untuk menyimpan nilai daya sebar cat dalam tipe data double, sehingga variabel ini dapat menyimpan desimal.

Pada baris [7] sampai [10] terdapat constructor `public Paint(double c)` dimana berfungsi untuk memberikan nilai awal pada variabel coverage saat objek Paint dibuat. `coverage = c;` berfungsi untuk menyimpan nilai ke dalam variabel c.

Pada baris [12] sampai [16] terdapat `public double amount(Shape s)` dimana berfungsi untuk menampilkan informasi objek. `System.out.println("Computing amount for " + s);` berfungsi untuk menampilkan teks ke layar berupa informasi. `return s.area() / coverage;` berfungsi untuk menghitung jumlah cat yang diperlukan dengan membagi luas dengan nilai coverage.

Pada baris [4] sampai [17] terdapat tanda `{}` dimana digunakan untuk membatasi blok kode yang ada pada program.

- **PaintThings.java**

Pada baris [1] terdapat `package soal1;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal1.

Pada baris [3] terdapat `import java.text.DecimalFormat;` dimana berfungsi untuk mengimpor class DecimalFormat dari package java.text, yang digunakan untuk mengatur tampilan angka desimal.

Pada baris [5] terdapat `public class PaintThings` dimana berfungsi untuk membuat class publik bernama PaintThings.

Pada baris [7] sampai [31] terdapat `public static void main(String[] args)` dimana berfungsi sebagai metode utama dari program Java. Semua kode di dalam blok ini akan dijalankan saat program dimulai.

Pada baris [9] terdapat `final double COVERAGE = 350;` dimana berfungsi untuk menyimpan nilai konstan sebesar 350.

Pada baris [10] terdapat `Paint paint = new Paint(COVERAGE);` dimana berfungsi untuk membuat objek Paint.

Pada baris [12] sampai [14] terdapat deklarasi variabel `Rectangle deck;`, `Sphere bigBall;`, dan `Cylinder tank;` dimana masing-masing berfungsi untuk menyimpan objek bentuk persegi panjang, bola besar, dan tabung.

Pada baris [16] terdapat deklarasi variabel `double deckAmt,` `ballAmt,` `tankAmt;` dimana berfungsi untuk menyimpan hasil perhitungan untuk masing-masing objek.

Pada baris [18] terdapat `deck = new Rectangle(20, 30);` berfungsi untuk membuat objek Rectangle dengan panjang 20 dan lebar 30.

Pada baris [19] terdapat `bigBall = new Sphere(15);` berfungsi untuk membuat objek Sphere dengan jari-jari 15.

Pada baris [20] terdapat `tank = new Cylinder(10, 30);` berfungsi untuk membuat objek Cylinder dengan jari-jari 10 dan tinggi 30.

Pada baris [22] terdapat `deckAmt = paint.amount(deck);` menghitung jumlah cat untuk objek Rectangle.

Pada baris [23] terdapat `ballAmt = paint.amount(bigBall);` menghitung jumlah cat untuk objek Sphere.

Pada baris [24] terdapat `tankAmt = paint.amount(tank);` menghitung jumlah cat untuk objek Cylinder.

Pada baris [26] terdapat `DecimalFormat fmt = new DecimalFormat("0.#");` dimana berfungsi untuk membuat objek yang akan menampilkan angka dengan satu digit di belakang koma.

Pada baris [27] terdapat `System.out.println("\nNumber of gallons of paint needed...");` dimana berfungsi untuk menampilkan judul hasil perhitungan ke layar.

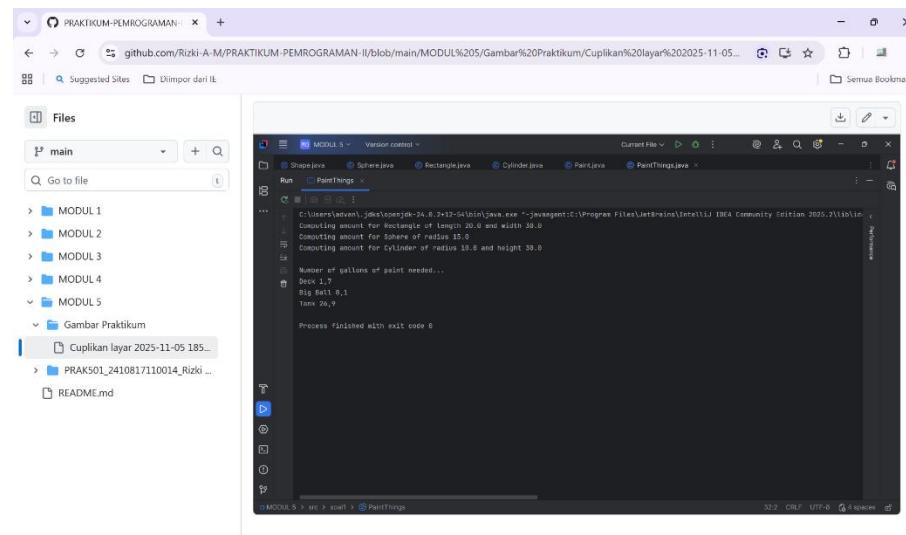
Pada baris [28] terdapat `"Deck " + fmt.format(deckAmt)` menampilkan hasil perhitungan untuk persegi panjang.

Pada baris [29] terdapat `"Big Ball " + fmt.format(ballAmt)` menampilkan hasil perhitungan untuk bola besar.

Pada baris [30] terdapat `"Tank " + fmt.format(tankAmt)` menampilkan hasil perhitungan untuk tabung.

Pada baris [6] sampai [32] terdapat tanda `{ }` dimana digunakan untuk membatasi blok kode yang ada pada program.

## D. GitHub



Gambar 5. Screenshot pada Repositori GitHub

## **TAUTAN GIT**

<https://github.com/Rizki-A-M/PRAKTIKUM-PEMROGRAMAN-II.git>