

# Pemrograman Berorientasi Objek

---

## Praktikum Modul 1

---

Disusun oleh:

Nama: Gian Alharitz Ueldy Secondri

NIM: 11241033

---

### Tugas dan Deskripsi Singkat

1. Dengan menerapkan *object* dan *method*, buatlah program Java untuk menghitung:

- Luas isi tabung
- Luas persegi panjang
- Luas trapesium

2. Buatlah dua program, yaitu:

- Program **prosedural**
- Program **Berorientasi Objek (OOP)**

Keduanya harus dapat menghitung **volume tabung**.

Tunjukkan perbedaan antara pemrograman prosedural dan OOP.

---

### Pembahasan

#### Main.java

```
public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5, 10);
        System.out.println("Rectangle Area: " + rectangle.calculateArea());

        Trapezium trapezium = new Trapezium(5, 10, 15);
        System.out.println("Trapezium Area: " + trapezium.calculateArea());

        Cylinder cylinder = new Cylinder(5, 10);
        System.out.println("Cylinder Area: " + cylinder.calculateArea());
        System.out.println("Cylinder Volume (OOP): " + cylinder.calculateVolume());
    }
}
```

#### Shape.java

```
public abstract class Shape {
    public abstract double calculateArea();
}
```

#### Rectangle.java

```
public class Rectangle extends Shape {
    private double width;
    private double height;

    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    @Override
    public double calculateArea() {
        return width * height;
    }
}
```

#### Trapezium.java

```
public class Trapezium extends Shape {
    private double base1;
    private double base2;
    private double height;

    public Trapezium(double base1, double base2, double height) {
        this.base1 = base1;
        this.base2 = base2;
        this.height = height;
    }

    @Override
    public double calculateArea() {
        return 0.5 * (base1 + base2) * height;
    }
}
```

#### Cylinder.java

```
public class Cylinder extends Shape {
    private double radius;
    private double height;

    public Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }

    @Override
    public double calculateArea() {
        return 2 * Math.PI * radius * (radius + height); // Luas permukaan
    }

    public double calculateVolume() {
        return Math.PI * radius * radius * height; // Volume tabung
    }
}
```

---

## Versi Prosedural (Untuk Perbandingan)

---

#### ProceduralVolume.java

```
public class ProceduralVolume {
    public static void main(String[] args) {
        double radius = 5;
        double height = 10;

        double volume = calculateCylinderVolume(radius, height);
        System.out.println("Cylinder Volume (Procedural): " + volume);
    }

    public static double calculateCylinderVolume(double r, double h) {
        return Math.PI * r * r * h;
    }
}
```

## Perbandingan Prosedural vs OOP

Aspek	Prosedural	OOP (Object-Oriented Programming)
Struktur Program	Fungsi berdiri sendiri	Tersusun atas class dan object
Reusability (Pemakaian Ulang)	Sulit karena tidak modular	Mudah dengan inheritance & polymorphism
Data dan Logika	Terpisah	Dikelompokkan dalam class (encapsulation)
Contoh	<code>calculateCylinderVolume()</code>	<code>cylinder.calculateVolume()</code>
Cocok Untuk	Program sederhana	Program besar dan kompleks