LAPORAN AKHIR PRAKTIKUM

Mata Praktikum	: Rekayasa Perangkat Lunak 2
Kelas	: 4IA07
Praktikum ke-	: 2
Tanggal	: 25 Oktober 2025
Materi	: Object-Oriented Programming (OOP)
NPM	: 51422084
Nama	: Muhammad Izzuddin Almansyur
Ketua Asisten	: Haikal Abizar
Jumlah Lembar	: 6

LABORATORIUM INFORMATIKA

UNIVERSITAS GUNADARMA

2025

1. Jelaskan konsep-konsep utama dalam Object-Oriented Programming (OOP) seperti Encapsulation, Inheritance, dan Polymorphism. Berikan contoh penerapan setiap konsep ini dalam program Java.

- **Encapsulation**

Encapsulation adalah proses menyembunyikan detail implementasi internal dari sebuah objek dan hanya memperlihatkan antarmuka publik yang diperlukan. Tujuannya adalah untuk melindungi data agar tidak diakses langsung dari luar class, melainkan melalui method yang telah disediakan. Biasanya dilakukan dengan menjadikan atribut bersifat private dan menyediakan method getter dan setter.

Contoh

```
public class AkunBank {
    private double saldo;

    public void setSaldo(double jumlah) {
        if (jumlah >= 0) {
            saldo = jumlah;
        }
    }

    public double getSaldo() {
        return saldo;
    }
}
```

Pada contoh di atas, atribut saldo tidak bisa diakses langsung. Hanya bisa dimodifikasi dan dibaca melalui method setSaldo() dan getSaldo(), sehingga data lebih aman dan terkontrol.

- **Inheritance**

Inheritance adalah proses pewarisan atribut dan method dari satu class ke class lain. Class yang mewariskan disebut superclass, sedangkan class yang menerima warisan disebut subclass. Konsep ini memungkinkan reuse kode dan memperluas fungsionalitas tanpa harus menulis ulang.

```java
public class Kendaraan {
    public void nyalakanMesin() {
        System.out.println("Mesin dinyalakan");
    }
}

public class Mobil extends Kendaraan {
    public void klakson() {
        System.out.println("Mobil membunyikan klakson");
    }
}
```

Class Mobil mewarisi method nyalakanMesin() dari class Kendaraan, dan menambahkan method baru klakson().

- **Polymorphism**

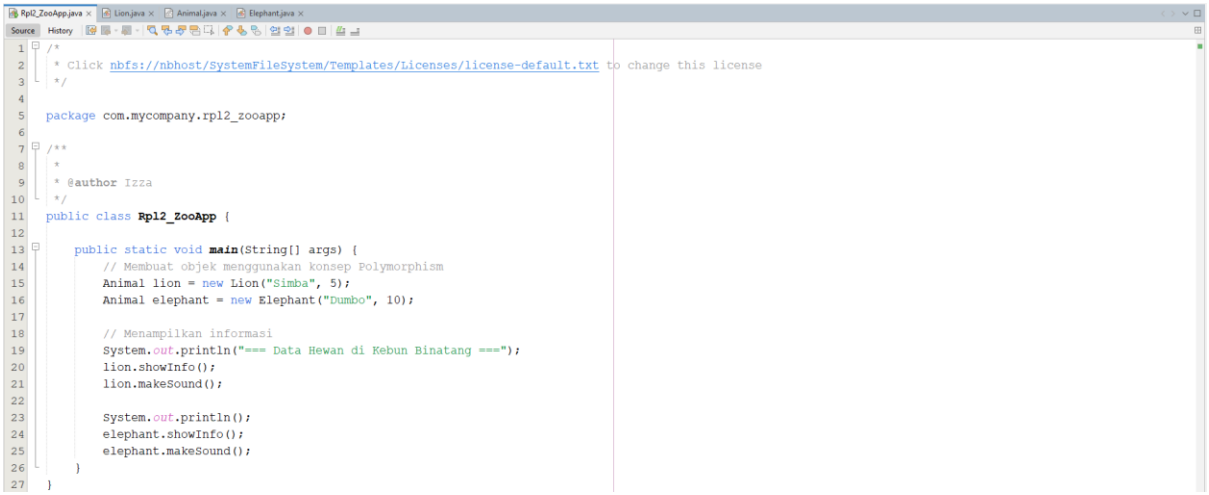
Polymorphism memungkinkan objek untuk memiliki banyak bentuk, terutama dalam hal method yang sama namun perilaku berbeda. Ada dua jenis utama: method overloading (compile-time polymorphism) dan method overriding (runtime polymorphism).

```java
public class Hewan {
    public void suara() {
        System.out.println("Hewan bersuara");
    }
}

public class Anjing extends Hewan {
    @Override
    public void suara() {
        System.out.println("Anjing menggonggong");
    }
}

public class Kucing extends Hewan {
    @Override
    public void suara() {
        System.out.println("Kucing mengeong");
    }
}
```

Method suara() dimiliki oleh class Hewan, tetapi perilakunya berbeda saat dipanggil oleh objek Anjing atau Kucing.

2. Buatlah sebuah progam yang memanfaatkan konsep-konsep OOP dan berikan penjelasannya

- **Rpl2_ZooApp**



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.rpl2_zooapp;

/**
 *
 * @author Izza
 */
public class Rpl2_ZooApp {

    public static void main(String[] args) {
        // Membuat objek menggunakan konsep Polymorphism
        Animal lion = new Lion("Simba", 5);
        Animal elephant = new Elephant("Dumbo", 10);

        // Menampilkan informasi
        System.out.println("=== Data Hewan di Kebun Binatang ===");
        lion.showInfo();
        lion.makeSound();

        System.out.println();
        elephant.showInfo();
        elephant.makeSound();
    }
}
```

- **Animal**

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.rpl2_zooapp;

/**
 *
 * @author Izza
 */
public abstract class Animal {
    // Enkapsulasi: atribut dibuat private
    private String name;
    private int age;

    // Constructor
    public Animal(String name, int age) {
        this.name = name;
        this.age = age;
    }

    // Getter & Setter (Encapsulation)
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    // Method abstract (Abstraction)
    public abstract void makeSound();

    // Method umum
    public void showInfo() {
        System.out.println("Nama: " + name + ", Umur: " + age + " tahun");
    }
}
```

- **Lion**

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.rpl2_zooapp;

/**
 *
 * @author Izza
 */
public class Lion extends Animal {

    public Lion(String name, int age) {
        super(name, age);
    }

    // Polimorfisme: override method abstract
    @Override
    public void makeSound() {
        System.out.println("Aummm! (Singa mengaum)");
    }
}
```

- **Elephant**

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.rpl2_zooapp;

/**
 *
 * @author Izza
 */
public class Elephant extends Animal {

    public Elephant(String name, int age) {
        super(name, age);
    }

    @Override
    public void makeSound() {
        System.out.println("Pawoooo! (Gajah bersuara)");
    }
}
```

- **Output**

```
--- exec:3.1.0:exec (default-cli) @ rpl2_ZooApp ---
=== Data Hewan di Kebun Binatang ===
Nama: Simba, Umur: 5 tahun
Aummm! (Singa mengaum)

Nama: Dumbo, Umur: 10 tahun
Pawoooo! (Gajah bersuara)
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.132 s
Finished at: 2025-10-27T16:00:20+07:00
------------------------------------------------------------------------
```

- **Penjelasan**

Program di atas merupakan contoh penerapan konsep-konsep dasar OOP (Object-Oriented Programming). Program ini bertema “Sistem Data Hewan di Kebun Binatang”, di mana terdapat kelas induk bernama Animal dan dua kelas turunan yaitu Lion dan Elephant.

Konsep **enkapsulasi** (**encapsulation**) diterapkan dengan membuat atribut name dan age di dalam kelas Animal bersifat private, sehingga tidak dapat diakses langsung dari luar kelas. Untuk mengakses atau mengubah nilai atribut tersebut, digunakan getter dan setter. Hal ini bertujuan untuk melindungi data agar tidak dimanipulasi secara langsung dan menjaga keamanan struktur data.

Konsep **pewarisan** (**inheritance**) terlihat dari hubungan antara kelas Animal sebagai induk dengan kelas Lion dan Elephant sebagai turunannya. Dengan menggunakan pewarisan, kedua kelas turunan secara otomatis memiliki atribut dan method yang terdapat pada kelas induk Animal. Ini mempermudah pengembangan kode karena tidak perlu menulis ulang atribut atau method yang sama.

Selanjutnya, konsep **abstraksi** (**abstraction**) diterapkan melalui penggunaan abstract class Animal, yang memiliki abstract method makeSound(). Method ini tidak memiliki implementasi di kelas induk, tetapi wajib diimplementasikan oleh setiap kelas turunan. Dengan cara ini, program hanya menampilkan perilaku umum hewan, sedangkan detail perilaku spesifik diatur oleh masing-masing kelas turunan.

Konsep terakhir adalah **polimorfisme** (**polymorphism**), yang tampak pada pembuatan objek dengan tipe referensi Animal, tetapi berisi objek Lion dan Elephant. Ketika method makeSound() dipanggil, sistem secara otomatis menyesuaikan perilaku sesuai tipe objek yang sebenarnya, bukan tipe referensinya. Artinya, lion.makeSound() akan menghasilkan suara singa, sedangkan elephant.makeSound() akan menghasilkan suara gajah.