UTS

PRAKTIKUM PBO

Nama : Muhamad Fikri Zaelani

Nim : 1227050078

Kelas : IF-C

---

SOURCE CODE

```java
// Interface
interface Animal {
    void eat();
    void makeSound();
}

// Encapsulation and Inheritance
class Cat implements Animal {
    private String name;

    public Cat(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    @Override
    public void eat() {
        System.out.println(name + " sedang makan");
    }

    @Override
    public void makeSound() {
        System.out.println(name + " sedang mengeong");
    }
}

// Polymorphism
class Dog implements Animal {
    private String name;
```

```java
    public Dog(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    @Override
    public void eat() {
        System.out.println(name + " sedang makan");
    }

    @Override
    public void makeSound() {
        System.out.println(name + " sedang menggonggong");
    }
}

// Abstract class
abstract class AnimalWithChild {
    abstract void nurse();
}

// Inheritance
class CatWithChild extends AnimalWithChild {
    @Override
    void nurse() {
        System.out.println("Susy sedang menyusui anaknya");
    }
}

// Main class
class Main {
    public static void main(String[] args) {
        System.out.println("===== UTS Praktikum PBO =====");
        System.out.println("--------------------------------");
        System.out.println("Nama\t: Muhamad Fikri Zaelani");
        System.out.println("NIM\t: 1227050078");
        System.out.println("Kelas\t: IF-C");
        System.out.println("================================\n");

        Cat garfield = new Cat("Garfield");
        Dog lucky = new Dog("Lucky");
        CatWithChild susy = new CatWithChild();

        garfield.eat();
        garfield.makeSound();
```

```
        lucky.eat();
        lucky.makeSound();

        susy.nurse();
    }
}
```

OUTPUT

```
===== UTS Praktikum PBO =====
------------------------------
Nama    : Muhamad Fikri Zaelani
NIM     : 1227050078
Kelas   : IF-C
==============================

Garfield sedang makan
Garfield sedang mengeong
Lucky sedang makan
Lucky sedang menggonggong
Susy sedang menyusui anaknya
```

Penjelasan singkat program sederhana yang menggunakan konsep-konsep dasar dari Pemrograman Berorientasi Objek (PBO)

1. Interface (Animal): Interface adalah kontrak untuk kelas-kelas lain yang akan mengimplementasikannya. Dalam contoh ini, Animal adalah sebuah interface yang mendefinisikan dua metode: eat() dan makeSound().

2. Encapsulation dan Inheritance (Cat dan Dog): Konsep ini menggabungkan pembungkusan data (encapsulation) dan pewarisan (inheritance). Kelas Cat dan Dog mewarisi sifat-sifat dari interface Animal. Mereka juga menunjukkan encapsulation dengan menyembunyikan atribut name dan memberikan akses kepadanya melalui metode getName().

3. Polymorphism: Polimorfisme adalah kemampuan sebuah objek untuk dapat memperlihatkan perilaku yang berbeda tergantung pada konteksnya. Dalam program ini, kita menggunakan polimorfisme saat objek garfield dan lucky dinyatakan sebagai tipe Animal dan kemudian dipanggil metode eat() dan makeSound().

4. Abstract Class (AnimalWithChild dan CatWithChild): Kelas abstrak adalah kelas yang tidak dapat diinstansiasi dan bisa memiliki metode abstrak yang harus diimplementasikan oleh kelas-kelas turunannya. Kelas AnimalWithChild adalah kelas abstrak yang memiliki metode abstrak nurse(). Kelas CatWithChild mewarisi

AnimalWithChild dan mengimplementasikan metode nurse(). Ini menunjukkan inheritance dan penggunaan kelas abstrak.

5. Main Class (Main): Kelas Main adalah tempat program dimulai. Di dalamnya, kita membuat objek garfield, lucky, dan susy, dan memanggil metode-metode yang sesuai untuk menunjukkan perilaku dari masing-masing objek tersebut.