

**LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK (PBO)**

**JOBSHEET 5**

**INHERITANCE**



**Oleh :**

**VINCENTIUS LEONANDA PRABOWO**

**2341720149 / 28**

**TI 2A**

**POLITEKNIK NEGERI MALANG  
JURUSAN TEKNOLOGI INFORMASI  
PRODI D-IV TEKNIK INFORMATIKA**

➤ Percobaan 1

Kode :

```
class Hewan {
    String nama;

    public Hewan(String nama){
        this.nama = nama;
    }

    void bersuara(){
        System.out.println("Suara hewan..")
    }
}

class Kucing extends Hewan {
    public Kucing(String nama){
        super(nama);
    }

    void mengeong(){
        System.out.println("Meong!");
    }
}

class Anjing extends Hewan {
    public Anjing(String nama){
        super(nama);
    }

    void menggonggong(){
        System.out.println("Guk! Guk!");
    }
}

public class HewanOOP1 {
    public static void main(String[] args) {
        Kucing kucing = new Kucing("Milo");
        kucing.bersuara();
        kucing.mengeong();

        Anjing anjing = new Anjing("Miluv");
        anjing.bersuara();
        anjing.menggonggong();
    }
}
```

Hasil :

```
run:
Suara hewan...
Meong!
Suara hewan...
Guk! Guk!
```

## ➤ Percobaan 2

Kode :

```
class Kucing extends Hewan {
    public Kucing(String nama) {
        super(nama);
    }

    @Override
    void bersuara() {
        System.out.println("Meong!");
    }

    void mengeong() {
        System.out.println("Meong!");
    }
}

class Anjing extends Hewan {
    public Anjing(String nama) {
        super(nama);
    }

    @Override
    void bersuara() {
        System.out.println("Guk! Guk!");
    }

    void menggonggong() {
        System.out.println("Guk! Guk!");
    }
}
```

Hasil :

```
run:
Meong!
Meong!
Guk! Guk!
Guk! Guk!
```

### ➤ Percobaan 3

Kode :

```
class Hewan {
    String nama;
    int umur;

    public Hewan(String nama, int umur){
        this.nama = nama;
        this.umur = umur;
        System.out.println("Konstruktor Hewan dipanggil");
    }

    void bersuara() {
        System.out.println("Suara hewan...");
    }
}

class Kucing extends Hewan {
    String ras;

    public Kucing(String nama, String ras, int umur){
        super(nama, umur);
        this.ras = ras;
        System.out.println("Konstruktor kucing dipanggil");
    }

    @Override
    void bersuara() {
        System.out.println("Meong!");
    }

    void mengeong() {
        System.out.println("Meong!, Saya " + nama + " ras " + ras);
    }
}

class Anjing extends Hewan {
    String ras;

    public Anjing(String nama, String ras, int umur){
        super(nama, umur);
        this.ras = ras;
        System.out.println("Konstruktor anjing dipanggil");
    }

    @Override
    void bersuara() {
        System.out.println("Guk! Guk!");
    }

    void menggonggong() {
        System.out.println("Guk! Guk!, Saya " + nama + " ras " + ras);
    }
}
```

Hasil :

```
run:
Konstruktor Hewan dipanggil
Konstruktor kucing dipanggil
Meong!
Meong!, Saya Milo ras Persia
Konstruktor Hewan dipanggil
Konstruktor anjing dipanggil
Guk! Guk!
Guk! Guk!, Saya Miluv ras Pudel
```

➤ Percobaan 4

Kode :

```
class Hewan {
    String nama;
    int umur;

    void berjalan(){
        System.out.println("Hewan berjalan");
    }

    public Hewan(String nama, int umur){
        this.nama = nama;
        this.umur = umur;
        System.out.println("Konstruktor Hewan dipanggil");
    }
}

class Mamalia extends Hewan {
    public Mamalia(String nama, int umur){
        super(nama, umur);
    }

    void menyusui(){
        System.out.println("Menyusui anaknya...");
    }
}

class Kucing extends Mamalia {
    public Kucing(String nama, int umur){
        super(nama, umur);
    }
}

class Anjing extends Mamalia {
    public Anjing(String nama, int umur) {
        super(nama, umur);
    }
}

public class HewanOOP1 {
    public static void main(String[] args) {
        Kucing kucing = new Kucing("Milo", 2);
        kucing.berjalan();
        kucing.menyusui();

        Anjing anjing = new Anjing("Miluv", 4);
        anjing.berjalan();
        anjing.menyusui();
    }
}
```

Hasil :

```
run:
Konstruktor Hewan dipanggil
Hewan berjalan
Menyusui anaknya...
Konstruktor Hewan dipanggil
Hewan berjalan
Menyusui anaknya...
```

## ➤ Percobaan 5

Kode :

```
class Hewan {
    private String nama;
    protected int umur;

    void berjalan(){
        System.out.println(nama + "berjalan...");
    }

    public void info(){
        System.out.println("Nama : " + nama);
        System.out.println("Umur : " + umur);
    }

    public Hewan(String nama, int umur){
        this.nama = nama;
        this.umur = umur;
    }

    void bersuara(){
        System.out.println("Suara hewan...");
    }
}

class Kucing extends Hewan {
    private String nama;
    private String warnaBulu;
    String ras;

    public Kucing(String nama, int umur, String warnaBulu){
        super(nama, umur);
        this.warnaBulu = warnaBulu;
    }

    public void info(){
        super.info();
        System.out.println("Warna bulu : " + warnaBulu);
    }

    @Override
    void bersuara() {
        System.out.println("Meong!");
    }

    void mcengeong(){
        System.out.println("Meong!, Saya " + nama + " ras " + ras);
    }
}

class Anjing extends Hewan {
    private String nama;
    private String warnaBulu;
    String ras;

    public Anjing(String nama, int umur, String warnaBulu){
        super(nama, umur);
        this.warnaBulu = warnaBulu;
    }

    public void info(){
        super.info();
        System.out.println("Warna bulu : " + warnaBulu);
    }
}
```

```

@Override
void bersuara() {
    System.out.println("Guk! Guk!");
}

void menggonggong(){
    System.out.println("Guk! Guk!, Saya " + nama + " ras " + ras);
}
}

class Mamalia extends Hewan {
    public Mamalia(String nama, int umur){
        super(nama, umur);
    }

    void menyusui(){
        System.out.println("Menyusui anaknya...");
    }
}

```

Hasil :

```

run:
Nama : Milo
Umur : 2
Warna bulu : Oranye
Miloberjalan...
Nama : Miluv
Umur : 4
Warna bulu : kuning
Miluvberjalan...

```

## ➤ Percobaan 6

Kode :

```
abstract class Kendaraan {
    String merk;
    int tahunProduksi;

    public Kendaraan(String merk, int tahunProduksi) {
        this.merk = merk;
        this.tahunProduksi = tahunProduksi;
    }

    abstract void jalankan();

    void info() {
        System.out.println("Merk: " + merk);
        System.out.println("Tahun produksi: " + tahunProduksi);
    }
}

class Mobil extends Kendaraan {
    int jumlahPintu;

    public Mobil(String merk, int tahunProduksi, int jumlahPintu) {
        super(merk, tahunProduksi);
        this.jumlahPintu = jumlahPintu;
    }

    @Override
    void jalankan() {
        System.out.println("Mobil " + merk + " berjalan...");
    }

    @Override
    void info() {
        super.info();
        System.out.println("Jumlah pintu: " + jumlahPintu);
    }
}

class Motor extends Kendaraan {
    int kapasitasCC;

    public Motor(String merk, int tahunProduksi, int kapasitasCC) {
        super(merk, tahunProduksi);
        this.kapasitasCC = kapasitasCC;
    }

    @Override
    void jalankan() {
        System.out.println("Motor " + merk + " berjalan...");
    }

    @Override
    void info() {
        super.info();
        System.out.println("Kapasitas CC: " + kapasitasCC);
    }
}
```



```
// Subclass baru Truk
class Truk extends Kendaraan {
    int kapasitasMuatan; // Kapasitas muatan truk dalam ton

    public Truk(String merk, int tahunProduksi, int kapasitasMuatan) {
        super(merk, tahunProduksi);
        this.kapasitasMuatan = kapasitasMuatan;
    }

    @Override
    void jalankan() {
        System.out.println("Truk " + merk + " berjalan dengan muatan...");
    }

    @Override
    void info() {
        super.info();
        System.out.println("Kapasitas muatan: " + kapasitasMuatan + " ton");
    }
}
```

Hasil :

```
run:
Merk: Toyota
Tahun produksi: 2022
Jumlah pintu: 4
Mobil Toyota berjalan...
-----
Merk: Honda
Tahun produksi: 2023
Kapasitas CC: 150
Motor Honda berjalan...
-----
Merk: Isuzu
Tahun produksi: 2021
Kapasitas muatan: 10 ton
Truk Isuzu berjalan dengan muatan...
```

#### ➤ Pertanyaan

1. Pada program latihan pertama, jelaskan bagaimana kelas Kucing memanfaatkan konsep inheritance dari kelas Hewan. Sebutkan atribut dan metode apa saja yang diwarisi oleh Kucing.  
Jawab : Kelas Kucing mewarisi atribut nama dan metode bersuara() dari kelas Hewan. Kucing dapat menggunakan metode dan atribut yang sama dengan hewan.
2. Pada program latihan kedua, apa yang dimaksud dengan method overriding dan bagaimana cara mengimplementasikannya? Berikan contoh dari program tersebut.  
Jawab : Method overriding adalah teknik mengganti implementasi metode di subclass. Pada contoh, metode bersuara() di kelas Anjing diganti agar outputnya menjadi "Guk!".
3. Pada program latihan ketiga, mengapa kita perlu menggunakan keyword super dalam konstruktor subclass? Jelaskan bagaimana super digunakan dalam program tersebut.  
Jawab : Keyword super digunakan untuk memanggil konstruktor dari superclass. Ini diperlukan untuk menginisialisasi atribut yang dimiliki superclass.

4. Pada program latihan keempat, jelaskan perbedaan antara single inheritance dan multilevel inheritance. Berikan contoh implementasi keduanya dari program tersebut.

Jawab : Single inheritance berarti subclass hanya memiliki satu superclass, sedangkan multilevel inheritance memungkinkan subclass mewarisi dari subclass lain. Contohnya: Kucing mewarisi dari Mamalia, yang juga mewarisi dari Hewan.

5. Pada program latihan kelima, jelaskan peran access control (public, private, protected) dalam inheritance. Bagaimana cara super digunakan untuk mengakses anggota dari superclass?

Jawab : public, protected, dan private mengontrol akses ke anggota kelas.

Keyword super digunakan untuk mengakses anggota superclass, terutama ketika terjadi overriding atau access control.

6. Pada program latihan keenam, bagaimana konsep abstract class dan method overriding digunakan untuk menciptakan struktur yang fleksibel dalam memodelkan berbagai jenis kendaraan

Jawab : Abstract class menyediakan kerangka dasar dengan method abstract yang harus diimplementasikan oleh subclass. Pada contoh, Kendaraan adalah abstract class, dan Mobil, Motor, dan Truk mengimplementasikan metode jalankan().