

LAPORAN JOBSHEET 6

INHERITANCE

Mata Kuliah : Praktikum Pemrograman Berbasis Objek

Dosen : Irsyad Arif Mashudi, S.Kom., M.Kom



Ilham Dharma Atmaja

24410702020

Kelas :TI 2D

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG TAHUN 2025

1. Tujuan Praktikum

Setelah menempuh materi ini, mahasiswa diharapkan mampu:

1. Memahami konsep dasar

inheritance, termasuk hubungan "is-a" dan bagaimana **subclass** mewarisi atribut dan metode dari **superclass**.

2. Mengimplementasikan inheritance dalam program Java menggunakan *keyword* `extends` dan `super()`.

3. Memahami dan menerapkan

method overriding.

4. Memahami konsep

single dan **multilevel inheritance**.

5. Memahami

access control (`public`, `private`, `protected`) dan penggunaan *keyword* `super` untuk mengakses anggota **superclass**.

6. Menerapkan konsep

abstract class dan **abstract method**.

2. Hasil Percobaan

3.1. Percobaan 1 & Tugas: Dasar-Dasar Inheritance (HewanOOPI)



```
1 class Anjing extends Hewan {
2     public Anjing(String nama) {
3         super(nama); // Memanggil konstruktor superclass Hewan
4     }
5
6     void menggonggong() {
7         System.out.println("Guk...Guuk!");
8     }
9 }
```

3.2. Percobaan 2 & Tugas: Method Overriding

```

1 class Anjing extends Hewan {
2     public Anjing(String nama) {
3         super(nama);
4     }
5
6     @Override
7     void bersuara() {
8         System.out.println("Guk!"); // Override metode bersuara()
9     }
10
11     void menggonggong() {
12         System.out.println("Guk...Guuk!");
13     }
14 }

```

3.3. Percobaan 3 & Tugas: Konstruktor dan super()

```

1 class Kucing extends Hewan {
2     private String ras; // Atribut tambahan untuk subclass
3
4     // Konstruktor untuk Tugas Praktikum
5     public Kucing(String nama, int umur, String ras) {
6         super(nama, umur); // WAJIB menjadi pernyataan pertama: memanggil konstruktor Hewan
7         this.ras = ras;
8         System.out.println("Konstruktor Kucing (dengan ras) dipanggil");
9     }
10
11     // Konstruktor asli Percobaan 3 (tanpa ras)
12     public Kucing(String nama, int umur) {
13         super(nama, umur);
14         System.out.println("Konstruktor Kucing (standar) dipanggil");
15     }
16
17     // Getter untuk ras (karena ras dibuat private, meskipun di soal tidak disebutkan)
18     public String getRas() {
19         return this.ras;
20     }
21 }

```

3.4. Percobaan 4 & Tugas: Multilevel Inheritance

```

1 class Anjing extends Mamalia { // Anjing mewarisi dari Mamalia
2     private String ras;
3
4     public Anjing(String nama, int umur, String ras) {
5         super(nama, umur); // Memanggil konstruktor Mamalia
6         this.ras = ras;
7         System.out.println("Konstruktor Anjing dipanggil");
8     }
9
10    void menggonggong() {
11        System.out.println(nama + " menggonggong: Guk Guk!");
12    }
13
14    public String getRas() {
15        return ras;
16    }
17 }

```

3.5. Percobaan 5 & Tugas: Access Control dan super

```
1 class Anjing extends Hewan {
2     private String jenisBulu; // Private: hanya bisa diakses dari dalam kelas Anjing
3
4     public Anjing(String nama, int i, String string) {
5         super(nama, umur); // Memanggil konstruktor Hewan
6         this.jenisBulu = jenisBulu;
7     }
8
9     public void menggonggong() {
10         // ... (metode tambahan)
11     }
12
13     public String getJenisBulu() { // Public method untuk mengakses atribut private
14         return this.jenisBulu;
15     }
16
17     @Override
18     public void info() {
19         super.info(); // Memanggil Hewan.info() untuk menampilkan nama (private) dan umur (protected)
20         System.out.println("Jenis bulu: " + jenisBulu);
21     }
22
23     public void bersuara() {
24         // TODO Auto-generated method stub
25         throw new UnsupportedOperationException("Unimplemented method 'bersuara'");
26     }
27
28     public void menyusui() {
29         // TODO Auto-generated method stub
30         throw new UnsupportedOperationException("Unimplemented method 'menyusui'");
31     }
32 }
```

3.6. Percobaan 6 & Tugas: Abstract Class

```
1 class Truk extends Kendaraan {
2     int jumlahRoda; // Atribut spesifik Truk
3
4     public Truk(String merk, int tahunProduksi, int jumlahRoda) {
5         super(merk, tahunProduksi);
6         this.jumlahRoda = jumlahRoda;
7     }
8
9     @Override
10    void jalankan() { // Wajib implementasi abstract method
11        System.out.println("Truk " + merk + " berjalan dengan " + jumlahRoda + " roda...");
12    }
13
14    @Override
15    void info() { // Overriding method info()
16        super.info(); // Memanggil info() dari Kendaraan
17        System.out.println("Jumlah roda: " + jumlahRoda);
18    }
19 }
```

4. Link Git Hub

<https://github.com/ianmen10/Pemrograman-Berbasis-Objek/tree/951d0ce8c9c5aefb09b0650fa51e884554a53edc/PRAKTIKUM/Minggu%206>