



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

Laporan Jobsheet 6

Pemrograman Berorientasi Objek (PBO)

Percobaan 1 : Dasar-Dasar Inheritance

- Buat project baru dengan nama **HewanOOP1**

```
1 package HewanOPP1;
2
3 public class Main {
4     public static void main(String[] args) {
5         Kucing kucing = new Kucing("Milo");
6         kucing.bersuara();
7         kucing.mengeong();
8     }
9 }
10
```

- Buat superclass **Hewan**

```
1 package HewanOPP1;
2
3 public class Hewan {
4     String nama;
5
6     public Hewan(String nama){
7         this.nama = nama;
8     }
9
10    void bersuara(){
11        System.out.println("Suara hewan...");
12    }
13 }
14
```

- Buat subclass **Kucing**

```
1 package HewanOPP1;
2
3 public class Kucing extends Hewan{
4     public Kucing(String nama){
5         super(nama);
6     }
7
8     void mengeong(){
9         System.out.println("Meong!");
10    }
11 }
12
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI_1H

- **Tugas Praktikum:** Buatlah subclass lain, misalnya Anjing, yang mewarisi dari Hewan dan memiliki metode menggonggong().

```
1 package HewanOPP1;
2
3 public class Anjing extends Hewan {
4
5     public Anjing(String nama) {
6         super(nama);
7     }
8
9     public void menggonggong() {
10         System.out.println(nama + " menggonggong: Guk guk!");
11     }
12 }
13
```

- **Output :**

```
ednat: java\jdk_ws\week 6_24062024
Suara hewan...
Meong!
Doggy menggonggong: Guk guk!
PS C:\Users\fdlpr\Polinema\Semester 1>
```

Percobaan 2 : Menggunakan Method Overriding

- Perbarui project HewanOOP1

```
1 package HewanOPP1;
2
3 public class Main {
4     public static void main(String[] args) {
5         Kucing kucing = new Kucing("Milo");
6         kucing.bersuara(); // Output : Meong! (karena metode di-override)
7         kucing.mengeong();
8     }
9 }
10
```

- **Superclass Hewan**

```
1 package HewanOPP1;
2
3 public class Hewan {
4     String nama;
5
6     public Hewan(String nama){
7         this.nama = nama;
8     }
9
10    void bersuara(){
11        System.out.println("Suara hewan...");
12    }
13 }
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

- Subclass **Kucing**

```
1 package HewanOPP1;
2
3 public class Kucing extends Hewan{
4     public Kucing(String nama){
5         super(nama);
6     }
7
8     @Override
9     void bersuara(){
10         System.out.println("Meong!");
11     }
12
13     void mengeong(){
14         System.out.println("Meong!");
15     }
16 }
17
```

- Subclass **Anjing**

```
1 package HewanOPP1;
2
3 public class Anjing extends Hewan {
4
5     public Anjing(String nama) {
6         super(nama);
7     }
8
9     public void menggonggong() {
10         System.out.println("Guk..Guuk!");
11     }
12 }
13
```

- **Tugas Praktikum:** Override metode bersuara() di kelas Anjing agar menghasilkan output "Guk!".

```
1 package HewanOPP1;
2
3 public class Anjing extends Hewan {
4
5     public Anjing(String nama) {
6         super(nama);
7     }
8
9     @Override
10    void bersuara(){
11        System.out.println("Guk!");
12    }
13    public void menggonggong() {
14        System.out.println("Guk..Guuk!");
15    }
16 }
17
```

```
Meong!
Meong!
Guk!
Guk..Guuk!
PS C:\Users\fdlpr\Polinema\Sen
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

Percobaan 3 : Konstruktor dan super()

- Buat project baru dengan nama HewanOOP2

```
1 package HewanOOP2;
2
3 public class Main {
4     public static void main(String[] args) {
5         kucing kucing = new Kucing("Milo", 2);
6     }
7 }
```

- Superclass Hewan

```
1 package HewanOOP2;
2
3 public class Hewan {
4     String nama;
5     int umur;
6
7     public Hewan(String nama, int umur){
8         this.nama = nama;
9         this.umur = umur;
10        System.out.println("Konstruktor Hewan dipanggil");
11    }
12
13    void bersuara(){
14        System.out.println("Suara hewan...");
15    }
16 }
```

- Subclass Kucing

```
1 package HewanOOP2;
2
3 public class Kucing extends Hewan{
4     public Kucing(String nama, int umur){
5         super(nama, umur);
6         System.out.println("Konstruktor Kucing dipanggil");
7     }
8 }
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

- Tugas Praktikum: Tambahkan atribut `ras` di kelas `Kucing` dan modifikasi konstruktornya agar menerima parameter `ras`. Pastikan konstruktor `Kucing` memanggil konstruktor `Hewan` dengan benar.

```
1 package HewanOOP2;
2
3 public class Kucing extends Hewan{
4     private String ras;
5
6     public Kucing(String nama, int umur, String ras) {
7         super(nama, umur);
8         this.ras = ras;
9     }
10
11     public void mengeong() {
12         System.out.println("Meong..");
13     }
14
15     public void tampilkanInfo() {
16         System.out.println("Nama: " + nama + ", Umur: " + umur + ", Ras: " + ras);
17     }
18 }
```

```
1 package HewanOOP2;
2
3 public class Main {
4     public static void main(String[] args) {
5         Kucing kucing = new Kucing("Milo", 2, "Persia");
6
7         kucing.tampilkanInfo();
8         kucing.bersuara();
9         kucing.mengeong();
10    }
11 }
```

```
Konstruktor Hewan dipanggil
Nama: Milo, Umur: 2, Ras: Persia
Suara hewan...
Meong..
```

Percobaan 4 : Multilevel Inheritance

- Buat project baru dengan nama **HewanOOP3**

```
1 package HewanOOP3;
2
3 public class Main {
4     public static void main(String[] args) {
5         Kucing kucing = new Kucing("Milo", 2);
6         kucing.berjalan();
7         kucing.menyusui();
8     }
9 }
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

- Superclass **Hewan**

```
1 package Hewan00P3;
2
3 public class Hewan {
4     String nama;
5     int umur;
6
7     public Hewan(String nama, int umur){
8         this.nama = nama;
9         this.umur = umur;
10        System.out.println("Konstruktor Hewan dipanggil");
11    }
12 }
```

- Subclass **Mamalia**

```
1 package Hewan00P3;
2
3 public class Mamalia extends Hewan{
4     public Mamalia(String nama, int umur){
5         super(nama, umur);
6     }
7
8     void menyusui(){
9         System.out.println("Menyusui anaknya..");
10    }
11 }
```

- Sub-Subclass **Kucing**

```
1 package Hewan00P3;
2
3 public class Kucing extends Mamalia{
4     public Kucing(String nama, int umur){
5         super(nama, umur);
6     }
7
8     public void berjalan(){
9         System.out.println( nama + " sedang berjalan...");
10    }
11 }
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

- **Tugas Praktikum:** Buatlah subclass lain, misalnya Anjing, yang mewarisi dari Mamalia. Tambahkan metode atau atribut yang sesuai untuk Anjing.

```
1 package Hewan00P3;
2
3 public class Anjing extends Mamalia {
4     private String jenis;
5
6     public Anjing(String nama, int umur, String jenis) {
7         super(nama, umur);
8         this.jenis = jenis;
9     }
10
11     public void menggonggong() {
12         System.out.println(nama + " menggonggong: Guk! Guk!");
13     }
14
15     public void tampilkanInfo() {
16         System.out.println("Nama: " + nama + ", Umur: " + umur + ", Jenis: " + jenis);
17     }
18 }
```

```
1 package Hewan00P3;
2
3 public class Main {
4     public static void main(String[] args) {
5         Kucing kucing = new Kucing("Milo", 2);
6         kucing.berjalan();
7         kucing.menyusui();
8
9         System.out.println();
10        Anjing anjing = new Anjing("Doggy", 3, "Bulldog");
11        anjing.tampilkanInfo();
12        anjing.menggonggong();
13    }
14 }
```

```
Konstruktor Hewan dipanggil
Milo sedang berjalan...
Menyusui anaknya..
```

```
Konstruktor Hewan dipanggil
Nama: Doggy, Umur: 3, Jenis: Bulldog
Doggy menggonggong: Guk! Guk!
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI_1H

Percobaan 5 : Access Control dan super

- Buat project baru dengan nama **HewanOOP4**

```
1 package HewanOOP4;
2
3 public class Main {
4     public static void main(String[] args) {
5         Kucing kucing = new Kucing("Milo", 2, "Oranye");
6         kucing.info();
7         kucing.berjalan();
8         System.out.println(kucing.nama); // Error, 'nama' adalah private
9         System.out.println(kucing.umur); // Valid, 'umur' adalah protected dan bisa diakses oleh subclass
10    }
11 }
```

- Superclass **Hewan**

```
1 package HewanOOP4;
2
3 class Hewan {
4     private String nama;
5     protected int umur;
6
7     public Hewan(String nama, int umur) {
8         this.nama = nama;
9         this.umur = umur;
10    }
11
12    public void berjalan() {
13        System.out.println(nama + " berjalan...");
14    }
15
16    public void info() {
17        System.out.println("Nama: " + nama);
18        System.out.println("Umur: " + umur);
19    }
20 }
```

- Subclass **Kucing**

```
1 package HewanOOP4;
2
3 class Kucing extends Hewan {
4     private String warnaBulu;
5
6     public Kucing(String nama, int umur, String warnaBulu) {
7         super(nama, umur);
8         this.warnaBulu = warnaBulu;
9     }
10
11    public void info() {
12        super.info(); // Menggunakan 'super' untuk mengakses metode dari parent class
13        System.out.println("Warna bulu: " + warnaBulu);
14    }
15 }
```




Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

- **Tugas Praktikum:** Buatlah subclass lain, misalnya Anjing, yang mewarisi dari Hewan. Tambahkan metode atau atribut yang sesuai untuk Anjing, misalkan atribut jenisBulu yang bersifat private, dan metode getJenisBulu() yang bersifat public untuk mengakses atribut tersebut.

```
1 package Hewan00P4;
2
3 public class Anjing extends Hewan {
4     private String jenisBulu;
5
6     public Anjing(String nama, int umur, String jenisBulu) {
7         super(nama, umur);
8         this.jenisBulu = jenisBulu;
9     }
10
11     public String getJenisBulu() {
12         return this.jenisBulu;
13     }
14
15
16     @Override
17     public void info() {
18         super.info();
19         System.out.println("Jenis Bulu: " + this.jenisBulu);
20     }
21 }
```

Percobaan 6 : Abstract Class

- Buat project baru dengan nama **KendaraanOOP**

```
1 package KendaraanOOP;
2
3 public class Main {
4     public static void main(String[] args) {
5         Mobil mobil = new Mobil("Toyota", 2022, 4);
6         Motor motor = new Motor("Honda", 2023, 150);
7
8         mobil.info();
9         mobil.jalankan();
10
11         System.out.println("-----");
12
13         motor.info();
14         motor.jalankan();
15     }
16 }
17
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

```
Merk : Toyota  
Tahun Produksi : 2022  
Jumlah Pintu : 4  
Mobil Toyota berjalan...  
-----  
Merk : Honda  
Tahun Produksi : 2023  
Kapasitas CC :150  
Motor Honda berjalan
```

- Superclass **Kendaraan**

```
1 package KendaraanOOP;  
2  
3 abstract class Kendaraan {  
4     String merk;  
5     int tahunProduksi;  
6  
7     public Kendaraan(String merk, int tahunProduksi){  
8         this.merk = merk;  
9         this.tahunProduksi = tahunProduksi;  
10    }  
11  
12    abstract void jalankan();  
13  
14    void info(){  
15        System.out.println("Merk : " + merk);  
16        System.out.println("Tahun Produksi : " + tahunProduksi);  
17    }  
18 }
```

- Subclass **Mobil**



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI_1H

```
1 package Kendaraan00P;
2
3 public class Mobil extends Kendaraan{
4     int jumlahPintu;
5
6     public Mobil(String merk, int tahunProduksi, int jumlahPintu){
7         super(merk, tahunProduksi);
8         this.jumlahPintu = jumlahPintu;
9     }
10
11     @Override
12     void jalankan(){
13         System.out.println("Mobil " + merk + " berjalan...");
14     }
15
16     void info(){
17         super.info();
18         System.out.println("Jumlah Pintu : " + jumlahPintu);
19     }
20 }
```

- Subclass **Motor**

```
1 package Kendaraan00P;
2
3 public class Mobil extends Kendaraan{
4     int jumlahPintu;
5
6     public Mobil(String merk, int tahunProduksi, int jumlahPintu){
7         super(merk, tahunProduksi);
8         this.jumlahPintu = jumlahPintu;
9     }
10
11     @Override
12     void jalankan(){
13         System.out.println("Mobil " + merk + " berjalan...");
14     }
15
16     void info(){
17         super.info();
18         System.out.println("Jumlah Pintu : " + jumlahPintu);
19     }
20 }
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI_1H

- **Tugas Praktikum:** Tambahkan subclass lain, misalnya `Truk`, yang mewarisi dari `Kendaraan`. Modifikasi kelas `Kendaraan` dan subclass lainnya sesuai kebutuhan.

```
1 package KendaraanOOP;
2
3 public class Truk extends Kendaraan {
4     double kapasitasMuatan;
5
6     public Truk(String merk, int tahunProduksi, double kapasitasMuatan) {
7         super(merk, tahunProduksi);
8         this.kapasitasMuatan = kapasitasMuatan;
9     }
10
11     @Override
12     void jalankan() {
13         System.out.println("Truk " + merk + " sedang mengangkut barang.");
14     }
15
16     void info() {
17         super.info();
18         System.out.println("Kapasitas Muatan : " + kapasitasMuatan + " Ton");
19     }
20 }
```

```
1 System.out.println("-----");
2
3 truk.info();
4 truk.jalankan();
```

```
-----
Merk : Fuso
Tahun Produksi : 2025
Kapasitas Muatan : 3.0 Ton
Truk Fuso sedang mengangkut barang.
```

Kesimpulan

Inheritance, konsep kunci dalam OOP, memungkinkan penggunaan kembali kode dari superclass, sehingga mengurangi duplikasi dan mempermudah pemeliharaan. Dengan `super`, subclass dapat mengakses anggota superclass dan memanggil konstruktornya.

Access control (`public`, `private`, `protected`) berperan penting dalam mengatur akses ke anggota kelas, menjaga enkapsulasi, dan mencegah perubahan yang tidak diinginkan. Inheritance sangat fleksibel, dapat diterapkan dalam berbagai kasus untuk memodelkan hubungan "is-a" antara kelas-kelas, menghasilkan kode yang lebih terstruktur, efisien, dan mudah dipahami.