

# “ INHERITANCE ”

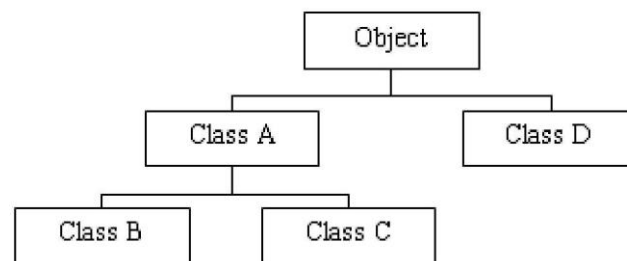
## A. Tujuan

- Memahami Konsep inheritance (pewarisan).
- Mendefinisikan superclasses dan subclasses
- Override method dari superclasses
- Menerapkan keyword this

## B. Dasar Teori

### 1. Pewarisan

*Inheritance* merupakan pewarisan atribut dan method pada sebuah class yang diperoleh dari class yang telah terdefinisi tersebut. Setiap *subclass* akan mewarisi *state* (variabel-variabel) dan *behaviour* (method-method) dari *superclass*-nya. *Subclass* kemudian dapat menambahkan *state* dan *behaviour* baru yang spesifik dan dapat pula memodifikasi (*override*) *state* dan *behaviour* yang diturunkan oleh *superclass*-nya.



Class hierarchy in Java.

Gambar 1. Kelas Hierarki Java

➤ Keuntungan dari inheritance adalah :

*Subclass* menyediakan *state/behaviour* yang spesifik yang membedakannya dengan *superclass*, hal ini akan memungkinkan programmer Java untuk menggunakan ulang *source code* dari *superclass* yang telah ada.

Programmer Java dapat mendefinisikan *superclass* khusus yang bersifat generik, yang disebut *abstract class*, untuk mendefinisikan *class* dengan *behaviour* dan *state* secara umum.

### 2. Key Word Super

Keyword ini digunakan untuk memanggil konstruktor dari superclass atau menjadi variabel yang mengacu pada superclass. Ada beberapa hal yang harus diingat ketika

menggunakan pemanggil constuktor super:

1. Pemanggil super() HARUS DIJADIKAN PERNYATAAN PERTAMA DALAM constructor.
2. Pemanggil super() hanya dapat digunakan dalam definisi constructor.
3. Termasuk constructor this() dan pemanggil super() TIDAK BOLEH TERJADI DALAM constructor YANG SAMA.

Pemakaian lain dari super adalah untuk menunjuk anggota dari superclass (seperti reference **this**).

### 3. Key Word This

Fungsi dari kata kunci THIS adalah : untuk mengakses variable instan dari masing-masing objek.

### 4. Overriding

Method *overriding* adalah mendefinisikan kembali method yang sama, baik nama method maupun signature atau parameter yang diperlukan dalam subclass.

## C. Latihan

### Latihan 1

- a. Tulis ketiga program di bawah ini :

```
class Satu{  
private int a = 5;
```

```
protected void terprotek () {  
System.out.println ("Method ini hanya untuk Anaknya ");  
}
```

```
public void info () {  
System.out.println ("a= "+a);  
System.out.println ("Dipanggil pada " + this.getClass ().getName());
```

```
class Dua extends Satu {  
    private int b = 7;  
  
    public void BacaSuper () {  
        System.out.println ("Nilai b : "+b);  
        terprotek();  
        info();  
    }  
}  
  
class TestSatuDua{  
    public static void main (string [] abc) {  
        Dua D2 = new Dua ();  
        D2.BacaSuper ();  
        D2.info ();  
  
        Satu S1 = new Satu ();  
        S1.terprotek ();  
        S1.info ();  
    }  
}
```

b. Kompilasi ketiga program secara berurutan :

c. Hilangkan baris S1.terprotek () dari class TestSatuDua

```
class TestSatuDua{  
    public.static.void.main (string [] abc) {  
        Dua D2 = new Dua ();  
        D2.BacaSuper ();  
        D2.info ();  
  
        Satu S1 = new Satu ();  
        S1.info ();  
    }  
}
```

d. Kompilasi ketiga class tersebut,hasil kompilasi adalah :

e. Hapus semua file yang ber ekstensi.class dari folder.

f. Kompilasi class TestSatuDua saja. Output yang dihasilkan adalah :

g. Jelaskan fungsi dari perintah **this.getClass().getName()**;

## Latihan 2

a. Tulis ketiga file berikut menjadi 3 file yang berbeda :

```
class Person {  
  
private String nama;  
  
private int usia;  
  
// Konstruktor  
  
public Person (String nama, int usia) {  
  
this.nama = nama;  
  
this.usia = usia;  
  
}  
  
// Method  
  
public void info () {  
  
System.out.println ("Nama : "+this.nama);  
  
System.out.println ("Usia : "+this.usia);  
  
}  
  
}  
  
// Akhir kelas person  
  
class Employ extends Person {  
  
private String noPegawai;  
  
//Konstruktor  
  
public Employ (String noPegawai, String nama,int usia) {
```

```
super(nama, usia);
this.noPegawai = noPegawai;
}

// Methode
public void info() {
System.out.println ("No. Pegawai : "+this.noPegawai);
super.info();
}
} // akhir class employ
public class KonstruktorSuperkelas {
public static void main (String [] args) {
Employ programmer1 = new Employ ("123456", "Yeye",18);
programmer1.info();
}
}
```

b. Keluaran dari program di atas adalah :

### Latian 3

- a. Tulis program berikut

```
public class Animal {
    public static void testClassMethod() {
        System.out.println ("The class method in Animal. ");
    }
    public void testInstanceMethod() {
        System.out.println ("The instance method in Animal. ");
    }
}

public class Cat extends Animal {
    public static void testClassMethod() {
        System.out.println ("The class method in Cat.");
    }
    //meng-override method pada class Animal
    public void testInstanceMethod() {
        System.out.println ("The instance method in Cat. ");
    }
}

public static void main (String[] args) {
    Cat myCat = new Cat();
    Animal myAnimal = myCat;
    Animal.testClassMethod();
    myAnimal.testInstanceMethod();
}
}
```

- b. Output program tersebut adalah :

#### Latian 4

- a. Tulis dan simpan code program berikut (perhatikan keyword super pada program)

```
class A {  
    private int a;  
  
    public void setA (int nilai){  
        a = nilai;  
    }  
    public int getA(){  
        return a;  
    }  
    public void tampilkanNilai(){  
        System.out.println("Nilai a: " + getA());  
    }  
}
```

```
class B extends A{  
    private int b;  
  
    public void setB (int nilai){  
        b = nilai;  
    }  
    public int getB(){  
        return b;  
    }  
    //melakukan override terhadap method tampilkanNilai()  
    //yang terdapat pada kelas A  
    public void tampilkanNilai(){  
        super.tampilkanNilai();//memanggil method dalam kelas A  
        System.out.println("Nilai b : " + getB());  
    }  
}
```

```
class DemoOverride2{  
    public static void main (String [] args){
```

```
        B obj = new B();
```

```
        obj.setA(100);  
        obj.setB(200);
```

```
        //akan memanggil method yang terdapat pada kelas B  
        obj.tampilkanNilai();  
    }  
}
```



b. Keluaran dari program tersebut adalah :

#### Latian 5

a. Tulis dan simpan code berikut

```
public class Bicycle {
//the Bicycle class has three fields
public int cadence;
public int gear;
public int speed;
//the Bicycle class has one constructor
public Bicycle(int startcadence, int startspeed, int startgear) {
gear = startgear;
cadence = startcadence;
speed = startspeed;
}
//the Bicycle class has four methods
public void setcadence(int newValue) {
cadence = newValue;}
public void setgear(int newValue) {
gear = newValue;}
public void applyBrake(int decrement) {
speed -= decrement; }
public void speedUp(int increment) {
speed += increment;
}}
```

```
public class MountainBike extends Bicycle {
//the MountainBike subclass adds one field
public int seatHeight;
//the MountainBike subclass has one constructor
public MountainBike(int startHeight, int startcadence, int startspeed, int startgear) {
super(startcadence, startspeed, startgear);
seatHeight = startHeight; }
//the MountainBike subclass adds one method
public void setHeight(int newValue) {
seatHeight = newValue; }
public static void main(String[] abc) {
```

```
System.out.println("Mulai Berjalan");
MountainBike MB = new MountainBike(0,0,0,0);
System.out.println("gear = "+MB.gear);
System.out.println("speed = "+MB.speed);
MB.setgear(1);
MB.speedUp(1);
System.out.println("gear = "+MB.gear);
System.out.println("speed = "+MB.speed);
MB.speedUp(1);
System.out.println("gear = "+MB.gear);
System.out.println("speed = "+MB.speed);
}
}
```

- b. Keluaran dari program tersebut adalah :