

Pewarisan / Inheritance & Polimorfisme

Pewarisan / Inheritance

Pengertian

- Inheritance (pewarisan) → proses pewarisan data dan method dari suatu kelas kepada kelas lain.
- Pewarisan bersifat menyeluruh
- Semua data dan method milik kelas asalnya diturunkan ke kelas baru.
- Kelas yang mewariskan → **Super Class** atau **Kelas Induk**
- Kelas yang diwariskan → **Sub Class** atau **Kelas Anak**
- Misal :
kelas B adalah kelas anak dari kelas A, maka sifat – sifat dari kelas A otomatis akan dimiliki juga oleh kelas B.
- Jika kelas induk merupakan turunan dari kelas lain, maka data dan method yang berasal dari kelas tersebut akan ikut diwariskan pula kepada kelas anaknya.
- Contoh :
 - Kelas C adalah kelas anak dari kelas B
 - Kelas B merupakan kelas anak dari kelas A.
 - Maka sifat – sifat yang diwariskan kelas A ke kelas B juga akan diwariskan ke kelas C.
- Keyword untuk pewarisan : **extends**

Contoh Program

```
class Induk {
private String var1; //hak akses private berarti tidak dapat diakses oleh kls
anak
public int var2; //hak akses public berarti dapat diakses oleh kls anak
//konstruktor
    Induk (){
    }
    //method
    public void cetakData() {
    }
}

class Anak extends Induk{
private String dataAnak; //variabel
//konstruktor
    Anak () {
    }
    public void cetak() {
        cetakData();//method milik Induk
        // ada jg yang memanggil dgn super.cetakData()
    }
}

class DemoPewarisan {
    public static void main(String args[]) {
        Anak anak = new Anak();
        anak.cetak();
        anak.cetakData();
        System.out.println("Demo Pewarisan");
        System.out.println("isi var 2="+anak.var2);
    }
}
```

Output Program:

```
C:\Documents and Settings\suherman\My Documents>javac
DemoPewarisan.java
```

```
C:\Documents and Settings\suherman\My Documents>java DemoPewarisan
Demo Pewarisan
isi var 2=0
```

Keuntungan Pewarisan

- Bersifat reusable
→ Tidak harus menyalin semua data dan method dari suatu kelas jika akan menggunakannya lagi
- Kemudahan dalam *manage* kelas yang memiliki data dan method yang sama
→ Untuk memodifikasi suatu data atau method untuk semua subkelas / kelas anak, maka tidak perlu melakukan perubahan di masing-masing kelas anak melainkan hanya pada kelas induk saja.

Polimorfisme

- Polimorfisme → sesuatu yang memiliki banyak bentuk.
- Dalam pemrograman, polimorfisme dapat diartikan sebagai modul yang memiliki nama sama, namun memiliki behaviour (tingkah laku) yang berbeda sehingga listing code implementasinya juga berbeda

Contoh Program

```
class Hewan {
    private String jenis;
    private int kaki;

    Hewan () {
    }
    Hewan (String jenis, int kaki) {
        this.jenis = jenis;
        this.kaki = kaki;
    }
    public void bersuara() {
        System.out.print("suara hewan "+jenis+" berkaki"+kaki);
    }
}

class Anjing extends Hewan {
    Anjing() {
        super();
    }
    public void bersuara() { //overriding method
        System.out.println("guk guk guk....");
    }
}
```

```
class Kucing extends Hewan {
    Kucing(String jenis, int kaki) {
        super(jenis, kaki);
    }
    public void bersuara() { //overriding method
        super.bersuara();
        System.out.println(" miaowwww....");
    }
}

class DemoPoli {
    public static void main(String args[]) {
        Anjing doggy = new Anjing();
        Kucing tom = new Kucing("Kucing", 4);
        doggy.bersuara();
        tom.bersuara();
    }
}
```

Output Program

C:\Documents and Settings\suherman\My Documents>javac DemoPoli.java

C:\Documents and Settings\suherman\My Documents>java DemoPoli
guk guk guk....
suara hewan Kucing berkaki 4 miaowwww....