

# PRAKTIKUM 10

---

## INHERITANCE 1

---

### A. TUJUAN PEMBELAJARAN

1. Memahami dan menerapkan konsep inheritance dalam pemrograman
2. Memahami pengaksesan member pada parent class
3. Memahami konsep single inheritance
4. Memahami konsep multi level inheritance

### B. DASAR TEORI

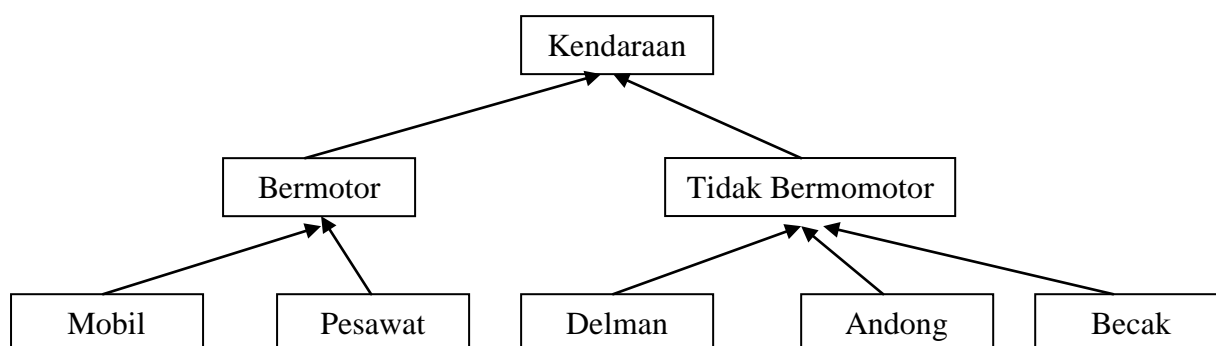
Konsep inheritance ini mengadopsi dunia riil dimana suatu entitas/obyek dapat mempunyai entitas/obyek turunan. Dengan konsep inheritance, sebuah class dapat mempunyai class turunan. Suatu class yang mempunyai class turunan dinamakan parent class atau base class. Sedangkan class turunan itu sendiri seringkali disebut subclass atau child class. Suatu subclass dapat mewarisi apa-apa yang dipunyai oleh parent class-nya, sehingga member dari suatu subclass adalah terdiri dari apa-apa yang ia punyai dan juga apa-apa yang ia warisi dari class parent-nya. Kesimpulannya, boleh dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extend) parent class-nya.

Di dalam Java untuk mendeklarasikan suatu class sebagai subclass dilakukan dengan cara menambahkan kata kunci extends setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class. Berikut adalah contoh deklarasi inheritance. Contoh:

```
public class B extends A {  
    ...  
}
```

Contoh diatas memberitahukan kompiler Java bahwa kita ingin meng-extend class A ke class B. Dengan kata lain, class B adalah subclass (class turunan) dari class A, sedangkan class A adalah parent class dari class B.

Java hanya memperkenankan adanya **single inheritance**. Konsep single inheritance hanya memperbolehkan suatu subclass mempunyai satu parent class. Dengan konsep single inheritance ini, masalah pewarisan akan dapat diamati dengan mudah. Konsep single inheritance bisa dilihat pada Gambar 11.1. Pada gambar tersebut bisa dilihat bahwa tiap class pasti memiliki hanya satu perant class. Java mengijinkan suatu sub class untuk memiliki anak lagi. Hal ini disebut dengan konsep multi level inheritance. Contohnya class Kendaraan memiliki class anak yaitu Bermotor, dan class Bermotor memiliki class anak lagi yaitu Mobil dan Pesawat.



Gambar 11.1 Gambaran konsep single dan multi level inheritance

Dalam konsep dasar inheritance dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extend) parent class-nya. Contoh :

```

public class Pegawai {
    public String nama;
    public double gaji;
}
  
```

```

public class Manajer extends Pegawai {
    public String departemen;
}
  
```

Pada saat class Manajer menurunkan atau memperluas (extend) class Pegawai, maka ia mewarisi data member yang dipunyai oleh class Pegawai. Dengan demikian, class Manajer mempunyai data member yang diwarisi oleh Pegawai (nama, gaji), ditambah dengan data member yang ia punyai (departemen).

Pengaksesan member yang ada di parent class dari subclass-nya tidak jauh berbeda dengan pengaksesan member subclass itu sendiri. Contoh:

```
public class Manajer extends Pegawai {
    public String departemen;

    public void IsiData(String n, String d) {
        nama = n;
        departemen = d;
    }
}
```

### C. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan inheritance?
2. Buatlah contoh kasus yang menerapkan konsep inheritance !
3. Adakah perbedaan cara mengakses member class milik parent dan member class milik sendiri? Jelaskan melalui contoh ! (Silahkan memanfaatkan jawaban soal nomor 2.)
4. Apa yang dimaksud dengan konsep single inheritance ?
5. Apa yang dimaksud dengan konsep multi level inheritance ?

### D. PERCOBAAN

#### Penyembunyian informasi

Buatlah class Pegawai seperti dibawah ini:

```
public class Pegawai {
    private String nama;
    public double gaji;
}
```

Kemudian buatlah class Manajer seperti ini dibawah ini.

```
public class Manajer extends Pegawai {
    public String departemen;

    public void IsiData(String n, String d) {
        nama=n;
        departemen=d;
    }
}
```

Sekarang cobalah untuk mengkompilasi class Manajer diatas. Apa yang terjadi?. Pesan kesalahan akan muncul seperti ini:

```
Manajer.java:5: nama has private access in Pegawai
        nama=n;
```

Ini membuktikan bahwa class Manajer tidak mewarisi data member nama dari parent class-nya (Pegawai).

## E. LATIHAN

**Latihan 1. Tempatkan class Base dan class Class1 di direktori yang sama. Apa yang terjadi ketika Class1.java dikompilasi dan dijalankan jika sebelumnya Base.java belum dikompilasi? Jelaskan !**

```
//Base.java
package Base;
class Base{
    protected void amethod(){
        System.out.println("amethod");
    } //End of amethod
} //End of class base
```

```

package Class1;
//Class1.java
public class Class1 extends Base{
    public static void main(String argv[]){
        Base b = new Base();
        b.amethod();
    }//End of main
} //End of Class1

```

### Latihan 2. Aturan overriding

- a. Berdasarkan kode di bawah ini, akses modifier (public, protected atau private) apa yang diijinkan di tambahkan sebelum myMethod() baris 3?
- b. Jika baris 3 seperti kode di bawah (apa adanya tanpa perubahan) keywords apa yang diijinkan ditambahkan sebelum myMethod baris 8?

```

1.     class HumptyDumpty
2.     {
3.         void myMethod() {}
4.     }
5.
6.     class HankyPanky extends HumptyDumpty
7.     {
8.         void myMethod() {}
9.     }

```

### Latihan 3.

- a. Apa yang terjadi bila kedua kode dibawah ini dikompile dan dijalankan dalam satu direktori? Jelaskan !
- b. Bagaimana solusi supaya tidak terjadi error?

```

//File P1.java
package MyPackage;

class P1{
    void aFancyMethod(){
        System.out.println("What a fancy method");
    }
}

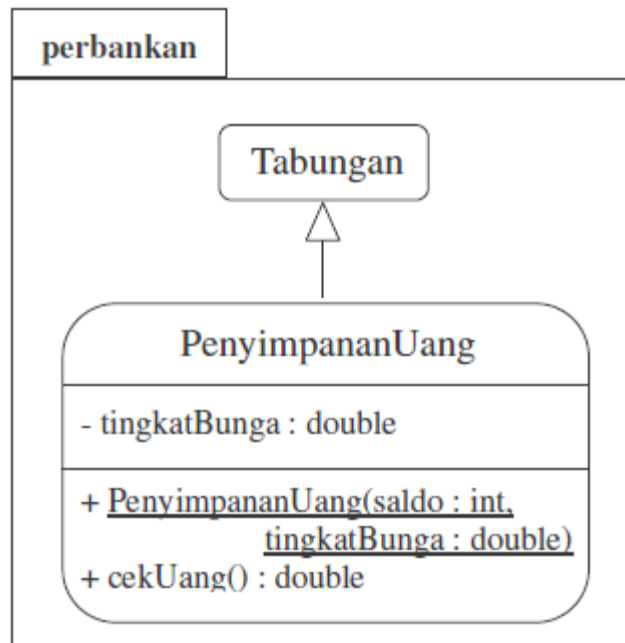
```

```

//File P2.java
public class P2 extends P1{
    public static void main(String argv[]){
        P2 p2 = new P2();
        p2.aFancyMethod();
    }
}

```

**Latihan 4. Mengimplementasikan UML class diagram dalam program untuk package perbankan**



Ubahlah mode akses atribut saldo pada Tabungan menjadi protected. Lalu Transformasikan class diagram diatas ke dalam bentuk program! Tulislah listing program berikut ini sebagai pengetesan.

```
import perbankan.*;

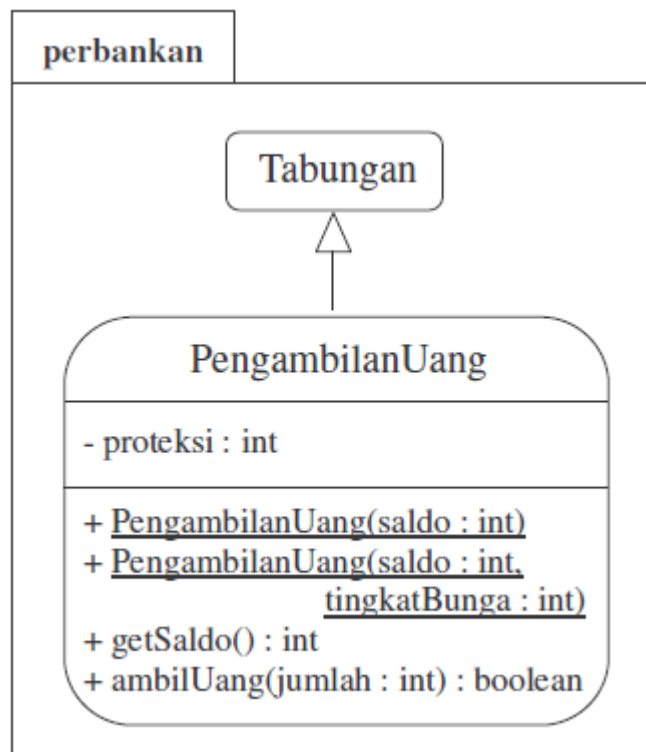
public class TesLatihan{
    public static void main(String args[]){
        PenyimpananUang tabungan = new PenyimpananUang(5000, 8.5/100);
        System.out.println("Uang yang ditabung : 5000");
        System.out.println("Tingkat bunga sekarang : 8.5%");
        System.out.println("Total uang anda sekarang : " +
            tabungan.cekUang());
    }
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```
Uang yang ditabung : 5000
Tingkat bunga sekarang : 8.5%
Total uang anda sekarang : 5425.0
```

## F. TUGAS

**Mengimplementasikan UML class diagram dalam program untuk package perbankan**



Transformasikan class diagram diatas ke dalam bentuk program! Tulislah listing program berikut ini sebagai pengetesan.

```
import perbankan.*;

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

```

    PengambilanUang tabungan = new PengambilanUang(5000, 1000);
    System.out.println("Uang yang ditabung : 5000");
    System.out.println("Uang yang diproteksi : 1000");
    System.out.println("-----");
    System.out.println("Uang yang akan diambil : 4500 " +
        tabungan.ambilUang(4500));
    System.out.println("Saldo sekarang : " + tabungan.getSaldo());
    System.out.println("-----");
    System.out.println("Uang yang akan diambil : 2500 " +
        tabungan.ambilUang(2500));
    System.out.println("Saldo sekarang : " + tabungan.getSaldo());
}
}

```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```

Uang yang ditabung : 5000
Uang yang diproteksi : 1000
-----
Uang yang akan diambil : 4500 false
Saldo sekarang : 5000
-----
Uang yang akan diambil : 2500 true
Saldo sekarang : 2500

```

## G. LAPORAN RESMI

Kumpulkan hasil latihan dan tugas di atas. Tambahkan analisa dalam laporan resmi.