

# **SIFAT PEWARISAN**



**Disusun Oleh**

**Nama : Bening Kusumahati**  
**NPM : 140810160044**

**TEKNIK INFORMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS PADJADJARAN**

# BAB I

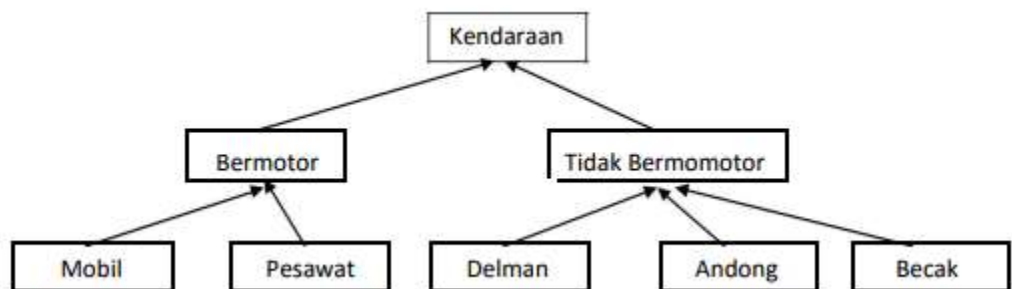
## PENDAHULUAN

### 1.1 TUGAS PENDAHULUAN

- **PENGERTIAN INHERITANCE**

- 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.
- Java hanya memperkenankan adanya single inheritance. Konsep single inheritance hanya memperbolehkan suatu subclass mempunyai satu parent class.

- **CONTOH KONSEP INHERITANCE**



- 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.

- **PERBEDAAN CARA MENGAKSES MEMBER CLASS MILIK PARENT DAN MILIK SENDIRI**

- Pengaksesan member class milik parent:

```
public class Pegawai {  
    public String nama;  
    public double gaji;  
}  
  
public class Manajer extends Pegawai {  
    public String departemen;  
}
```

- Pengaksesan member class milik sendiri:

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

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

- **PENGERTIAN KONSEP SINGLE INHERITANCE**

- Java hanya memperkenalkan 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. Contoh konsep single inheritance dapat dilihat pada contoh di poin kedua.

- **PENGERTIAN KONSEP MULTI LEVEL INHERITANCE**

- Pada gambar di poin kedua bisa dilihat bahwa tiap class pasti memiliki hanya satu parent class. Java mengizinkan 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.

- **MODIFIER UNTUK PENGONTROLAN AKSES**

Tabel 5.1. Akses modifier

Modifier	class yang sama	package yang sama	subclass	class manapun
private	√			
default	√	√		
protected	√	√	√	
public	√	√	√	√

- Akses Private pengaksesan class hanya dapat diakses oleh class dimana tipe ini dibuat
- Akses Default adalah tipe yang hanya class dalam package yang sama yang memiliki hak akses terhadap variabel dan methods dalam class.
- Akses Protected mengizinkan class member untuk diakses oleh method dalam class tersebut dan elemen – elemen subclass
- Akses Public adalah class member untuk diakses baik dari dalam dan luar class.

- **KEGUNAAN KATA KUNCI SUPER**

- Kata kunci super dipakai untuk merujuk pada member dari parent class, sebagaimana kata kunci this yang dipakai untuk merujuk pada member dari class itu sendiri.

- **KONSTRUKTOR TIDAK DIWARISKAN**

- Dalam inheritance, member kelas yang diwariskan hanyalah variabel dan method saja. Sedangkan konstruktor parent class tidak diwariskan ke sub class. Tetapi ketika suatu obyek anak dibuat dalam artian ketika konstruktor anak dijalankan maka konstruktor parent class dijalankan terlebih dahulu dan selanjutnya menyelesaikan konstruktor anak

## BAB II

### PERCOBAAN DAN LATIHAN

#### 1.1 HASIL PERCOBAAN

- **PERCOBAAN 1**

Analisa : Penyembunyian informasi

Hasil Percobaan :

```
E:\OOP>javac Pegawai.java

E:\OOP>java Pegawai
Error: Main method not found in class Pegawai, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

- **PERCOBAAN 2**

Analisa : Menggunakan kata kunci super

Hasil Percobaan :

```
E:\OOP>javac NilaiX.java

E:\OOP>java NilaiX
Author: 140810160044 Bening
Nilai x sebagai parameter = 20
Data member x di class Child = 10
Data member x di class Parent = 5
```

- **PERCOBAAN 3**

Analisa : Konstruktor tidak diwariskan

Hasil Percobaan :

```
E:\DOCS\task.bbr\Programming\Java>javac Childx.java
Childx.java:9: error: call to super must be first statement in constructor
    super();
    ^
1 error
```

Seharusnya pemanggilan konstruktor untuk parent class harus dilakukan di baris pertama pada konstruktor subclassnya

```
E:\DOCS\task.bbr\Programming\Java>javac Childx.java

E:\DOCS\task.bbr\Programming\Java>java Childx
Error: Main method not found in class Childx, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

## 1.2 HASIL LATIHAN

- **LATIHAN 1**

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

Hasil :

```
//Base.java
//package Base;

class Base {
    protected void amethod(){
        System.out.println("Author: 140810160044 Bening");
        System.out.println("amethod");
    }
    //end of amethod
}
//end of class Base

//Class1.java
package Class1;

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

```

E:\DOCS\task.bbr\Programming\Java>javac Base.java

E:\DOCS\task.bbr\Programming\Java>javac Class1.java
Class1.java:4: error: cannot find symbol
public class Class1 extends Base {
                        ^
    symbol: class Base
Class1.java:6: error: cannot find symbol
    Base b = new Base();
                ^
    symbol:   class Base
    location: class Class1
Class1.java:6: error: cannot find symbol
    Base b = new Base();
                  ^
    symbol:   class Base
    location: class Class1
3 errors

```

Pembuatan package menjadi penyebab error karena dalam pewarisan sifat dalam 1 direktori tidak perlu menggunakan package. Solusi:

```
//Base.java
```

```
//package Base;
```

```

class Base {
protected void amethod(){
    System.out.println("Author: 140810160044 Bening");
    System.out.println("amethod");
}
//end of amethod
}
//end of class Base

```

```
//Class1.java
```

```
//package Class1;
```

```

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

```

```
E:\OOP>javac Base.java
E:\OOP>javac Class1.java
E:\OOP>java Class1
Author: 140810160044 Bening
amethod
```

## • LATIHAN 2

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

```
class HumptyDumpty {
    void myMethod(){ }
}
Class HankyPanky extends HumptyDumpty {
    void myMethod(){ }
```

Berdasarkan Source code di atas, akses modifier yang diijinkan untuk ditambahkan sebelum myMethod() di baris 3 yaitu Private. Sedangkan, akses modifier yang diijinkan untuk ditambahkan sebelum myMethod() di baris 8 yaitu Public dan Protected.

## • LATIHAN 3

- Apa yang terjadi bila kedua kode di bawah ini dikompile dan dijalankan dalam satu direktori? Jelaskan!
- Bagaimana solusi supaya tidak terjadi error?
- Hasil :

```
Package MyPackage;
```

```
Class P1 {
    Void aFancyMethod(){
        System.out.println("What a fancy method");
    }
}
```

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



```

E:\DOCS\task.bbr\Programming\Java>javac P1.java

E:\DOCS\task.bbr\Programming\Java>javac P2.java
P2.java:1: error: cannot access P1
public class P2 extends P1 {
                ^
    bad class file: .\P1.class
        class file contains wrong class: MyPackage.P1
        Please remove or make sure it appears in the correct subdirectory of the classpath.

```

Seharusnya packagenya dihapus saja atau dibuat folder dengan nama MyPackage dan P1.java dimasukkan ke dalam folder tersebut. Jadi yang benar seperti ini.

```

Class P1 {
    Void aFancyMethod(){
        System.out.println("Author: 140810160044 Bening");
        System.out.println("What a fancy method");
    }
}

```

```

E:\OOP>javac P1.java

E:\OOP>javac P2.java

E:\OOP>java P2
Author: 140810160044 Bening
What a fancy method

```

- **LATIHAN 4**

Analisa : Mengimplementasikan UML class diagram untuk package perbankan  
 Hasil :

package perbankan;

```

public class PenyimpananUang extends Tabungan {
    private double tingkatBunga;
    public PenyimpananUang(int saldo, double tingkatBunga){
        super(saldo);
        this.tingkatBunga = tingkatBunga;
    }
    public double cekUang(){
        return saldo+(saldo*tingkatBunga);
    }
}

```

package perbankan;

```

public class Tabungan {
    protected int saldo;
    public Tabungan(int saldo){
        this.saldo = saldo;
    }
    public int getSaldo(){
        return saldo;
    }
    public void simpanUang(int jumlah){
        saldo+=jumlah;
    }
    public boolean ambilUang(int jumlah){
        saldo-=jumlah;
        if(saldo==0) return true;
        else {
            saldo+=jumlah;
            return false;
        }
    }
}

```

```

import perbankan.Tabungan;
import perbankan.PenyimpananUang;

```

```

public class Test {
    public static void main(String args[]){
        System.out.println("Author: 140810160044 Bening");
        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 sekarang: "+tabungan.cekUang());
    }
}

```

```

E:\OOP\latihan4>javac Test.java

E:\OOP\latihan4>java Test
Author: 140810160044 Bening
Uang yang ditabung: 5000
Tingkat bunga sekarang: 8.5%
Total uang sekarang: 5425.0

E:\OOP\latihan4>

```

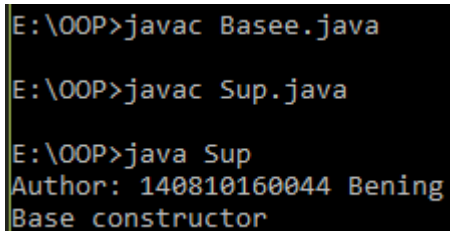
- **LATIHAN 5**

Analisa : Konstruktor tidak diwariskan

Hasil :

```
class Basee {  
    Basee(int i){  
        System.out.println("Author: 140810160044 Bening");  
        System.out.println("Base constructor");  
    }  
    Basee(){  
    }  
}
```

```
public class Sup extends Basee {  
    public static void main(String argv[]){  
        Sup s = new Sup();  
        //baris 1  
    }  
    Sup(){  
        //baris 2  
        super(1);  
    }  
    public void derived(){  
        //baris 3  
    }  
}
```



```
E:\OOP>javac Basee.java  
  
E:\OOP>javac Sup.java  
  
E:\OOP>java Sup  
Author: 140810160044 Bening  
Base constructor
```

- **LATIHAN 6**

Analisa : Konstruktor tidak diwariskan  
Hasil :

```
class Basen {  
    Basen(){  
        int i = 100;  
        System.out.println(i);  
    }  
}  
  
public class Pri extends Basen {  
    static int i = 200;  
    public static void main(String args[]){  
        System.out.println("Author: 140810160044 Bening");  
        Pri p = new Pri();  
        System.out.println(i);  
    }  
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Basen.java  
Basen.java:1: error: modifier private not allowed here  
private class Basen {  
1 error
```

Class yang diprivate tidak bisa digunakan untuk file class lainnya. Seharusnya privatenya dihapus saja.

```
class Basen {  
    Basen(){  
        int i = 100;  
        System.out.println(i);  
    }  
}
```

```
E:\OOP>javac Pri.java  
  
E:\OOP>java Pri  
Author: 140810160044 Bening  
100  
200
```

- **LATIHAN 7**

Analisa : Apa yang tampil di layar bila kode di bawah ini dijalankan?  
Hasil :

```
class X {  
    Y b = new Y();  
    X(){  
        System.out.print("X");  
    }  
}
```

```
class Y {  
    Y(){  
        System.out.print("Y");  
    }  
}
```

```
public class Z extends X {  
    Y y = new Y();  
    Z(){  
        System.out.print("Z");  
    }  
    public static void main (String[] args){  
        new Z();  
    }  
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Z.java  
E:\DOCS\task.bbr\Programming\Java>java Z  
YXYZ
```

- **LATIHAN 8**

Analisa : Kompile dan jalankan program berikut! Apa yang terjadi?  
Jelaskan!

Hasil :

```
public class Hope {  
    public static void main(String argv[]) {  
        Hope h = new Hope();  
    }  
    protected Hope() {  
        for(int i=0;i<10;i++){  
            System.out.println(i);  
        }  
    }  
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Hope.java
```

```
E:\DOCS\task.bbr\Programming\Java>java Hope
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

### 1.3 HASIL TUGAS

- **TUGAS 1**

Analisa : Mengimplementasikan UML class diagram dalam program untuk  
package perbankan  
Hasil :

```
package bank;
```

```
public class Tabungan {  
    int saldo;  
    public Tabungan(int iniSaldo){  
        this.saldo = iniSaldo;  
    }  
}
```

```
package bank;
```

```
public class PengambilanUang extends Tabungan {  
    private int proteksi;  
    public PengambilanUang(int saldo){  
        super(saldo);  
    }  
    public PengambilanUang(int saldo, int proteksi){  
        super(saldo);  
        this.proteksi=proteksi;  
    }  
    public int getSaldo(){  
        return saldo;  
    }  
    public boolean ambilUang(int jumlah){  
        int temp=saldo;  
        saldo-=jumlah;  
        if(saldo<proteksi){  
            saldo=temp;  
            return false;  
        }  
        else return true;  
    }  
}
```

```
import bank.*;
```

```
public class Tugas1{
```

```

public static void main(String args[]){
    System.out.println("Author: 140810160044 Bening\n");
    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());
}
}

```

```

E:\OOP\tugas1>javac Tugas1.java

E:\OOP\tugas1>java Tugas1
Author: 140810160044 Bening

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

```

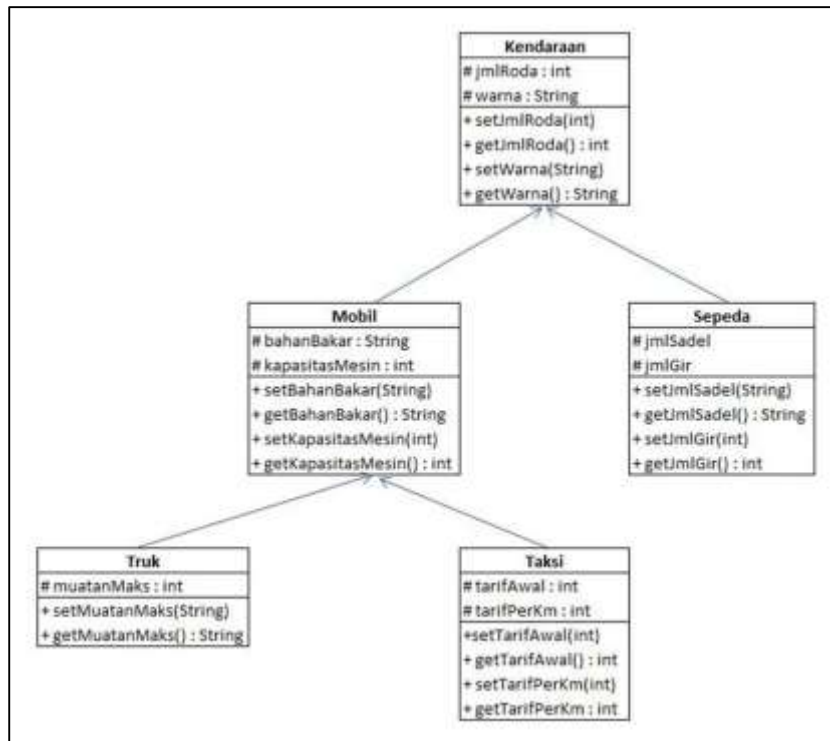


- **TUGAS 2**

Analisa : Mengimplementasikan UML class diagram dalam program

Hasil :

- Buatlah kelas-kelas berdasarkan UML class diagram yang ada!



- Selanjutnya buatlah kelas Tes.java yang membuat obyek-obyek serta mengeset nilai variabel seperti pada Tabel 5.2 dan tampilkan data per obyek.

package transportasi;

```

public class Kendaraan{
    protected int jmlRoda;
    protected String warna;

    public void setJmlRoda(int roda){
        jmlRoda=roda;
    }
    public int getJmlRoda(){
        return jmlRoda;
    }
    public void setWarna(String warna){
        this.warna=warna;
    }
    public String getWarna(){
        return warna;
    }
}
  
```

```
    }  
}
```

```
package transportasi;
```

```
public class Mobil extends Kendaraan {  
    protected String bahanBakar;  
    protected int kapasitasMesin;  
  
    public void setBahanBakar(String bahanBakar){  
        this.bahanBakar=bahanBakar;  
    }  
    public String getBahanBakar(){  
        return bahanBakar;  
    }  
    public void setKapasitasMesin(int kapasitasMesin){  
        this.kapasitasMesin=kapasitasMesin;  
    }  
    public int getKapasitasMesin(){  
        return kapasitasMesin;  
    }  
}
```

```
package transportasi;
```

```
public class Sepeda extends Kendaraan{  
    protected int jmlSadel;  
    protected int jmlGir;  
  
    public void setJmlSadel(int jmlSadel){  
        this.jmlSadel=jmlSadel;  
    }  
    public int getJmlSadel(){  
        return jmlSadel;  
    }  
    public void setJmlGir(int jmlGir){  
        this.jmlGir=jmlGir;  
    }  
    public int getJmlGir(){  
        return jmlGir;  
    }  
}
```

```
package transportasi;
```

```
public class Truk extends Mobil{  
    protected int muatanMaks;  
  
    public void setMuatanMaks(int muatanMaks){  
        this.muatanMaks=muatanMaks;  
    }  
    public int getMuatanMaks(){  
        return muatanMaks;  
    }  
}
```

```
package transportasi;
```

```
public class Taksi extends Mobil{  
    protected int tarifAwal;  
    protected int tarifPerKm;  
    public void setTarifAwal(int tarifAwal){  
        this.tarifAwal=tarifAwal;  
    }  
    public int getTarifAwal(){  
        return tarifAwal;  
    }  
    public void setTarifPerKm(int tarifPerKm){  
        this.tarifPerKm=tarifPerKm;  
    }  
    public int getTarifPerKm(){  
        return tarifPerKm;  
    }  
}
```

```
import transportasi.*;
```

```
public class Tugas2 {  
    public static void main(String args[]){  
        System.out.println("Author: 140810160044 Bening\n");  
        Truk truk1=new Truk();  
        truk1.setJmlRoda(4);  
        truk1.setWarna("Kuning");  
        truk1.setBahanBakar("Solar");  
        truk1.setKapasitasMesin(1500);  
    }  
}
```

```
truk1.setMuatanMaks(5000);
```

```
Truk truk2=new Truk();  
truk2.setJmlRoda(6);  
truk2.setWarna("Merah");  
truk2.setBahanBakar("Solar");  
truk2.setKapasitasMesin(2000);  
truk2.setMuatanMaks(5000);
```

```
Taksi taksi1=new Taksi();  
taksi1.setJmlRoda(4);  
taksi1.setWarna("Oranye");  
taksi1.setBahanBakar("Bensin");  
taksi1.setKapasitasMesin(1500);  
taksi1.setTarifAwal(10000);  
taksi1.setTarifPerKm(5000);
```

```
Taksi taksi2=new Taksi();  
taksi2.setJmlRoda(4);  
taksi2.setWarna("Biru");  
taksi2.setBahanBakar("Bensin");  
taksi2.setKapasitasMesin(1300);  
taksi2.setTarifAwal(7000);  
taksi2.setTarifPerKm(3500);
```

```
Sepeda sepeda1=new Sepeda();  
sepeda1.setJmlRoda(3);  
sepeda1.setWarna("Hitam");  
sepeda1.setJmlSadel(1);  
sepeda1.setJmlGir(2);
```

```
Sepeda sepeda2=new Sepeda();  
sepeda2.setJmlRoda(2);  
sepeda2.setWarna("Putih");  
sepeda2.setJmlSadel(2);  
sepeda2.setJmlGir(5);
```

```
System.out.println("Obyek\tjmlRoda\twarna\tbahanBakar\tKapasitasMesin\tmuatanMaks");
```

```
System.out.println("=====");  
=====);
```



```

        System.out.println("Sepeda2\t"+sepeda2.getJmlRoda()+"\t"+sepeda2.getWarna()
+"\\t"

        +sepeda2.getJmlSadel()+"\\t"+sepeda2.getJmlGir());
    }
}

```

```

E:\OOP\tugas2>javac Tugas2.java

E:\OOP\tugas2>java Tugas2
Author: 140810160044 Bening

```

Obyek	jmlRoda	warna	bahanBakar	KapasitasMesin	muatanMaks	
Truk1	4	Kuning	Solar	1500	5000	
Truk2	6	Merah	Solar	2000	5000	
					tarifAwal	trfPerKm
Taksi1	4	Oranye	Bensin	1500	10000	5000
Taksi2	4	Biru	Bensin	1300	7000	3500
			jmlSadel	jmlGir		
Sepeda1	3	Hitam	1	2		
Sepeda2	2	Putih	2	5		