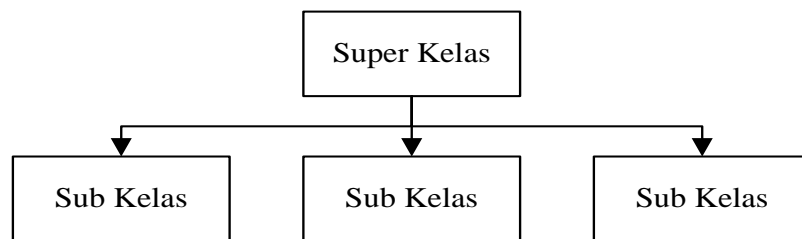


## PEWARISAN (*INHERITANCE*)

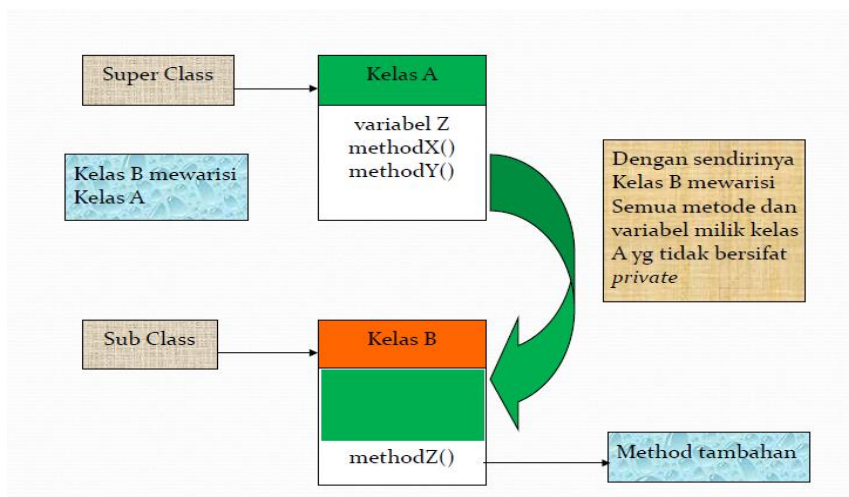
Pewarisan (*inheritance*) dalam bahasa pemrograman Java hampir sama dengan pewarisan yang ada di dunia nyata. Pada dunia nyata, pewarisan dapat diartikan sebagai **penurunan sifat dari suatu objek**.

Pewarisan dalam bahasa pemrograman Java merupakan penurunan sifat dari suatu kelas dan penambahan kapabilitas baru ke dalam kelas tertentu.

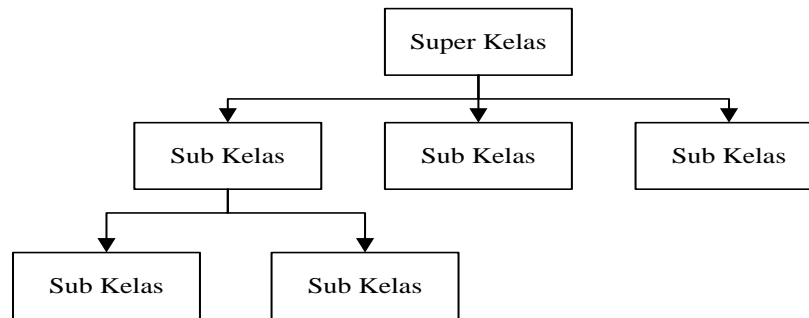
Komponen pada pewarisan terbagi menjadi 2, yaitu **super kelas dan sub kelas**.



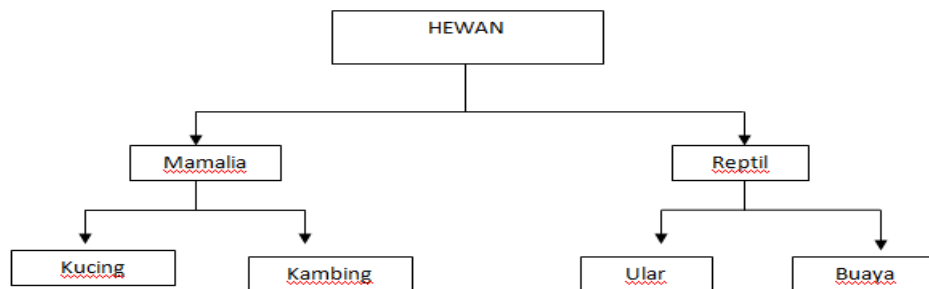
Berikut contoh inheritance Kelas A, dan Kelas B.



Suatu super kelas mewariskan sifatnya ke dalam sub kelas. Sifat yang diwariskan berupa **data dan metode**. Setiap sub kelas dapat terdiri dari data dan metodenya masing-masing, sehingga sub kelas lebih spesifik dari super kelas. Suatu sub kelas dapat menjadi super kelas bagi sub kelas di selanjutnya. Hal tersebut ditunjukkan seperti gambar di bawah ini.



Berikut contoh nya



Kata kunci (*keyword*) yang digunakan untuk menghubungkan antara super kelas dengan sub kelas adalah **extends**.

#### PEWARISAN

```

class SuperKelas{
}
class SubKelas extends SuperKelas{
}
  
```

**IMPLEMENTASI PROGRAM.**

1. membuat program pewarisan kelas Herbivora yang mewariskan sifatnya ke kelas Sapi dan Kambing.

**File: Herbivora.java.**

```
1 -public class Herbivora {  
2     String warna;  
3     int umur;  
4  
5 -    void ciriHerbivora() {  
6         System.out.println("Memakan tumbuhan");  
7     }  
8 }
```

**File: Sapi.java.**

```
1 -public class Sapi extends Herbivora {  
2     double tinggi;  
3     double berat;  
4  
5 -    void ciriSapi() {  
6         System.out.println("Warna sapi : " + warna);  
7         System.out.println("Umur sapi : " + umur);  
8         System.out.println("Tinggi sapi : " + tinggi);  
9         System.out.println("Berat sapi : " + berat);  
10    }  
11 }
```

**File: Kambing.java.**

```
1 -public class Kambing extends Herbivora {  
2     double tinggi;  
3     double berat;  
4  
5 -    void ciriKambing() {  
6         System.out.println("Warna kambing : " + warna);  
7         System.out.println("Umur kambing : " + umur);  
8         System.out.println("Tinggi kambing : " + tinggi);  
9         System.out.println("Berat kambing : " + berat);  
10    }  
11 }
```

File: RunHerbivora.java.

```

1  - public class RunHerbivora {
2  -     public static void main(String args[]) {
3      Sapi sapi = new Sapi();
4
5      sapi.warna = "Putih";
6      sapi.umur = 2;
7      sapi.tinggi = 1.5;
8      sapi.berat = 150;
9      sapi.ciriSapi();
10
11     System.out.println();
12
13     Kambing kambing = new Kambing();
14
15     kambing.warna = "Hitam";
16     kambing.umur = 3;
17     kambing.tinggi = 1;
18     kambing.berat = 70;
19     kambing.ciriKambing();
20 }
21 }
```

### 1. Pewarisan class A dan Class B

```

//demo kelas turunan
class A {
    private int a;
    public int getA(){
        return a;
    }
    public void setA (int nil){
        a= nil;
    }
}
// membuat subclass
class B extends A {
    private int b;
    public int getB(){
        return b;
    }
    public void setB(int nil){
        b= nil;
    }
}
public class CobaKlsTurunan{
    public static void main(String[] args){
        B ob = new B();
        ob.setA(10);
        ob.setB(30);
        System.out.println("Nilai a="+ob.getA());
        System.out.println ("Nilai a="+ ob.getB());
    }
}
```

## 2. Class kakek, bapak, cucu

```

public class Bapak {

    protected String nama, alamat;

    public Kakek ()
    {
        System.out.println ("program demo inheritance");
        System.out.println ("masukkan konstruktor kakek");
        nama="Ahmad";
        alamat="solo";
    }
    public Kakek(String nama,String alamat)
    {
        this.nama=nama;
        this.alamat=alamat;
    }
    public String getNama ()
    {
        return nama;
    }
    public String getAlamat ()
    {return alamat;
    }
}

```

## Class Bapak

```

public class Bapak extends Kakek {

    protected String nama, alamat;

    public Bapak ()
    {
        System.out.println ("Nama kakek:"+ nama);
        System.out.println ("masukkan alamat kakek:"+alamat);
        System.out.println ("masukkan konstrukror bapak");
        nama="Abdul";
        alamat="jogja";
    }
    public Bapak(String nama,String alamat)
    {
        this.nama=nama;
        this.alamat=alamat;
    }
    public String getNama ()
    {
        return nama;
    }
    public String getAlamat ()
    {return alamat;
    }
}

```

Class Cucu . dan merupakan main class

```
public class Cucu extends Bapak
{
    public Cucu ()
    {
        System.out.println ("nama Bapak:" + nama);
        System.out.println ("alamat Bapak:" + alamat);
    }
    public static void main (String [] args)
    {
        String nama;
        Cucu cc=new Cucu();
        System.out.println ("saya adalah cucu");
        System.out.println ("nama saya: Anggun");
    }
}
```

### KATA KUNCI SUPER

Pewarisan dalam bahasa pemrograman Java berhubungan erat dengan kata kunci super. Kata kunci super ini berfungsi untuk: Memanggil suatu variabel atau data dari super kelas, memanggil konstruktor dari super class, memanggil metode dari super class. Berikut contoh memanggil suatu variabel dari super class.

**File: Herbivora1.java.**

```
1 - public class Herbivora1 {
2     String warna = "Putih";
3     int umur = 2;
4 }
```

**File: Sapi1.java.**

```
1 - public class Sapi1 extends Herbivora1 {
2     String warna = "Hitam";
3     int umur = 3;
4
5     void ciriSapi() {
6         System.out.println(super.warna);
7         System.out.println(super.umur);
8     }
9 }
```

**File: RunHerbivora1.java.**

```

1 - public class RunHerbivora1 {
2 -     public static void main(String args[]){
3         Sapi1 sapi = new Sapi1();
4         sapi.ciriSapi();
5     }
6 }

```

Contoh kedua penggunaan kata kunci **Super**

```

class Siswa {
    protected String nama;
    protected String nim;
    public Siswa() {
        nama = ""; nim = ""; }
    public Siswa(String nama, String nim) {
        this.nama = nama; this.nim = nim; }
    public void tampilSiswa() {
        System.out.println("Nama = " + nama);
        System.out.println("Nim = " + nim); } }

class DataSiswa extends Siswa {
    private int umur;
    public DataSiswa(String nama, String nim, int umur) {
        super(nama, nim); //memanggil constructorSiswa
        this.umur = umur; }
    public int getUmur() { return umur; } }

public class DemoTurunanSiswa {
    public static void main(String[] args) {
        DataSiswa ani = new DataSiswa("Ani", "09.01.1111", 18);
        ani.tampilSiswa();
        System.out.println("Umur = " + ani.getUmur()); } }

```

### Method Final Dan Class Final.

Beberapa class dalam Java API dideklarasikan secara final untuk memastikan sifatnya tidak dapat di-*override*. Contoh-contoh dari class ini adalah Integer, Double, dan String. Ini memungkinkan dalam Java membuat method yang tidak dapat di-*override*. Method ini dapat kita panggil **method final**, begitu dengan juga kelas bisa dipanggil **class final**. Untuk mendeklarasikan method untuk menjadi final, kita tambahkan kata **kunci final** ke dalam deklarasi method.

```
public final int getA(){ return a;  
}
```

Dengan memakai kata kunci *final* maka *subclass* yang ingin mendeklarasikan method seperti yang telah dideklarasikan di *superclass*, maka akan terjadi kesalahan. Untuk mendeklarasikan kelas untuk menjadi final, kita tambahkan kata kunci final ke dalam deklarasi kelas.



**LATIHAN MANDIRI**

1. Buatlah program untuk menghitung keliling dan luas lingkaran, buat class Geometri sebagai *superclass* yang menampung variabel r (jari-jari) dan class Lingkaran sebagai *subclass* yang menampung variabel phi. Pada class Lingkaran buat method untuk menghitung keliling dan luas lingkaran, gunakan kata kunci super dalam merefrensi variabel r (jari-jari).

- Keliling Lingkaran =  $2 * \text{phi} * r$

- Luas Lingkaran =  $\text{phi} * r^2$

2. Rancanglah program pewarisan dengan deskripsi sebagai berikut:

a. Super kelas bernama Pegawai dan sub kelas bernama Pegawai Tetap dan Pegawai Kontrak.

b. Di dalam kelas Pegawai terdapat data berupa nama, NIP, dan alamat.

c. Di dalam kelas Pegawai Tetap terdapat data berupa gaji tetap dan tunjangan.

d. Di dalam kelas Pegawai Kontrak terdapat data berupa gaji per hari dan gaji lembur.

e. Hitung gaji Pegawai dengan rumus:

Pegawai Tetap = Gaji Tetap + Tunjangan.

Pegawa Kontrak = Gaji per Hari + Gaji Lembur.