

# Pertemuan 10



## INHERITANCE (PEWARISAN)

# Pengenalan



- Inheritance (Pewarisan) merupakan salah satu dari tiga konsep dasar OOP.
- 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*.
- class turunan itu sendiri seringkali disebut *subclass* atau *child class*.
- Suatu *subclass* dapat mewarisi apa-apa yang dipunyai oleh *parent class*.
- Karena suatu subclass dapat mewarisi apa-apa yang dipunyai oleh parent class-nya, maka 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.**

# Deklarasi Inheritance



- 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.

## Deklarasi inheritance

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



- Semua class di dalam Java adalah merupakan subclass dari class super induk yang bernama **Object**.
- Pada saat dikompilasi Kompiler Java akan membacanya sebagai subclass dari class Object.

```
public class A {  
    .....  
}
```



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

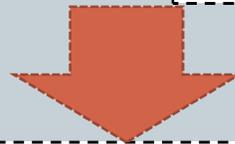
# Penerapan Inheritance



- Penerapan inheritance jika ada suatu class yang dapat diperluas dari class lain

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

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



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

# Keyword “super”



- Kata kunci **super** dipakai untuk merujuk pada member dari parent class.
- Format penulisannya adalah sebagai berikut:
  - `super.data_member`  
merujuk pada data member pada parent class.
  - `super.function_member()`  
merujuk pada function member pada parent class.
  - `super()`  
merujuk pada konstruktor pada parent class.

# Konstruktor tidak diwariskan



- Konstruktor dari parent class tidak dapat diwariskan ke subclass-nya.
- Setiap kali kita membuat suatu subclass, maka kita harus memanggil konstruktor parent class di konstruktor subclass.
- Pemanggilan konstruktor parent harus dilakukan pada baris pertama dari konstruktor subclass.
- Jika kita tidak mendeklarasikannya secara eksplisit, maka kompiler Java akan menambahkan deklarasi pemanggilan konstruktor parent class di konstruktor subclass.
- Sebelum subclass menjalankan konstruktornya sendiri, subclass akan menjalankan constructor superclass terlebih dahulu.
  - Hal ini terjadi karena secara implisit pada constructor subclass ditambahkan pemanggilan `super()` yang bertujuan memanggil constructor superclass oleh kompiler.

# Contoh 1 Program Inheritance



```
class cabang{
    String nama;
    int jmlpem;
    void mulai(){
        System.out.println("pertandingan dimulai");
    }
}
public class bola extends cabang{
    int luaslap;
    void mulai1(){
        super.mulai();
        System.out.println("tendang bola");
    }
    public static void main(String[]args){
        bola cb=new bola();
        cb.mulai1();
    }
}
```

# Contoh 2 Program Inheritance



```
class kelasInduk{
    kelasInduk()
    {
        System.out.println("tampilan ini dari konstruktor kelas induk");
    }
    void methodInduk()
    {
        System.out.println("tampilan ini dari method yg dimiliki oleh kelas induk");
    }
}
class kelasAnak extends kelasInduk{
    kelasAnak(){
        super();
    }
    void methodAnak()
    {
        super.methodInduk();
    }
}
public class inheritancel
{
    public static void main(String [] args)
    {
        System.out.println("tampilan ini dari kelas main method");
        kelasAnak ka=new kelasAnak();
        ka.methodAnak();
    }
}
```

# Contoh 3 Program Inheritance

```
class rupiah{
    int rp;
    rupiah(int rp)
    {    this.rp=rp;    }
    int cetakRp()
    {    return rp;    }
}

class dollar extends rupiah{
    int $,kurs;
    dollar(int $, int rp)
    {    super(rp);
      this.$=$;
      kurs=14000;    }
    int cetak$()
    {    return $;    }
    int cetakkurs()
    {    return $*kurs;    }
}

public class inheritance{
    public static void main(String [] args)    {
        rupiah andi=new rupiah(20000);
        dollar bilal=new dollar(2,30000);
        System.out.println("Uang rupiah andi : "+andi.cetakRp());
        System.out.println("Uang rupiah bilal : "+bilal.cetakRp());
        System.out.println("Uang dolar bilal : "+bilal.cetak$()+"\n");
        System.out.println("Uang dolar bilal jika dikonversi ke rupiah : "+bilal.cetakkurs());
        System.out.println("jumlah rupiah bilal : "+(bilal.cetakkurs()+bilal.cetakRp()));
    }
}
```