



Modul Praktikum **PBO**



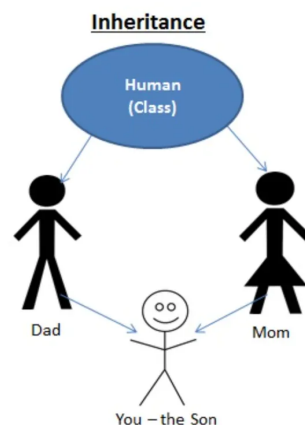
TUJUAN PEMBELAJARAN

- A. Mahasiswa dapat memahami konsep Inheritance
- B. Mahasiswa diharapkan dapat mengimplementasikan Inheritance

DASAR TEORI

1. INHERITANCE

Inheritance merupakan proses **pewarisan data** dan method dari suatu class yang telah ada kepada suatu class baru. Class yang mewariskan disebut dengan **superclass / parent class / base class**, sedangkan class yang mewarisi (class yang baru) disebut dengan **subclass / child class / derived class**. Akan tetapi subclass tidak dapat mewarisi anggota private dari superclass-nya.



Dengan Inheritance, class yang baru (**subclass**) akan mirip dengan class yang lama (**superclass**), namun memiliki karakteristik yang baru. Dalam Java, subclass hanya bisa memiliki satu superclass (single inheritance) sedangkan superclass bisa memiliki satu subclass atau lebih.

Pada Java, untuk mendapatkan properti dari kelas yang lain, kita membuat konsep inheritance itu menjadi 2 yaitu :

- 1. Subclass – Kelas hasil turunan dari kelas lain
- 2. Superclass – kelas yang menurunkan



Penerapan *Inheritance* gunakan statement “extends” :

```
src > data > ParentClass.java > ...  
1  package data;  
2  
3  public class ParentClass {  
4  
5  }  
6  
7  class ChildClass extends ParentClass{  
8  
9  }  
10
```

a. Keyword This dan Super

i. This

Keyword “this” adalah keyword untuk mereferensi *variable/method* yang berada di dalam kelas tersebut. Kata kunci **“this”** sangat berguna untuk menunjukkan suatu member dalam class-nya sendiri. Kata kunci **“this”** dapat digunakan untuk data member, untuk method, dan untuk konstruktor. Adapun format penulisannya adalah

this.data_member => merujuk pada data member

this.nama_method => merujuk pada method

this() => merujuk pada konstruktor

```
src > data > NamaClass.java > ...  
1  package data;  
2  
3  public class NamaClass {  
4      private String variable1;  
5      private int variable2;  
6  
7      public NamaClass(){  
8          this.variable1 = "praktikum PBO";  
9          this.variable2 = 2022;  
10     }  
11 }
```



ii. Super

Keyword “Super” digunakan oleh subclass untuk memanggil konstruktor yang berada pada superclass-nya. **“Super”** adalah keyword untuk membedakan anggota superclass dari subclass, jika mereka memiliki nama yang sama. Bisa juga digunakan untuk memanggil konstruktor superclass dari subclass. Contoh program :

```
src > data > Super.java > Sub > show()
1  package data;
2
3  public class Super {
4      public void desc(){
5          System.out.println(x: "INI KELAS SUPER");
6      }
7  }
8
9  class Sub extends Super{
10     public void desc(){
11         System.out.println(x: "INI ADALAH KELAS SUB");
12     }
13
14     public void show(){
15         this.desc();
16         super.desc();
17     }
18 }
19
```

b. Tipe Inheritance

i. Single Inheritance

Suatu class yang hanya mempunyai satu parent class. Konsep single inheritance hanya memperbolehkan suatu subclass mempunyai satu parent class.

```
src > data > Animal.java > Dog
1  package data;
2
3  public class Animal {
4      void eat(){
5          System.out.println(x: "makan...");
6      }
7  }
8
9  class Dog extends Animal{
10     void barking(){
11         System.out.println(x: "menggonggong...");
12     }
13 }
```



ii. Multi Level Inheritance

Multilevel Inheritance adalah pewarisan class berjenjang dari satu class ke class lain. Konsep multilevel inheritance **memperbolehkan** suatu **subclass mempunyai subclass** lagi. Misalkan, kita memiliki class A yang diturunkan kepada class B. Kemudian class B ini juga diturunkan kepada class C, sehingga sudah terjadi sebuah *multilevel inheritance*.

```
src > data > Animal.java > Dog > barking()
1  package data;
2
3  public class Animal {
4      void eat(){
5          System.out.println(x: "makan....");
6      }
7  }
8
9  class Dog extends Animal{
10     void barking(){
11         System.out.println(x: "mehggonggong...");
12     }
13 }
14
15 class BabyDog extends Dog{
16     void weeping(){
17         System.out.println(x: "merengek...");
18     }
19 }
```

iii. Hierarchical Inheritance

Hierarchical Inheritance adalah konsep dimana child class yang memiliki parent class yang sama. Ketika dua atau lebih kelas mewarisi satu kelas, itu dikenal sebagai inheritance hierarchical.

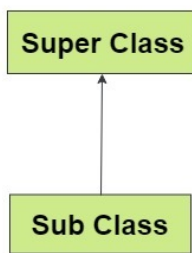
```
src > data > Animal.java > BabyDog
1  package data;
2
3  public class Animal {
4      void eat(){
5          System.out.println(x: "makan....");
6      }
7  }
8
9  class Cat extends Animal{
10     void meow(){
11         System.out.println(x: "Meowwww....");
12     }
13 }
14
15 class Dog extends Animal{
16     void barking(){
17         System.out.println(x: "mengggonggong...");
18     }
19 }
20
21 class BabyDog extends Dog{
22     void weeping(){
23         System.out.println(x: "merengek...");
24     }
25 }
```



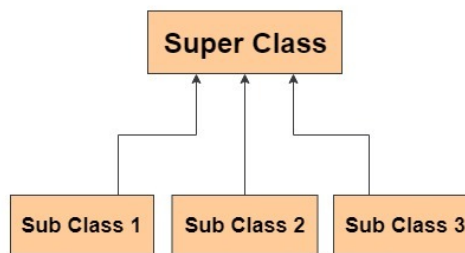
iv. Multiple Inheritance

Multiple Inheritance adalah konsep dimana satu kelas memiliki dua atau lebih parent class.

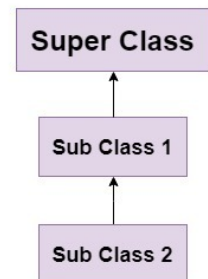
Single Inheritance



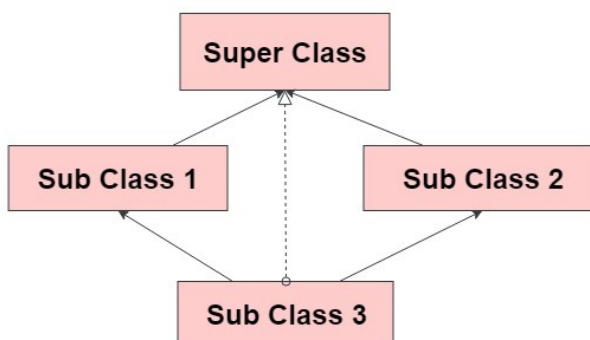
Hierarchial Inheritance



MultiLevel Inheritance



Hybrid Inheritance



Multiple Inheritance

