

INHERITANCE

:: Materi – 4 ::

- ✓ *Inheritance* dapat didefinisikan sebagai proses di mana satu kelas memperoleh properti (metode dan atribut) dari kelas lain.
- ✓ Dengan menggunakan *Inheritance*, informasi dapat dikelola dalam tatanan hierarkis.
- ✓ Kelas yang mewarisi properti lain disebut subkelas (kelas turunan, kelas anak) dan kelas yang propertinya diwarisi disebut superkelas (kelas dasar, kelas induk).

Extends Keyword

SLIDE 3

- ✓ **extends** adalah kata kunci yang digunakan untuk mewarisi properti suatu kelas.
- ✓ Berikut ini adalah sintaks kata kunci extends.

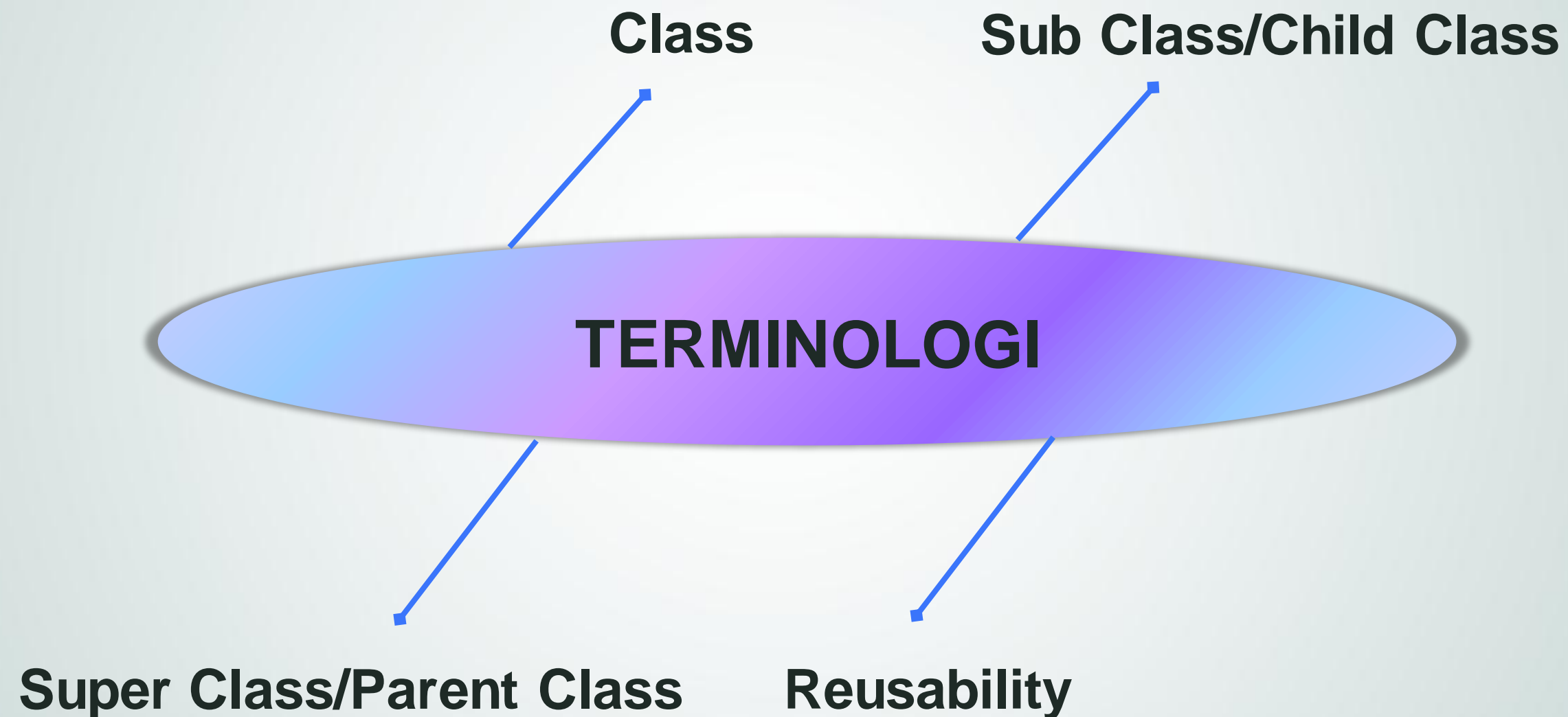
```
class Super {  
    int x = 5;  
}
```

```
class Sub extends Super {  
    int x = 5;  
}
```

- ✓ *Inheritance* mewakili **IS-A relationship** yang juga dikenal sebagai hubungan a *parent-child* relationship.
- ✓ Ide di balik pewarisan di Java adalah Anda dapat membuat kelas baru baru yang dibangun berdasarkan kelas yang sudah ada. Saat mewarisi dari kelas yang sudah ada maka, bisa menggunakan kembali metode dan atribut kelas induk. Selain itu, juga dapat menambahkan metode dan atribut baru di kelas yang dibuat.

Terms used in Inheritance

SLIDE 5



Terms used in Inheritance

SLIDE 6

- ✓ **Class** : sekelompok objek yang memiliki properti umum. Ini adalah templat atau cetak biru dari mana objek dibuat.
- ✓ **Sub Class/Child Class** : kelas yang mewarisi kelas lainnya. Ini juga disebut kelas turunan, kelas diperluas, atau kelas anak.
- ✓ **Super Class/Parent Class** : kelas tempat subkelas yang mewarisi fitur-fiturnya. Ini disebut kelas dasar atau kelas induk.
- ✓ **Reusability** : mekanisme yang memfasilitasi untuk menggunakan kembali atribut dan metode kelas yang ada saat membuat kelas baru. Anda dapat menggunakan atribut dan metode yang sama yang telah ditentukan di kelas sebelumnya.

Syntax of Java Inheritance

SLIDE 7

- ✓ Kata kunci **extends** menunjukkan bahwa Anda membuat kelas baru yang berasal dari kelas yang sudah ada. Arti dari "**extends**" adalah untuk meningkatkan fungsionalitas.
- ✓ Dalam terminologi Java, kelas yang diwarisi disebut induk atau superkelas, dan kelas baru disebut anak atau subkelas.

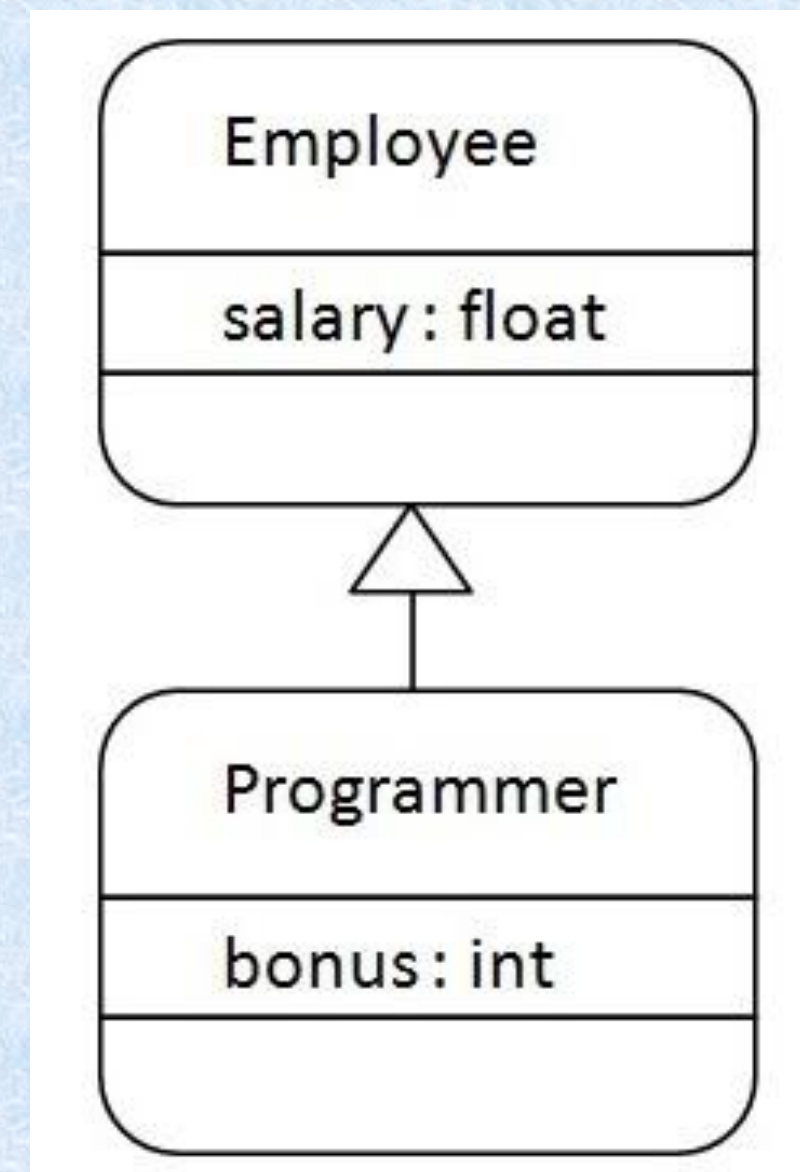
```
class Subclass-name extends Superclass-name {  
    //methods and fields  
}
```


Class Diagram Example

SLIDE 8

Programmer adalah subkelasnya dan Employee adalah superkelasnya. Hubungan antara kedua kelas tersebut adalah **Programmer IS-A Employee**.

Artinya Programmer adalah tipe Karyawan.



Dalam contoh di samping,
objek Programmer dapat mengakses
bidang kelasnya sendiri
serta kelas Karyawan
dapat digunakan kembali.

```
class Employee {  
    float salary = 40000;  
}  
  
class Programmer extends Employee {  
  
    int bonus = 10000;  
  
    public static void main(String args[]) {  
        Programmer p = new Programmer();  
        System.out.println("Programmer salary is:" + p.salary);  
        System.out.println("Bonus of Programmer is:" + p.bonus);  
    }  
}
```

Types of Inheritance

SLIDE 10

Berdasarkan kelasnya, ada tiga jenis pewarisan di java antara lain:



Catatan: *Multiple and hybrid inheritance* tidak didukung di Java melalui kelas. Oleh sebab itu wajib melalui *interface*

Ketika suatu kelas mewarisi kelas lain, hal itu dikenal sebagai *single inheritance*.

Pada contoh di samping, kelas Burung mewarisi kelas Hewan, jadi hanya terdapat warisan tunggal.

```
class Hewan {  
    void eat() {  
        System.out.println("Mematuk makanan...");  
    }  
}  
  
class Burung extends Hewan {  
    void klii() {  
        System.out.println("kli-kli-kli-kli...");  
    }  
}  
  
class TestInheritance {  
    public static void main(String args[]) {  
        Burung br = new Burung();  
        br.klii();  
        br.eat();  
    }  
}
```


Why multiple inheritance is not supported in java

SLIDE 12

- ✓ Untuk mengurangi kompleksitas dan menyederhanakan bahasa, pewarisan berganda tidak didukung di java.
- ✓ Pertimbangan skenario dimana A, B, dan C adalah tiga kelas. Kelas C mewarisi kelas A dan B. Jika kelas A dan B memiliki metode yang sama dan ketika memanggilnya dari objek kelas anak, akan ada ambiguitas dalam memanggil metode kelas A atau B.
- ✓ Karena kesalahan waktu kompilasi lebih baik daripada kesalahan waktu proses, Java membuat kesalahan waktu kompilasi jika kode yang di tulis mewarisi 2 kelas.

Why multiple inheritance is not supported in java

SLIDE 13

```
class A {  
    void msg() {  
        System.out.println("Hello");  
    }  
}  
  
class B {  
    void msg() {  
        System.out.println("Welcome");  
    }  
}  
  
public class C extends A, B{//suppose if it were  
  
    public static void main(String args[]) {  
        C obj = new C();  
        obj.msg();//Now which msg() method would be invoked?  
    }  
}
```

Aggregation in Java

SLIDE 14

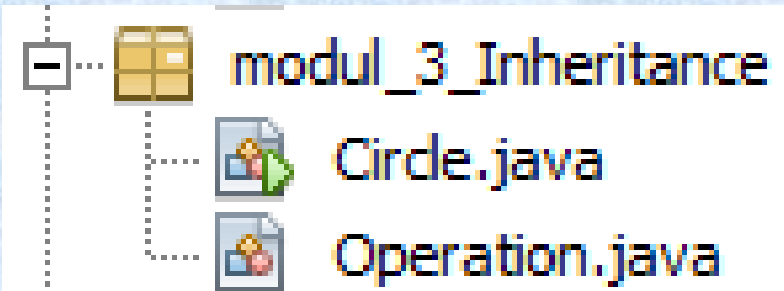
- ✓ Jika suatu kelas memiliki referensi entitas maka akan dikenal sebagai Agregasi
- ✓ Agregasi mewakili hubungan HAS-A

Dalam kasus seperti ini, Karyawan memiliki alamat referensi entitas, sehingga hubungannya adalah ***Employee HAS-A address***

```
class Employee {  
    int id;  
    String name;  
    Address address; //Address is a class  
    ....  
}
```

Example

SLIDE 15



```
public class Operation {  
  
    int square(int n) {  
        return n * n;  
    }  
}
```

Output: 78.5

```
public class Circle {  
  
    Operation op;//aggregation  
    double pi = 3.14;  
  
    double area(int radius) {  
        op = new Operation();  
  
        //code reusability (i.e. delegates the method call)  
        int rsquare = op.square(radius);  
        return pi * rsquare;  
    }  
  
    public static void main(String args[]) {  
        Circle c = new Circle();  
        double result = c.area(5);  
        System.out.println(result);  
    }  
}
```

That's all. Thank you very much! 😊

Any Questions?