



Pemrograman Berorientasi Objek

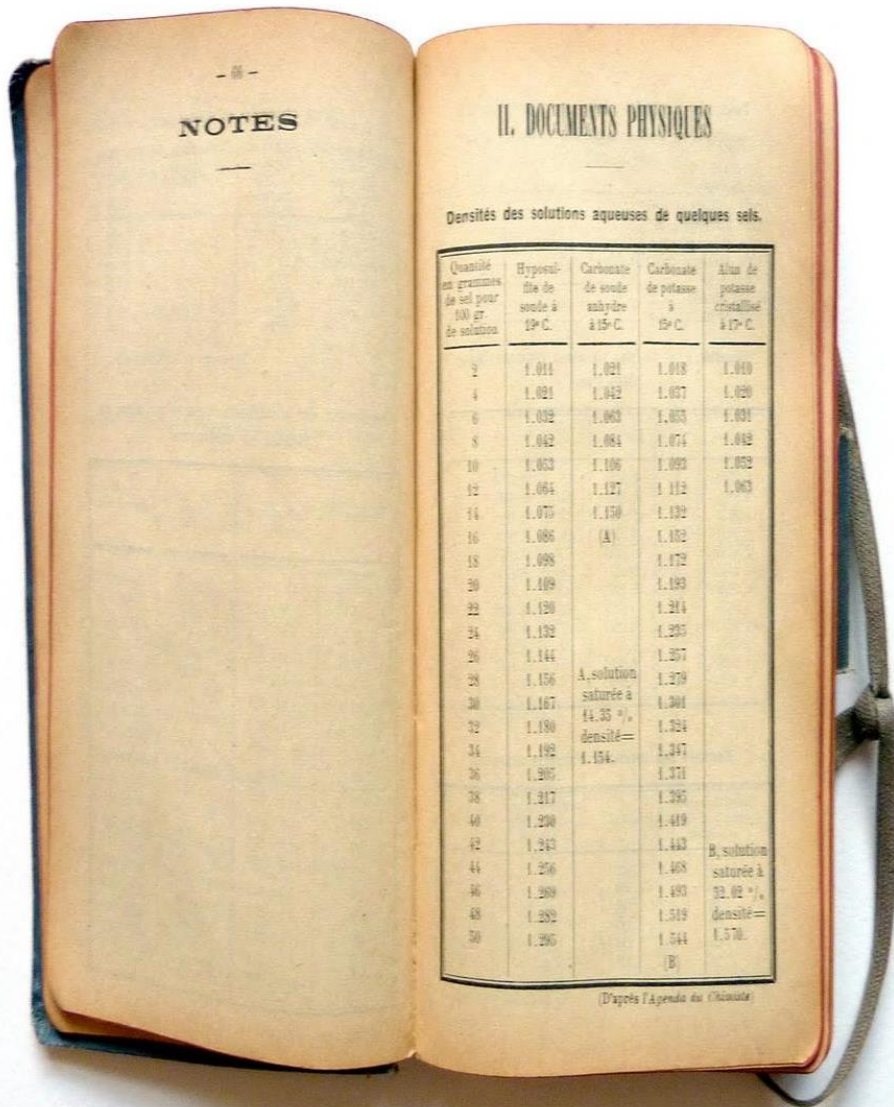


Pertemuan 7

Polymorphism

Pemateri : Chrismikha Hardyanto S.Kom., M.Kom.

KONTEN PERKULIAHAN



- Konsep Dasar Polymorphism
- Menerapkan Polymorphisme pada JAVA
- Method Overriding
- Contoh Kasus Polymorphism pada program JAVA

Setelah kita memahami **Inheritance**, Maka selanjutnya kita lihat konsep PBO lainnya yang erat kaitannya dengan Inheritance yaitu **Polymorphisme**

Konsep Polymorphism

- ❑ **Polimorfisme (Polymorphism)** berasal dari bahasa Yunani yang berarti **"banyak bentuk"** atau **"bermacam-macam bentuk"**
- ❑ Jika dilihat dari **sisi teknis** Pemrograman, Polymorphism adalah kemampuan sebuah object untuk dapat **berubah bentuk menjadi bentuk yang lain**
- ❑ Polymorphism erat kaitannya dengan Inheritance, Karena polymorphism hanya dapat diterapkan **pada class – class turunan** (memiliki hubungan extends)

{OOP}

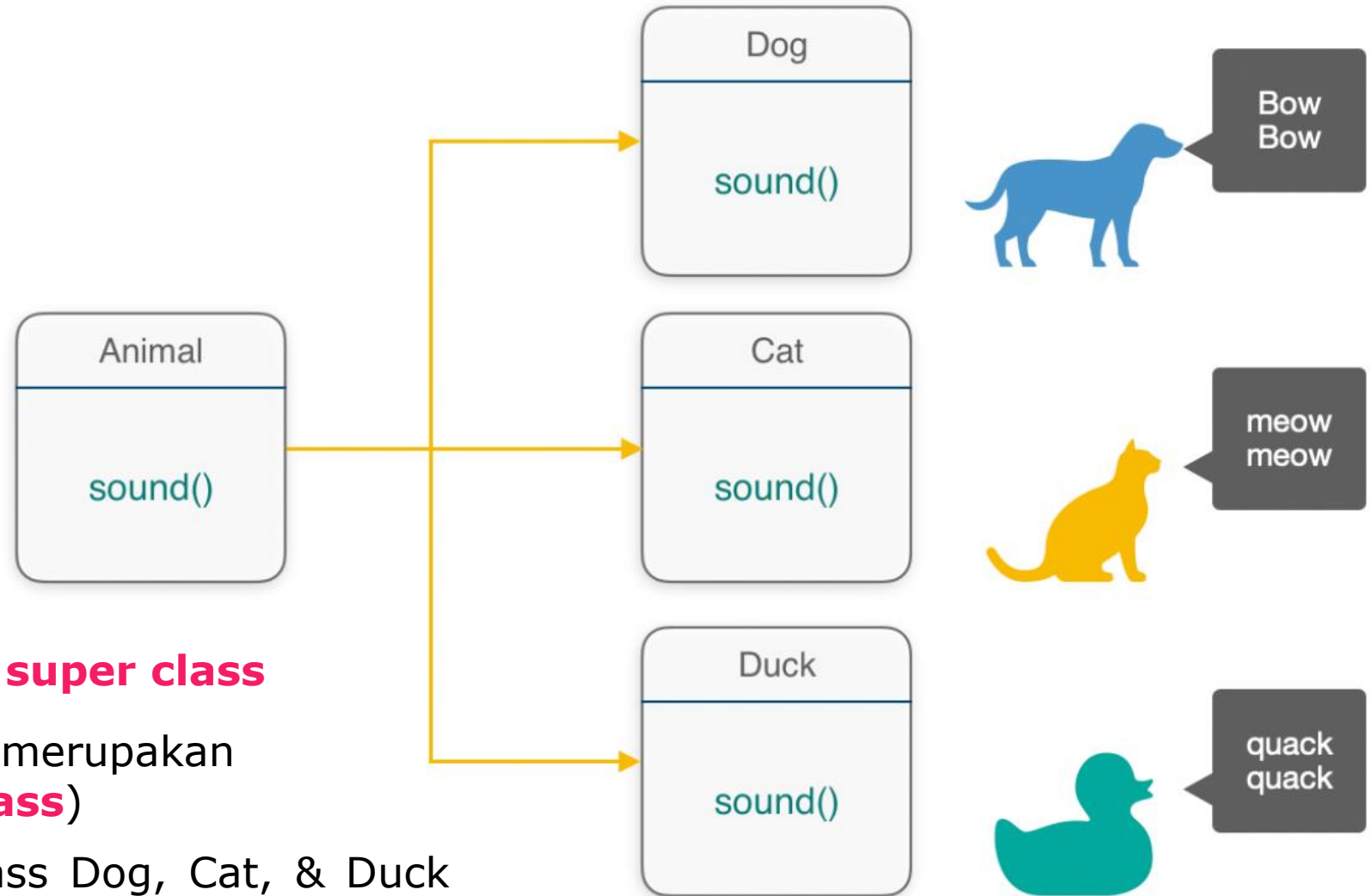
Ilustrasi Polimorphism

- ❑ Sebagai contoh, perhatikan ilustrasi berikut. Kita tahu bahwa Kucing, bebek dan anjing **merupakan hewan (animal)**.
- ❑ Jika hewan **dapat bersuara**, maka Kucing, Bebek, & Anjing pun dapat melakukan **hal yang sama** (karena mereka **adalah hewan**)
- ❑ Akan tetapi suara dihasilkan oleh kucing, bebek, dan anjing **akan berbeda satu sama lain**. Dengan kata lain hewan (animal) mempunyai **banyak bentuk** suara bergantung **jenis hewan apa yg diacu nya**



Ilustrasi Polimorphism

Konsep sebelumnya jika diimplementasikan didalam pemrograman dapat digambarkan sebagai berikut:



- ❑ Class Animal merupakan **super class**
- ❑ Class Dog, Cat, & Duck merupakan **turunan** Animal (**sub class**)
- ❑ Sehingga Object dari class Dog, Cat, & Duck juga **merupakan object dari class Animal**

Ilustrasi Polimorphism

Jika kita tuliskan kode programnya dengan JAVA, maka kita bisa menuliskan instansiasi objectnya seperti ini :

```
1  package tutorial2;
2
3  public class Main {
4
5      public static void main(String[] args) {
6
7          //Deklarasi Object dari masing-masing Class
8          Animal animal = new Animal();
9          Animal dog = new Dog();
10         Animal cat = new Cat();
11         Animal duck = new Duck();
12
13     }
14 }
```

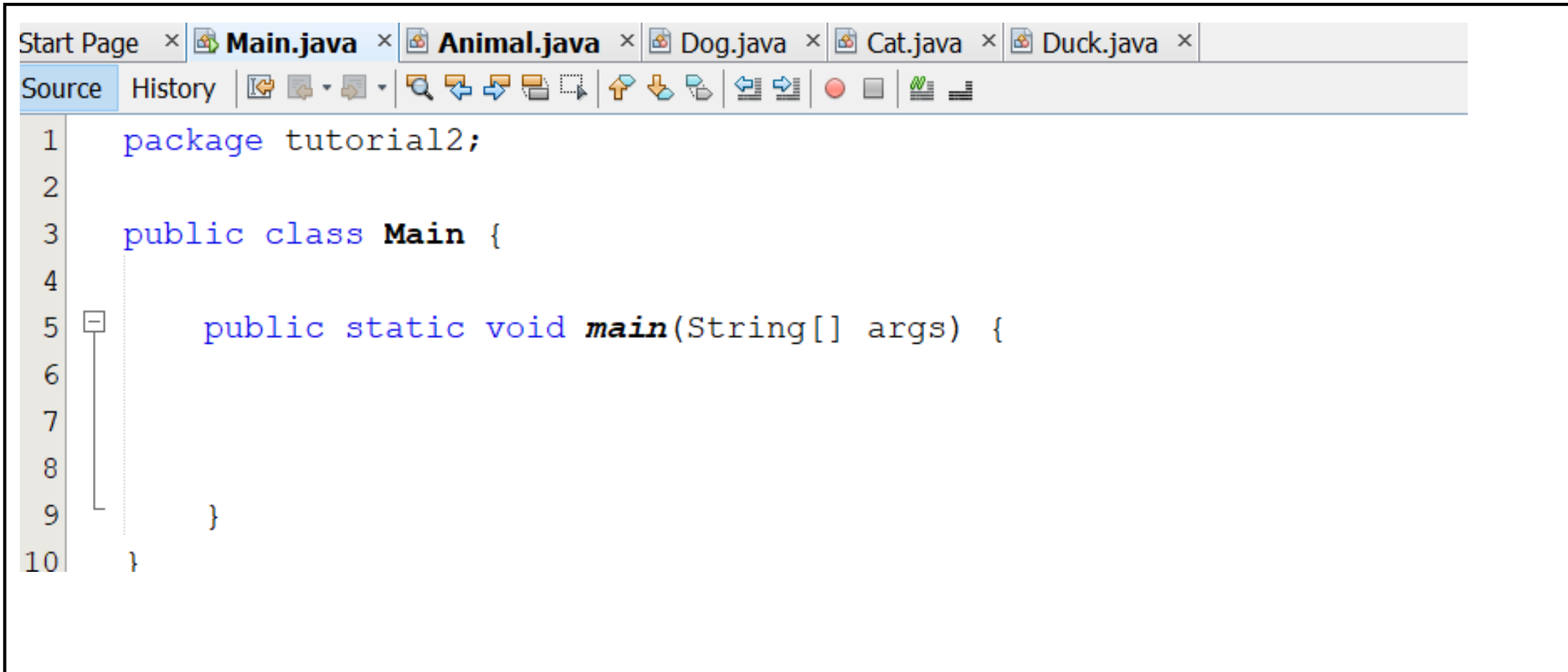
Apakah Ada yang **BERBEDA** Dari penggalan Kode ini ???

*Pendefinisian object diatas baru dapat dilakukan dengan syarat class tersebut sudah menerapkan inheritance didalam kodenya. Inilah yang disebut dengan **polymorphism (satu object punya banyak bentuk)**

Contoh Polymorphism Pada JAVA

Berikut adalah contoh program polymorphism sederhana. Mari kita praktekan :

- ❑ **Langkah 1 :** Buatlah 4 buah class dengan nama Animal, Dog, Cat,& Duck



```
Start Page x Main.java x Animal.java x Dog.java x Cat.java x Duck.java x
Source History | [Icons]
1 package tutorial2;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7
8
9     }
10 }
```


Contoh Polymorphism Pada JAVA

- ❑ **Langkah 2 :** Isilah Class Animal dengan atribut dan method berikut

```
1  package tutorial2;
2
3  public class Animal {
4      //Deklarasi Atribut dari Class Animal
5      private String nama;
6
7      //Deklarasi Konstruktor
8      public Animal() {
9          nama = "...";
10     }
11
12     public String getNama() { ...3 lines }
13
14
15     public void setNama(String nama) { ...3 lines }
16
17
18
19
20     public void sound() {
21         System.out.println(nama+" dapat BERSUARA");
22         System.out.println("");
23     }
24
25 }
```

Contoh Polymorphism Pada JAVA

- ❑ **Langkah 3 :** Isilah Class Dog kode berikut, Lakukan hal yang sama untuk 2 class lainnya

```
1 package tutorial2;
2
3 public class Dog extends Animal {
4
5     public Dog() {
6         super.setNama("DOG");
7     }
8 }
9
```

```
1 package tutorial2;
2
3 public class Duck extends Animal {
4
5     public Duck() {
6         super.setNama("DUCK");
7     }
8 }
9
```

```
1 package tutorial2;
2
3 public class Cat extends Animal {
4
5     public Cat() {
6         super.setNama("CAT");
7     }
8 }
9
```

Contoh Polymorphism Pada JAVA

- ❑ **Langkah 4 :** Buatlah object pada class Main, lalu panggil method sound dari setiap object

```
1  package tutorial2;
2
3  public class Main {
4
5      public static void main(String[] args) {
6
7          //Deklarasi Object dari masing-masing Class
8          Animal animal = new Animal();
9          Animal dog = new Dog();
10         Animal cat = new Cat();
11         Animal duck = new Duck();
12
13         //Panggil Method sound dari masing - masing Object
14         animal.sound();
15         dog.sound();
16         cat.sound();
17         duck.sound();
18
19     }
20 }
```

Output - Tutorial2 (run) ×

```
run:
... dapat BERSUARA

DOG dapat BERSUARA

CAT dapat BERSUARA

DUCK dapat BERSUARA

BUILD SUCCESSFUL (total
|
```

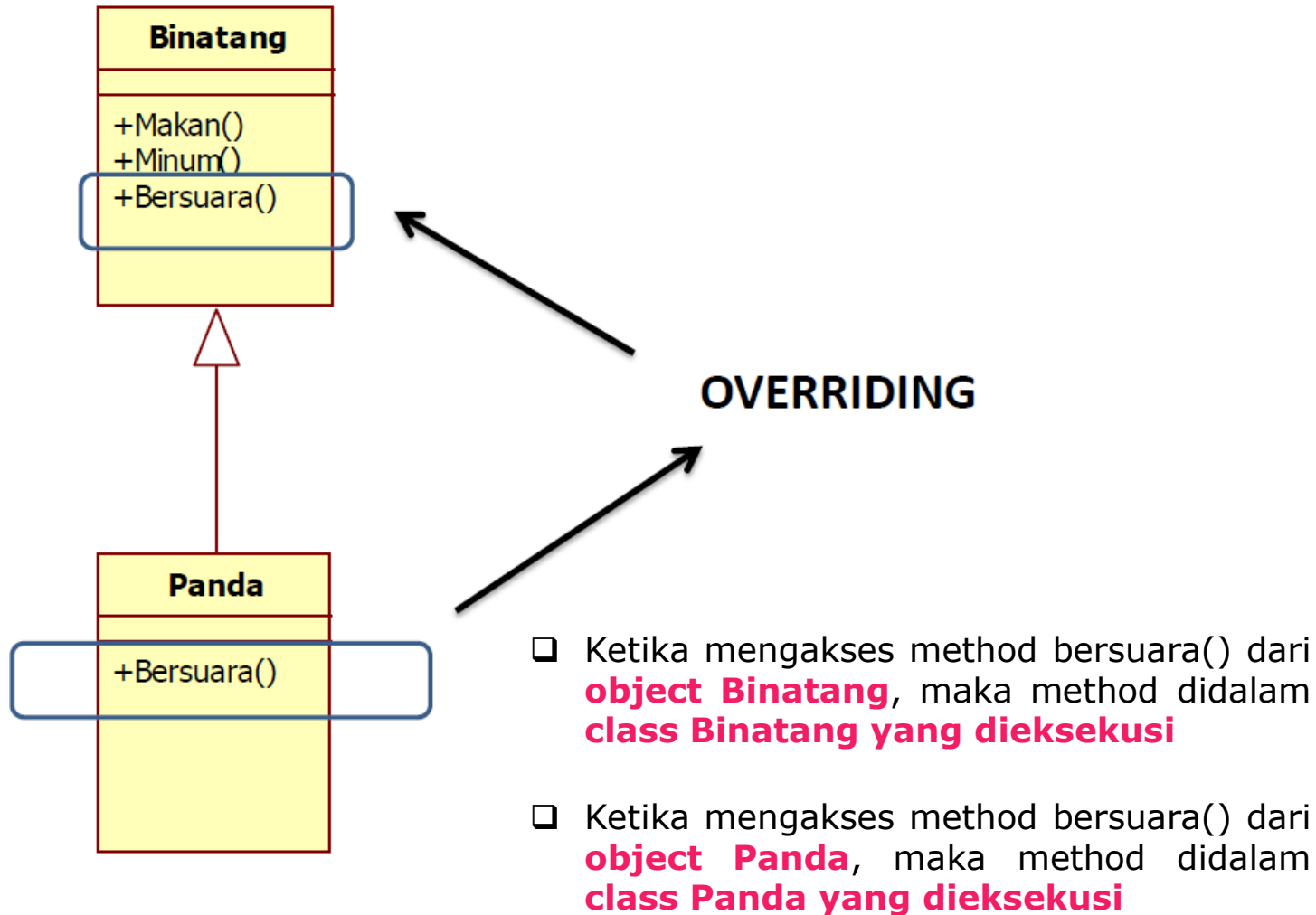
Sampai sini seharusnya kita sudah terbayang bagaimana
Polymorphisme didalam **PBO**
Namun konsep tadi masih **belum lengkap**

Overriding Method

- ❑ **Method overriding** adalah kemampuan **mendeklarasikan/menulis ulang** isi method di child class (sub class), yang sudah ada di parent class (super class).
- ❑ Dengan overriding kita bisa membuat **operasi/aksi yang berbeda** pada method yang sama. Dimana aksi yang dijalankan dilihat dari **object yang mengaksesnya**
- ❑ Saat melakukan proses overriding tersebut, secara otomatis ketika kita membuat object dari child class, method yang ada di class parent **tidak akan bisa diakses lagi**

{OOP}

Overriding Method



{OOP}

Overriding Method

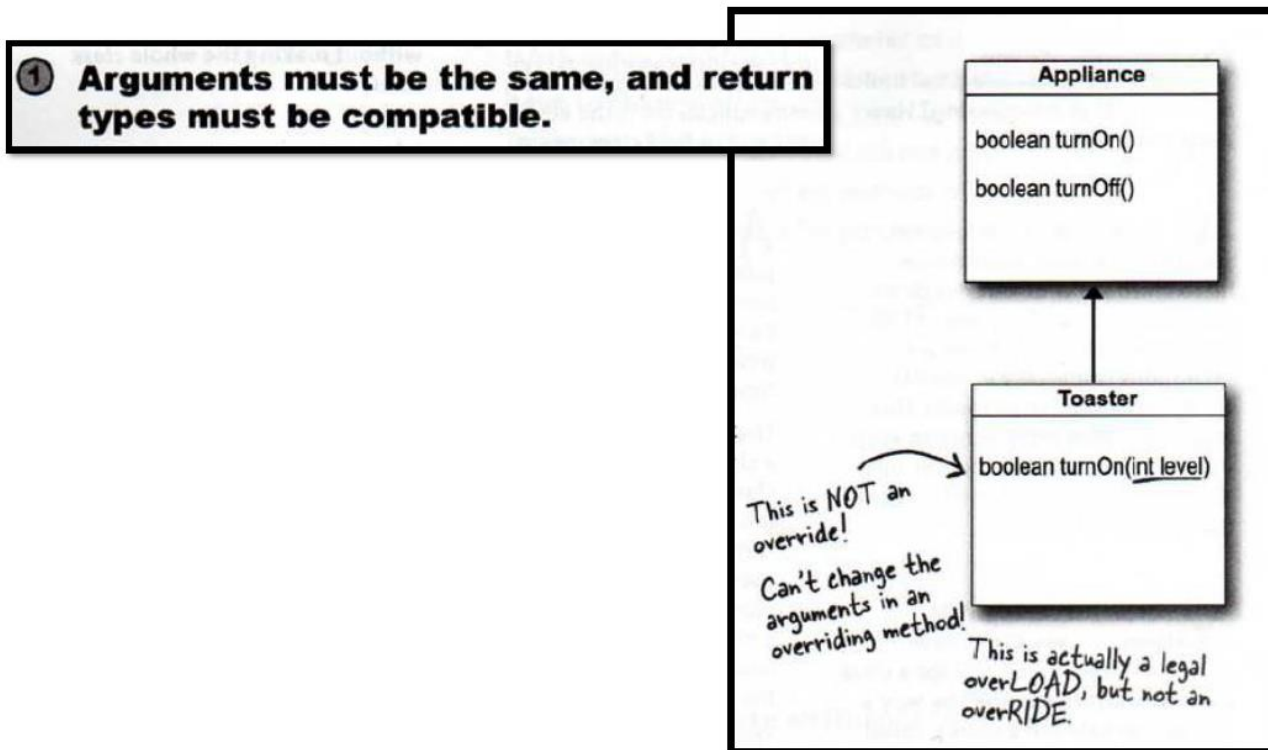
- ❑ Overriding method **berbeda** dengan overloading method

An overloaded method is just a different method that happens to have the same method name. It has nothing to do with inheritance and polymorphism. An overloaded method is NOT the same as an overridden method.

{OOP}

Aturan Overriding

- ❑ Ada **beberapa aturan** yang harus kita ikuti untuk membuat overriding method pada JAVA. Antara lain :

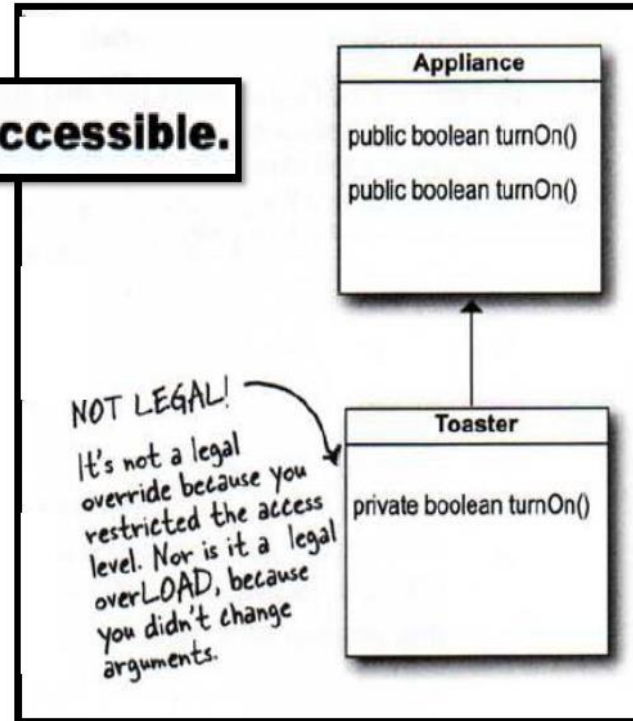


**Pendeklarasian Method pada Sub Class
harus sama dengan Super Class**

{OOP}

Aturan Overriding

② The method can't be less accessible.



Tingkat akses pada method di Sub Class tidak boleh lebih tinggi tingkatnya dari Super Class

{OOP}

Contoh Overriding Pada JAVA

Berikut adalah contoh program Overriding sederhana. Mari kita praktekan dengan program sebelumnya:

- ❑ **Langkah 1** : Buka kembali project sebelumnya di slide ini, lalu tambahkan method `sound()` pada setiap Sub Class dari Animal **sesuai dengan suara dari hewan** tersebut.

```
1  package tutorial2;
2
3  public class Dog extends Animal {
4
5      public Dog() {
6          super.setNama("DOG");
7      }
8
9      @Override
10     public void sound() {
11         System.out.println(super.getNama()+" Bersuara BOW BOW");
12         System.out.println("");
13     }
14
15 }
```

Contoh Overriding Pada JAVA

- ❑ **Langkah 1** : method sound() pada class Cat

```
1  package tutorial2;  
2  
3  public class Cat extends Animal {  
4  
5      public Cat() {  
6          super.setNama("CAT");  
7      }  
8  
9      @Override  
10     public void sound() {  
11         System.out.println(super.getNama() + " Bersuara MEOW MEOW");  
12         System.out.println("");  
13     }  
14 }  
15
```

Contoh Overriding Pada JAVA

- ❑ **Langkah 1** : method sound() pada class Duck

```
1  package tutorial2;
2
3  public class Duck extends Animal {
4
5      public Duck() {
6          super.setNama("DUCK");
7      }
8
9      @Override
10     public void sound() {
11         System.out.println(super.getNama() + " Bersuara QUACK QUACK");
12         System.out.println("");
13     }
14 }
15
```

Contoh Overriding Pada JAVA

- ❑ **Langkah 2 :** Coba run program tersebut (Biarkan class Main seperti sebelumnya), coba lihat output dari program sekarang

```
1  package tutorial2;
2
3  public class Main {
4
5      public static void main(String[] args) {
6
7          //Deklarasi Object dari masing-masing Class
8          Animal animal = new Animal();
9          Animal dog = new Dog();
10         Animal cat = new Cat();
11         Animal duck = new Duck();
12
13         //Panggil Method sound dari masing - masing Object
14         animal.sound();
15         dog.sound();
16         cat.sound();
17         duck.sound();
18     }
19 }
20 }
```

run:

Animal dapat BERSUARA

DOG Bersuara BOW BOW

CAT Bersuara MEOW MEOW

DUCK Bersuara QUACK QUACK

BUILD SUCCESSFUL (total time:

Sekarang program dapat menampilkan suara hewan **yang spesifik** sesuai dengan jenis hewannya (**dengan menggunakan method yg sama**)

Setelah memahami **Materi Hari ini** pada JAVA,
mari kita berlatih

***Silakan buka modul praktikum pertemuan 6&7**

Terima Kasih