

Object, Class & Encapsulation

Pertemuan 3

Dosen Pengampu:

- Ghifari Munawar
- Yadhi Aditya P
- Zulkifli Arsad



Topics

1. Object
2. Class
3. Messages
4. Encapsulation



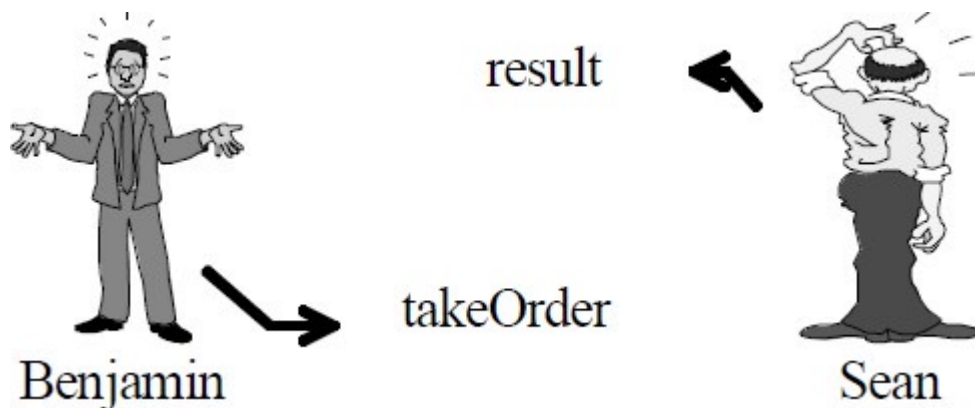
1. Object

- Objek merupakan sesuatu (**tangible** / **intangible**) yang memiliki **state** (attributes) dan **behaviour** (methods).
- Objek disebut juga “**instance of class**” atau representasi dari sebuah kelas.
- Interaksi antar objek dilakukan melalui “**sending message**”.



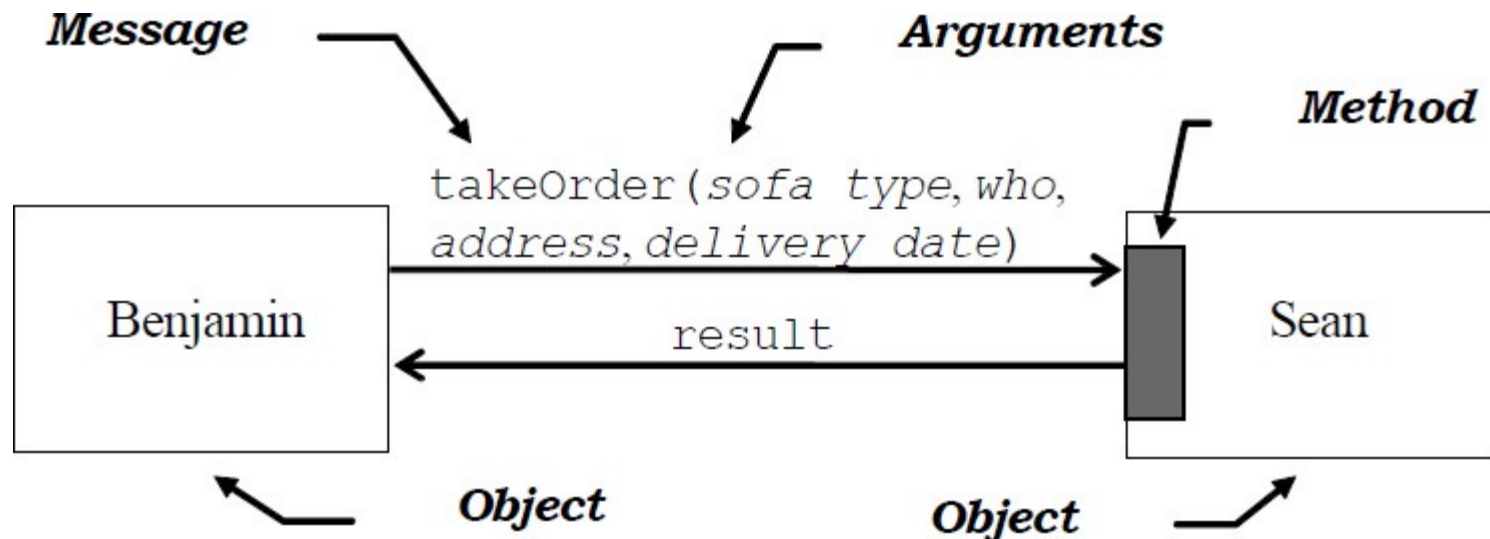
Contoh interaksi antar objek

- Benjamin ingin memesan kursi sofa, 5 seater-set, dan dikirimkan ke JTK pada tanggal 10 September 2024.
- Sean merupakan sales dari toko furniture.
- Bean mengirimkan pesan “**takeOrder**”, lalu Sean akan mengembalikan “**result**” sesuai pesanan.



Contd..

- Benjamin ingin memesan kursi sofa, 5 seater-set, dan dikirimkan ke JTK pada tanggal 10 September 2024
- Pesan : **takeOrder**(sofa type, who, address, delivery date)



Object characteristics

- Objects Behavior - What can you do with this object, or what method can you apply it ?
- Objects State - How does the object react when you invoke those method ?
- Objects Identity - How is the object distinguished from others ?



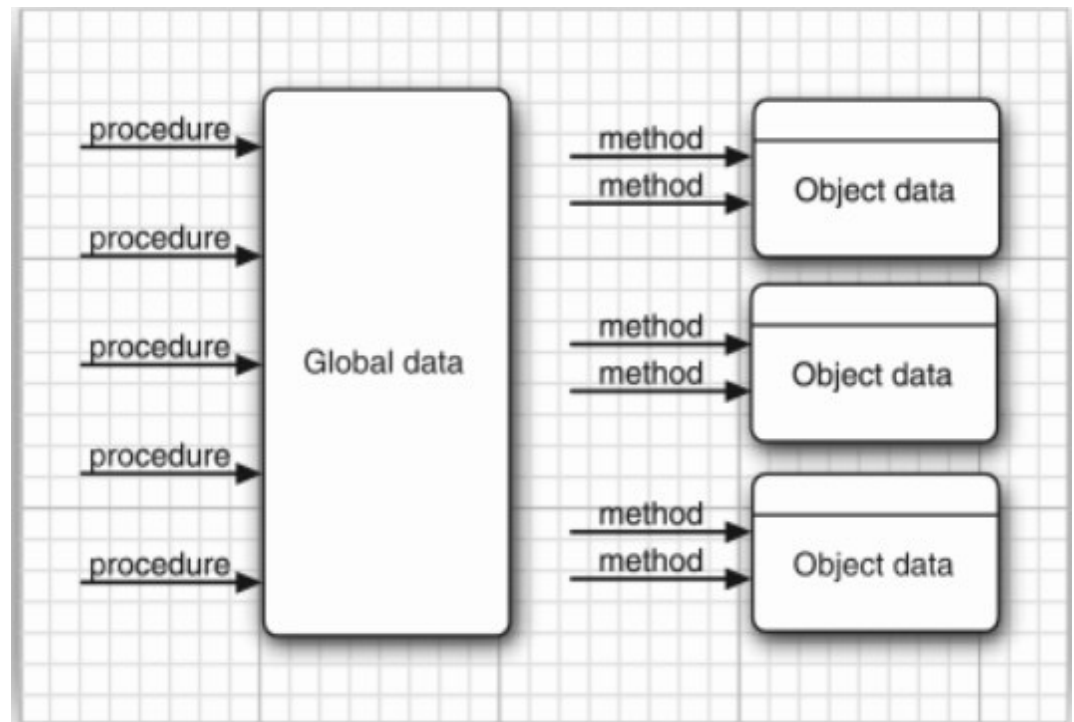
2. Class

- Class merupakan suatu **blueprint** dari objek.
- Class is a “**definition of template for structuring and creating objects with the same attributes and same methods**”.



Prosedural vs OOP

- Problem Solving
- Maintainability
- Modularity

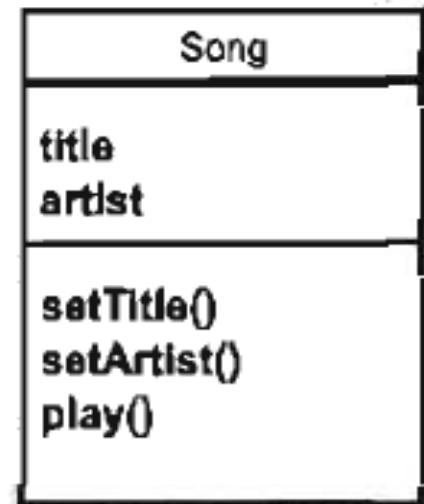


Contd..

- Struktur kelas terdiri dua, yakni
 1. State (instance variables)
 2. Behaviour (methods)

**instance
variables**
(state)

methods
(behavior)



Contoh struktur objek :

Benjamin as an Object

Attributes:

```
name = "Benjamin"
address = "1, Robinson Road"
budget = "2000"
```

Methods:

```
purchase()    {send a purchase request to a salesperson}
getBudget()   {return budget}
```

Bernie as an Object

Attributes:

```
name = "Bernie"
address = "18, Sophia Road"
budget = "1000"
```

Methods:

```
purchase()    {send a purchase request to a salesperson}
getBudget()   {return budget}
```



Contoh desain kelasnya :

Class Customer

Attributes:

name

address

budget

Methods:

purchase () {send a purchase request to a salesperson}

getBudget () {return budget}



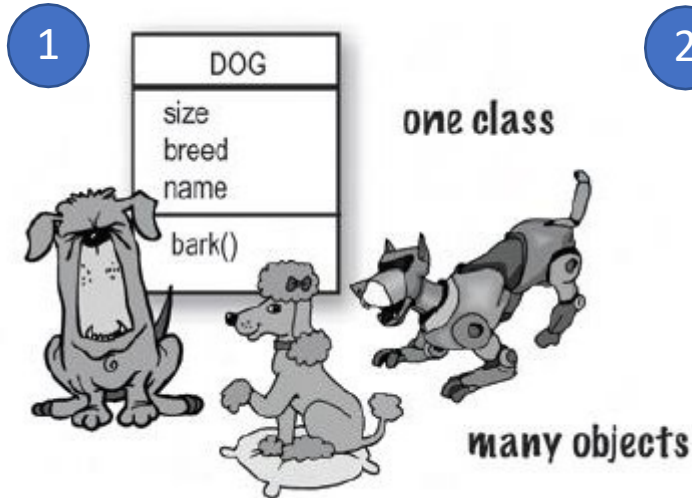
Identifying Classes

- Class -> Look for nouns in the problem analysis.
- Method -> corresponds to Verbs



Contoh Implementasi Class :

Abstraksi kelas



Implementasi kelas

2

```
class Dog {  
    int size;  
    String breed;  
    String name;  
  
    void bark() {  
        System.out.println("Ruff! Ruff!");  
    }  
}
```

instance variables

a method

DOG
size breed name
bark()

3 Instansiasi Objek

```
class DogTestDrive {  
    public static void main (String[] args) {  
        Dog d = new Dog();  
        d.size = 40;  
        d.bark();  
    }  
}
```

make a Dog object

use the dot operator (.) to set the size of the Dog and to call its bark() method

dot operator

3. Message

- Message merupakan suatu pemanggilan method (**method call**) dari pengirim pesan (**sender**) kepada penerima pesan (**receiver**).
- Informasi tambahan yang dikirimkan melalui “method call” disebut **arguments**.



Message Component

1. **Object identifier** : objek yang menerima pesan (receiver)
2. **Method name** : nama method yang dipanggil
3. **Arguments** : informasi tambahan yang dikirimkan melalui method, biasanya berupa input parameter.



Contd..

Benjamin as an Object

Attributes:

name = "Benjamin"

address = "1, Robinson Road"

budget = "2000"

Methods:

purchase() {

Sean.takeOrder("Benjamin", "sofa", "1, Robinson Road",
"12 November")

}

getBudget() {return budget}

The message `Sean.takeOrder(who, stock, address, date)` is interpreted as follows:

- Sean is the receiver of the message;
- takeOrder is a method call on Sean;
- "Benjamin", "stock", "address", "date" are arguments of the message.



4. Enkapsulasi (pembungkusan)

- Enkapsulasi merupakan salah satu konsep OOP yang digunakan untuk **membungkus (bundling)** sekumpulan attributes and methods kedalam satu objek dan **menyembunyikan (information hiding)** detail implementasinya dari objek lain.
- Enkapsulasi terdiri dari dua konsep, yaitu :
 1. Bundling
 2. Information Hiding



Keuntungan Enkapsulasi

- Memudahkan proses maintenance (**maintainability**)
- Perubahan kode bersifat independent
- Meningkatkan **usability**
- Meningkatkan **extensibility**



To achieve encapsulation in Java

- Declare variable of class as **private**.
- Provide **public setter** and **getter** methods to modify and view variables values.



Contoh Enkapsulasi :

```
/* File name : EncapTest.java */
public class EncapTest {
    private String name;
    private String idNum;
    private int age;

    public int getAge() {
        return age;
    }

    public String getName() {
        return name;
    }

    public String getIdNum() {
        return idNum;
    }
}
```

```
    public void setAge( int newAge) {
        age = newAge;
    }

    public void setName(String newName) {
        name = newName;
    }

    public void setIdNum( String newId) {
        idNum = newId;
    }
}
```

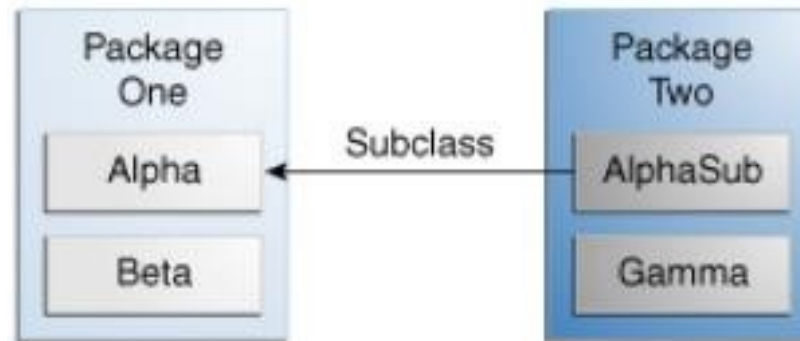


Visibility (Access Modifier)

Access Levels

Modifier	Class	Package	Subclass	World
<code>public</code>	Y	Y	Y	Y
<code>protected</code>	Y	Y	Y	N
<i>no modifier</i>	Y	Y	N	N
<code>private</code>	Y	N	N	N

Contoh implementasi Access Modifier



Visibility

Modifier	Alpha	Beta	Alphasub	Gamma
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Kesimpulan :

- Object merupakan sesuatu yang nyata / tidak nyata yang memiliki **state** (attributes) dan **behaviour** (methods).
- Komunikasi antar objek dilakukan dengan pertukaran pesan (**sending message**).
- Class merupakan **blueprint** dari sebuah objek
- Message merupakan suatu pemanggilan method (**method call**) dari pengirim pesan (**sender**) kepada penerima pesan (**receiver**).
- Enkapsulasi merupakan suatu konsep OOP yang memiliki dua konsep, yaitu (1) **bundling** dan (2) **information hiding**.



Questions



References :

1. Object oriented programming and Java
2nd Edition - Chapter 1& 2
2. Head First Java 2nd Edition



Tugas 2

Object, Class & Encapsulation



Tugas 2 :

- Buatlah sebuah implementasi Class dan Object pada dengan Bahasa java.
 - Kasus : Peminjaman barang di Teknisi JTK atau Pembelian makanan di Kantin Pujasera Polban
- Gunakan enkapsulasi untuk membungkus **attributes**-nya !
- **Deskripsikan** aplikasi yang ingin dibuatkan!
- Copykan **source code & screenshot hasil implementasi**-nya pada word



Contd..

- Tuliskan lesson learned dari pekerjaan ini
- Convert ke pdf dan Beri nama file “Tugas02_kelas_nim_nama.pdf”.
- Kumpulkan pada e-learning bagian Topic 3 - Tugas 2 : Class & Object.



Next week:

Quiz

Object and class part 2:

Relationship Between Class
Static Field & Method
Method Parameters
Object Construction
Packages

