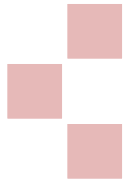
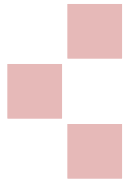


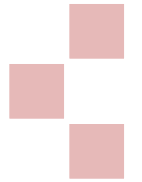
KONSEP DASAR PEMROGRAMAN BERORIENTASI OBJEK

Nurochman



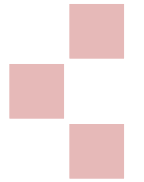
Kompetensi

- Menjelaskan paradigma pemrograman berorientasi objek
- Membedakan antara paradigma pemrograman berorientasi objek dan pemrograman prosedural
- memahami konsep class dan object
- membuat program sederhana dengan Java



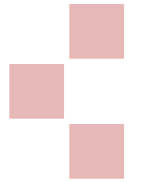
Materi

1. Prosedural **VS** OOP
2. Object, Atribut dan Behaviour
3. Class, Method, Constructor
4. Inheritance, Polymorphism
5. Instalasi JDK (Java Development Kit)
6. Kompilasi dan Run program Java



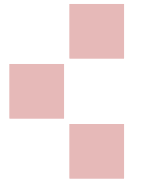
Pemrograman Prosedural

- Serangkaian tugas diselesaikan dalam bentuk fungsi atau prosedur.
- Program adalah suatu urutan instruksi.
- Programmer harus me-break down suatu problem menjadi sub problem yang lebih simple.
- Fungsi dan prosedur menjadi fokus utama.
- Fungsi dan prosedur digunakan untuk memanipulasi data.
- Data bersifat pasif.
- Penggunaan ulang fungsi dan prosedur yang sudah dibuat, harus mengkopi kode programnya



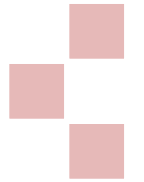
Apa itu OOP?

- Paradigma pemrograman yang memandang sistem yang akan dibangun berdasarkan objek-objek yang terlibat dan interaksinya dalam sistem
- Sistem dibangun dengan membuat objek-objek dan menginteraksikannya untuk menyelesaikan suatu pekerjaan
- Fungsi dan data menjadi satu kesatuan yang disebut sebagai **Object**
- program adalah serangkaian object yang berinteraksi untuk menyelesaikan suatu problem.



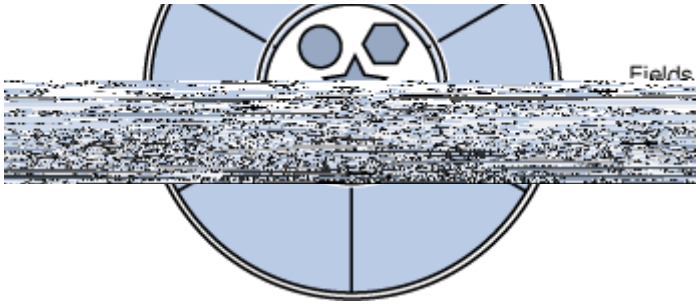
Apa itu Object ?

- An object is a software bundle of related state and behavior
- Semua benda yang ada di dunia nyata dapat dianggap sebagai obyek.
- Contoh : rumah, mobil, sepeda, motor, gelas, komputer, meja dll.

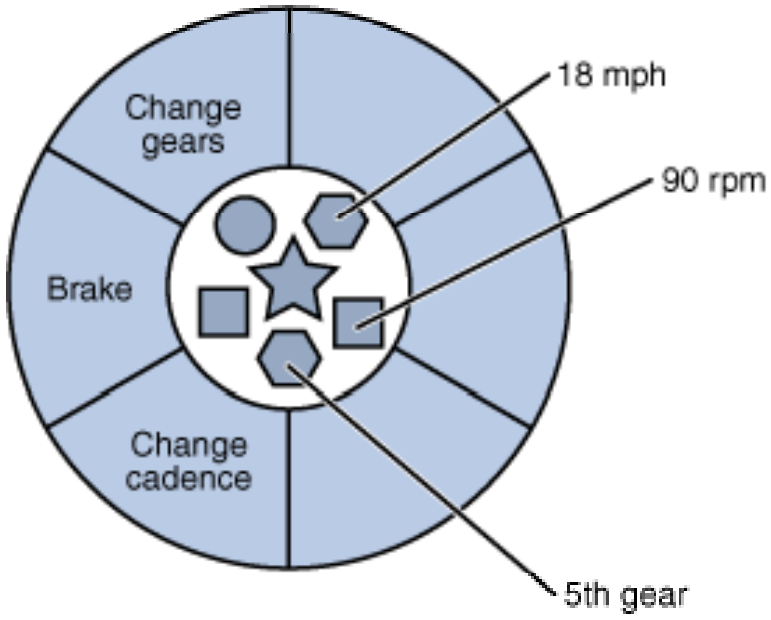


Object

- combination of code and data
- calling a function = sending a message
- implementation of function = method

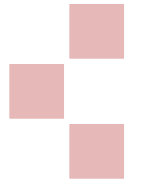


A Software Object



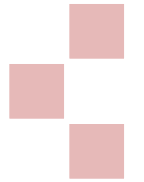
A Bicycle modeled as a software object

```
class Bicycle {
        int cadence = 0;
        int speed = 0;
        int gear = 1;

        void changeCadence(int newValue) {
             cadence = newValue;
        }

        void changeGear(int newValue) {
             gear = newValue;
        }

        void speedUp(int increment) {
             speed = speed + increment;
        }

        void applyBrakes(int decrement) {
             speed = speed - decrement;
        }

        void printStates() {
             System.out.println("cadence:"+cadence+" speed:"+speed+" gear:"+gear);
```



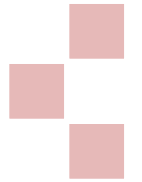
What is Class?

- A class is a blueprint or prototype from which objects are created



Class

- pattern for an object
- object factory



Class

- Definisi class :

  - merupakan template untuk membuat obyek.

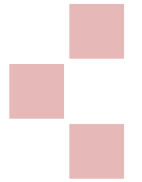
  - merupakan prototipe / blue print yang mendefinisikan variable dan method secara umum.

- Obyek (***instances of class)*** merupakan hasil instansiasi dari suatu class, proses pembentukan obyek dari suatu class disebut dengan ***instantiation***.

- Objek-objek yang dibuat dari class yang sama mempunyai atribut dan method yang sama, namun nilainya bisa saja berbeda

Atribut

- Definisi atribut : adalah **data** yang dimiliki objek dan membedakan antara obyek satu dengan yang lain.
- Dalam class atribut disebut **Variabel**.
- Contoh objek manusia mempunyai atribut: nama, tinggi badan, berat badan, warna kulit, jenis rambut, dll



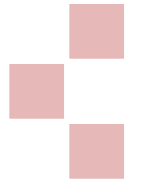
Variabel

- **Instance variable**: adalah atribut untuk tiap obyek dari class yang sama.
- Tiap obyek mempunyai dan menyimpan nilai atributnya sendiri.
- Jadi tiap obyek dari class yang sama boleh mempunyai nilai yang sama atau beda
- **Class variable**: adalah atribut untuk semua obyek yang dibuat dari class yang sama.
- Semua obyek mempunyai nilai atribut yang sama.
- Jadi semua obyek dari class yang sama mempunyai hanya satu nilai yang sama.

Behavior

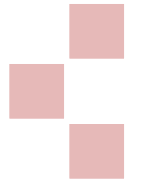
- Behavior / perilaku adalah hal-hal yang bisa dilakukan oleh obyek dari suatu class.
- Behavior dapat digunakan untuk mengubah nilai atribut suatu obyek, menerima informasi dari obyek lain, dan mengirim informasi ke obyek lain untuk melakukan suatu tugas.
- Dalam class, behavior disebut juga sebagai **methods**.
- Contoh manusia memiliki perilaku:

  jalan, lari, duduk, menyapa, makan, dll.



Method

- Method: adalah suatu perintah dalam class yang berisi serangkaian instruksi untuk melakukan tugas tertentu.
- Method dapat memiliki parameter ataupun tidak.



Konsep Dasar OOP

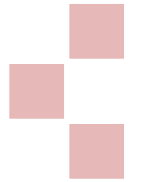
- Enkapsulasi (Encapsulation)
- Pewarisan (Inheritance)
- Polimorfisme (Polymorphism)

Enkapsulasi

- **Definisi enkapsulasi**: Pembungkusan variabel dan method dalam sebuah obyek yang terlindungi serta menyediakan interface untuk mengakses variabel tersebut.
- Variabel dan yang method yang dipunyai suatu obyek, bisa ditentukan hak aksesnya.
- Contoh : ketika kita ingin mempercepat laju motor yg kita naiki, mk cukup kita tarik handlenya tanpa perlu tahu proses apa yg terjadi sehingga laju semakin cepat.

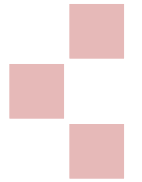
Pewarisan

- Pewarisan merupakan penurunan kemampuan (atribut dan method) dari sebuah class ke class lainnya. Pengembangan fitur sebuah class.
- Class yang mewarisi → superclass
- Class yang diwarisi → subclass
- Subclass bisa berlaku sebagai superclass bagi class lainnya → multilevel inheritance.
- Contoh : sepeda ontel merupakan superclass dari sepeda motor



Keuntungan Pewarisan

- Subclass menyediakan state/behaviour yang spesifik yang membedakan dengan superclass → memungkinkan programmer untuk menggunakan ulang source code dari superclass yang telah ada.
- Programmer dapat mendefinisikan superclass khusus yang bersifat generik, yang disebut abstract class, untuk mendefinisikan class dengan tingkah laku dan state secara secara umum.

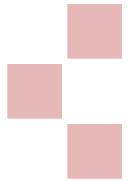


Single VS Multiple

- C→ multiple inheritance

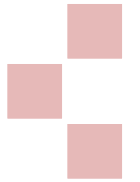
  Suatu class diperbolehkan untuk mempunyai lebih dari satu superclass. Variabel dan method yang diwariskan merupakan kombinasi dari superclass-nya.
- Java → single inheritance

  Suatu class hanya boleh mempunyai satu superclass
- Dalam multilevel inheritance, suatu subclass bisa menjadi superclass bagi yang lain.



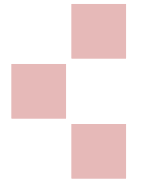
Java Technology

- Official Site : http://java.sun.com
- **J2SE** (Java 2 standard edition)
  - Desktop Application
  - Applet (Aplikasi Java di Browser)
- **J2ME** (Java 2 Micro Edition)
  - MIDlet Application/ Micro Devices App
- **J2EE** (Java 2 Enterprise Edition)
  - Web Application (JSP)
  - Servlet Application



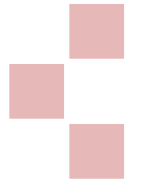
Editor Java

- Editor teks : notepad, vi, emacs
- IDE java free : Netbeans IDE, Eclipse
- IDE berlisensi : Borland Jbuilder, Visual Age



Program Hello

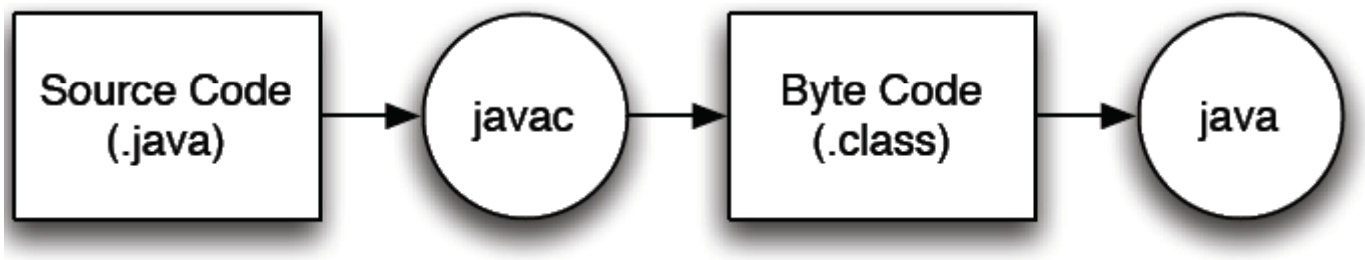
```
class ClassName {
	public static void main(String args[]) {
		//isi program
	}
}

class Hello {
	public static void main(String args[]) {
		System.out.println(“Hello Java World”);
	}
}
```



Kompilasi dan Run

- Kompilasi dengan javac

  javac Hello.java
- Running dengan java

  java Hello

Proses kompilasi dan Running