

Konsep Dasar Pemrograman Berorientasi Obyek



Berorientasi Object?



Attribute:

Topi, Baju, Tas
Punggung, Jaket,
Tangan, Kaki, Mata

Behavior:

Cara Jalan ke Depan
Cara Jalan Mundur
Cara Belok ke Kiri
Cara Memanjat

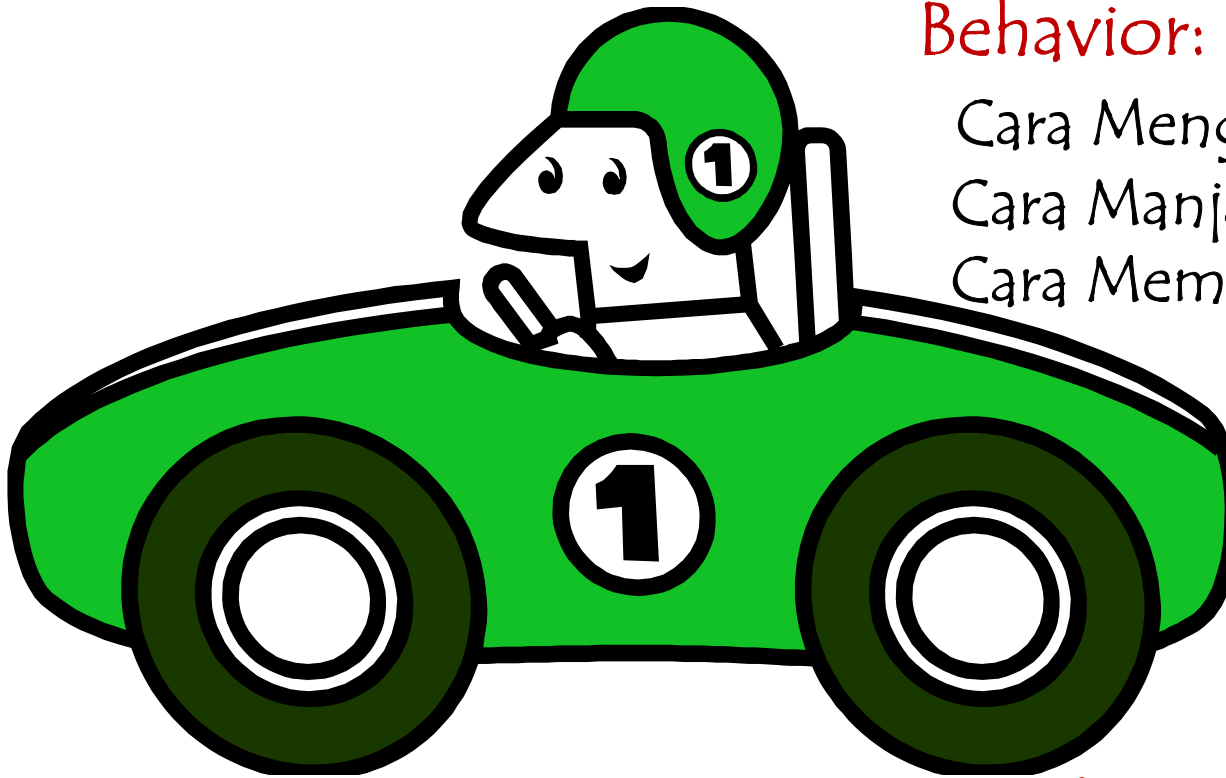
Berorientasi Object?

Attribute (State):

Ban, Stir, Pedal Kopling/Gas/Rem,
Warna, Tahun Produksi

Behavior:

Cara Menghidupkan Mesin
Cara Manjalankan Mobil
Cara Memundurkan Mobil



Attribute → Variable(Member)

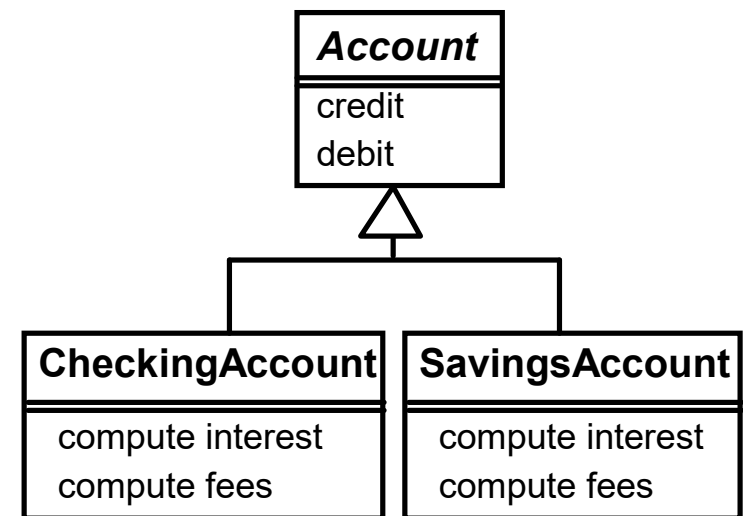
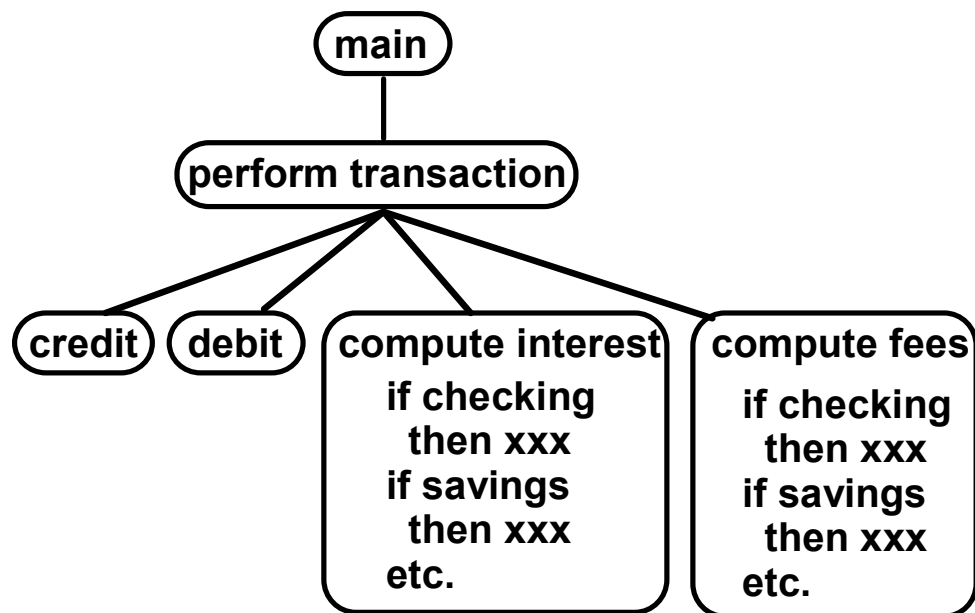
Behavior → Method(Fungsi)



Tugas

1. Lakukan kegiatan berorientasi objek dengan target **Sepeda**
2. Pikirkan apa **atribut dan behavior** yang dimiliki oleh Sepeda tersebut

Procedural vs Object-Oriented



Object-Oriented Program

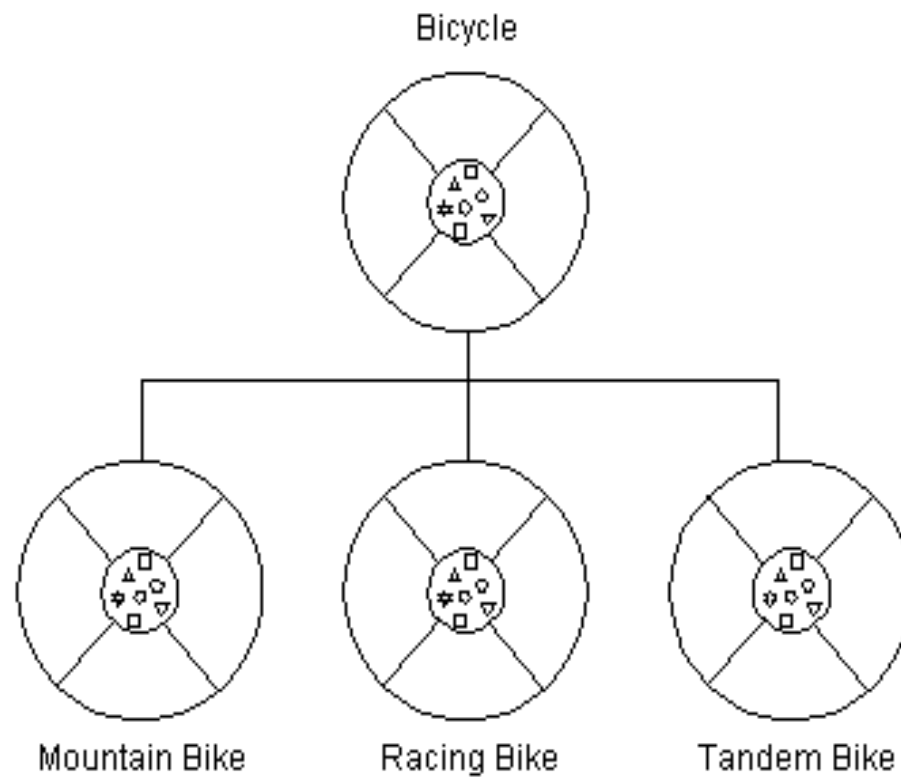
```
public class Halo{  
  
    public static void main(String[] args){  
        System.out.println("Halo Semarang");  
    }  
}
```

Karakteristik OOP -1-

- **Abstraction**: Cara kita melihat suatu sistem dalam bentuk yang lebih sederhana, yaitu sebagai suatu kumpulan subsistem (object) yang saling berinteraksi.
 - Mobil adalah kumpulan sistem pengapian, sistem kemudi, sistem pengereman
- **Encapsulation**: Mekanisme menyembunyikan suatu proses dalam sistem untuk menghindari interferensi dan menyederhanakan penggunaan sistem itu sendiri
 - Tingkat transmisi (gigi) pada mobil
 - Tombol on/off/pengaturan suhu pada AC

Karakteristik OOP -2-

- **Inheritance**: Suatu class dapat **mewariskan atribut dan method** kepada class lain (subclass), serta membentuk class hierarchy



Karakteristik OOP -3-

- **Modularity**: Sifat object adalah modular. Object dapat ditulis dan dimaintain terpisah (independen) dari object lain
- **Polymorphism**: Suatu object dapat memiliki berbagai bentuk, sebagai object dari classnya sendiri atau object dari superclassnya
 - **Overloading**: Penggunaan satu nama untuk beberapa method yang berbeda (beda parameter)
 - **Overriding**: Terjadi ketika deklarasi method subclass dengan nama dan parameter yang sama dengan method dari superclassnya

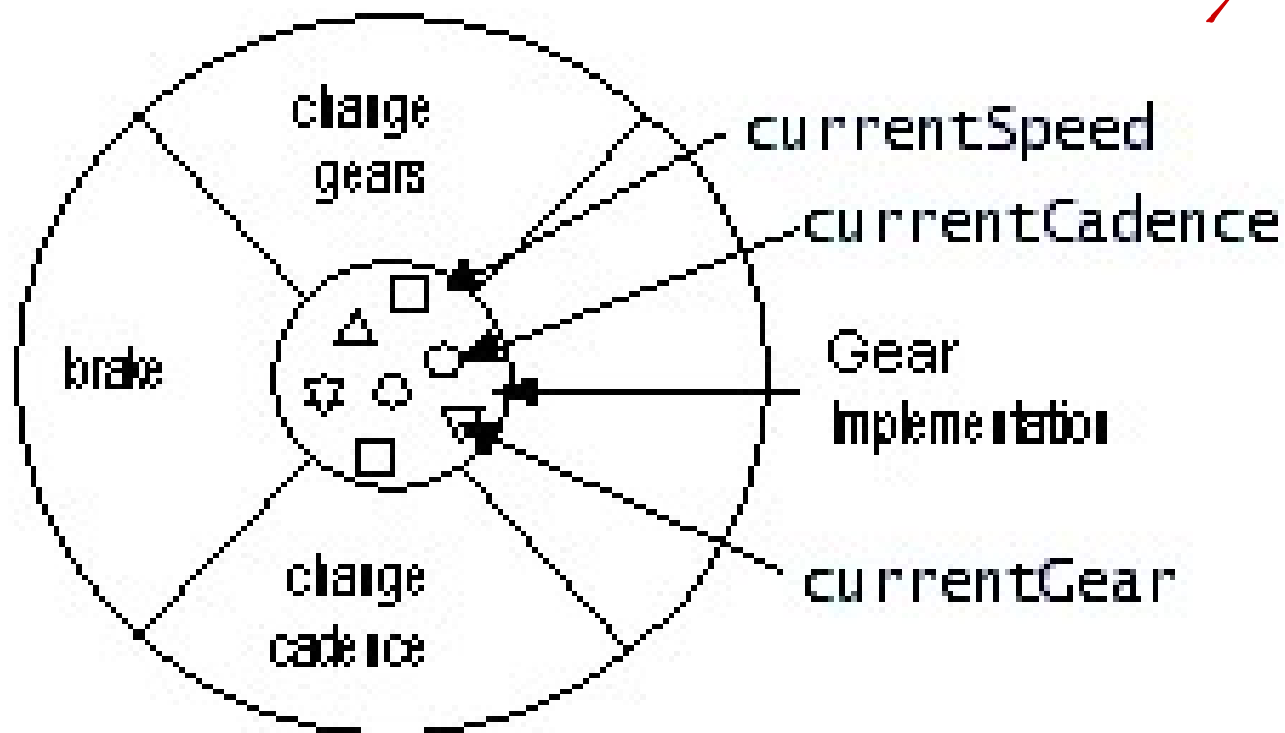
Class dan Object -1-

- Class: mobil
- Object: mobilnya pak Joko, mobilku
- Class seperti cetakan kue, dimana kue yg dihasilkan dari cetakan kue itu adalah object
- Warna kue bisa bermacam-macam meskipun berasal dari cetakan yang sama (object memiliki sifat independen)
- Object adalah instance dari Class

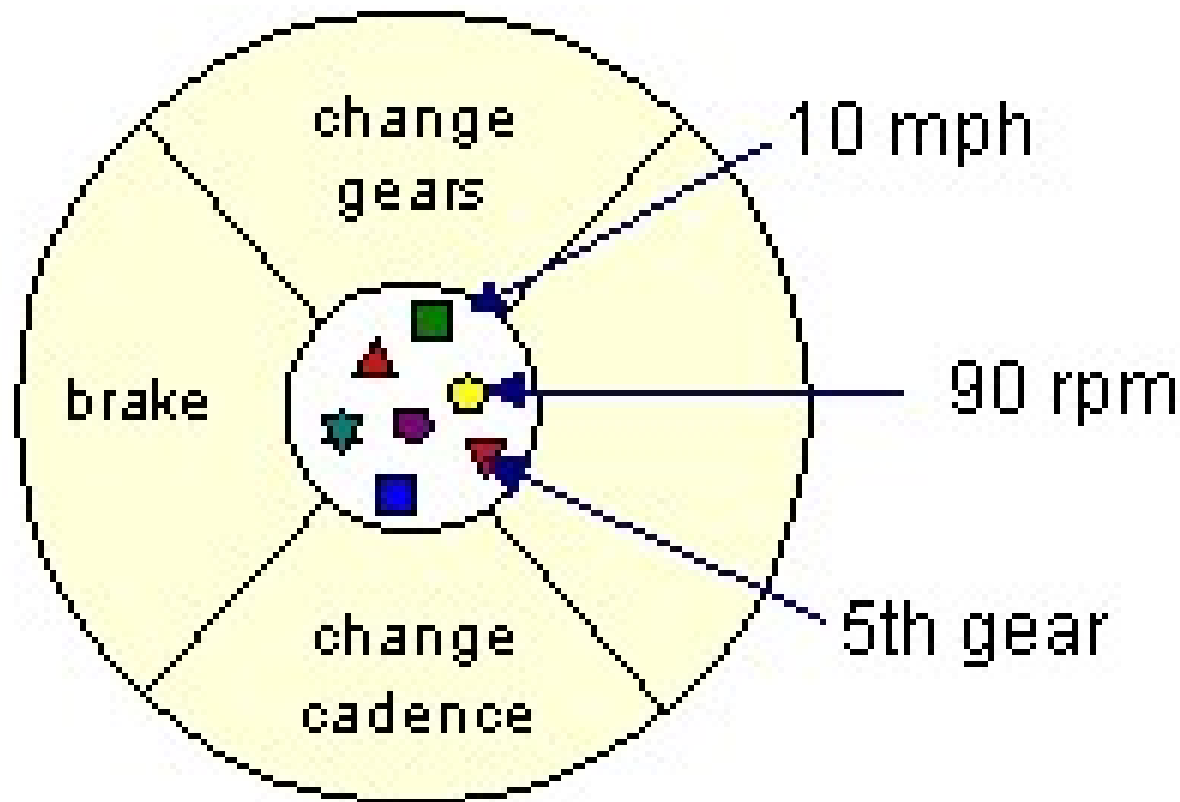
Class and Object -2-

Class = method+attribute

The Bicycle class



Class dan Object -3-



Bicycle dimodelkan sebagai sebuah object:

- 10 mph, 90 rpm, 5th gear adalah **instance variable**
- changegears, brake, changecadence adalah **instance method**



Attribute

Attribute adalah berbagai variable yang mengitari class, dengan nilai datanya bisa ditentukan di object. Name, age, dan weight adalah attribute dari class Person

Person
name : string
age : integer

Class with Attributes

(Person)
Joe Smith
24

(Person)
Mary Sharp
52

Objects with Values

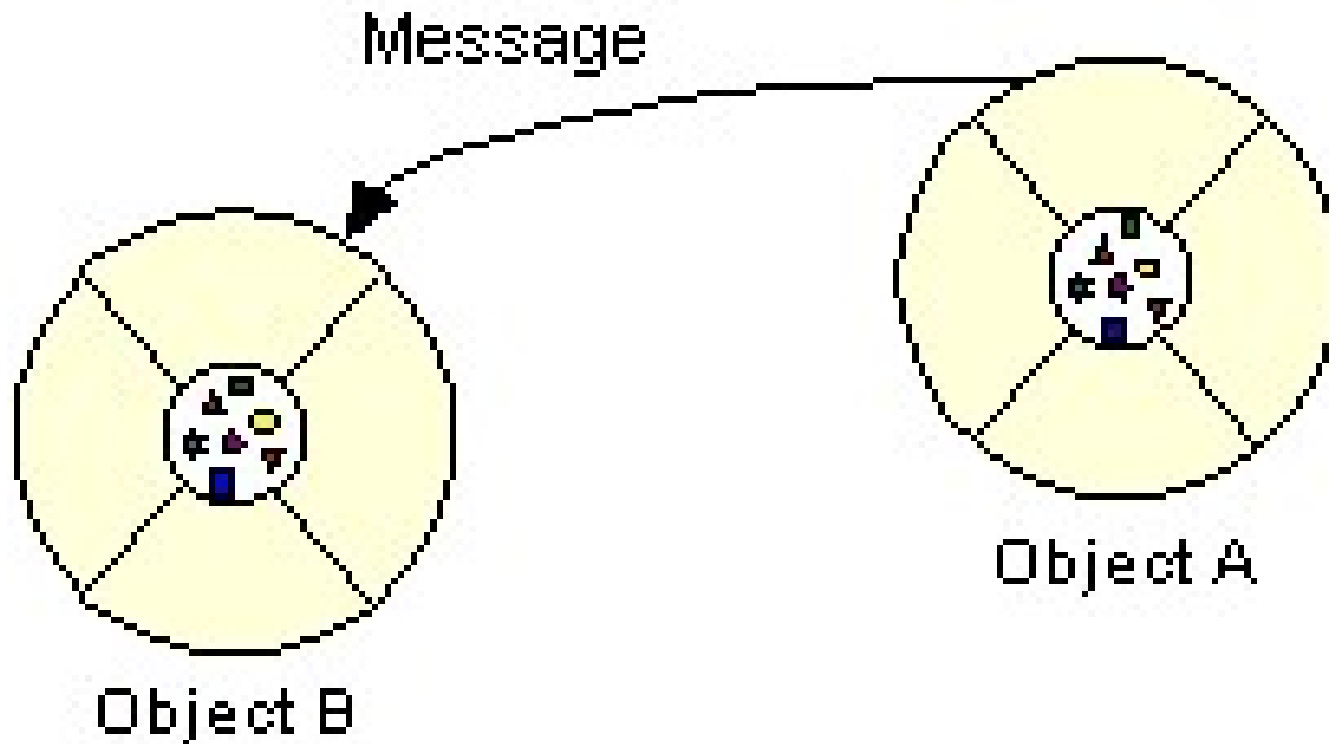


Method

- Method adalah implementasi dari bagaimana bekerjanya sebuah class
- Method melakukan:
 1. Manipulasi data
 2. Perhitungan matematika
 3. Memonitor kejadian dari suatu event

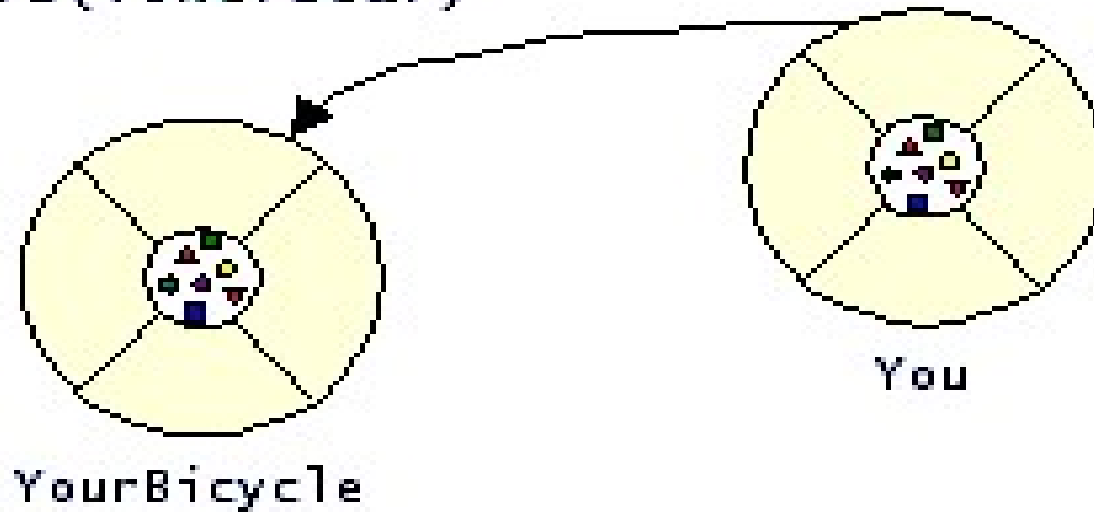
Message -1-

- Bicycle akan berguna apabila ada object lain (misalnya anda) yang berinteraksi dengan Bicycle tersebut
- Object software berinteraksi dan berkomunikasi dengan object lain dengan cara mengirimkan *message*. Informasi dalam message ini dikenal dengan nama *Parameter*



Message -2-

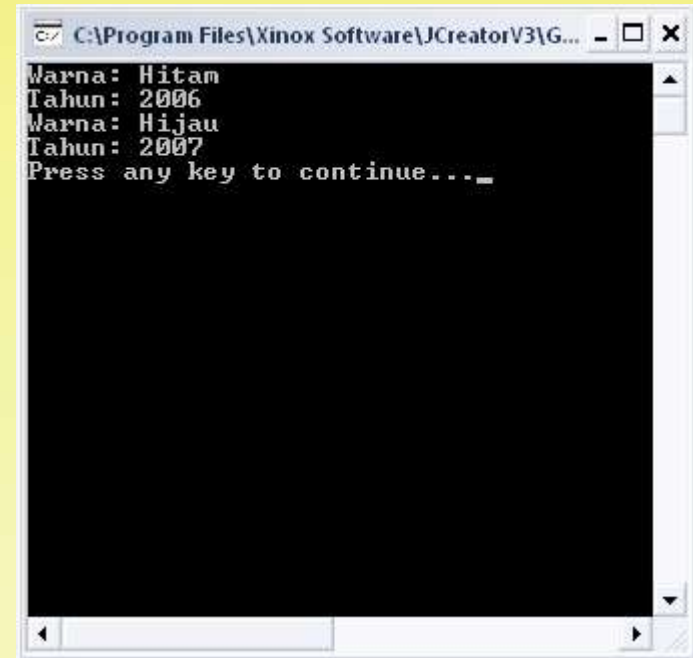
`changeGears(lowerGear)`



1. You → object pengirim (sender)
2. YourBicycle → object penerima (receiver)
3. changeGears → the message, the method to perform
4. lowerGear → information from You to YourBicycle, the parameters needed by the method

Membuat Class dan Memanggil Object (MobilDemo.java)

```
class Mobil {  
    String warna;  
    int tahunProduksi;  
}  
  
public class MobilDemo {  
    public static void main(String[] args){  
        // Membuat object  
        Mobil mobilku = new Mobil();  
  
        //memanggil atribut dan memberi nilai  
        mobilku.warna = "Hitam";  
        mobilku.tahunProduksi = 2006;  
        System.out.println("Warna: " + mobilku.warna);  
        System.out.println("Tahun: " + mobilku.tahunProduksi);  
    }  
}
```



```
Warna: Hitam  
Tahun: 2006  
Warna: Hijau  
Tahun: 2007  
Press any key to continue..._
```



Tugas

- Kembangkan Class **Mobil**, ubah nama menjadi **Mobil2** masukkan method:
 - **hidupkanMobil**
 - **matikanMobil**
 - **ubahGigi**
- Panggil method-method diatas dari **MobilDemo2**

Membuat Class (Bicycle.java)

```
class Bicycle {  
    int speed = 0;  
    int gear = 0;  
  
    // method  
    void changeGear(int newValue) {  
        gear = gear + newValue;  
        System.out.println("\nGear:" + gear);  
    }  
    void speedUp(int increment) {  
        speed = speed + increment;  
        System.out.println("\nSpeed:" + speed);  
    }  
}
```



Membuat dan Memanggil Object

```
class BicycleDemo {  
    public static void main(String[] args) {  
        // Membuat object  
        Bicycle bike = new Bicycle();  
  
        //memanggil atribut dan memberi nilai  
        bike.speed=10;  
        bike.gear=2;  
  
        // Memanggil method dan menunjuk nilai parameter  
        bike.speedUp(10);  
        bike.changeGear(2);  
    }  
}
```



Tugas

- Buat Class bernama **matematika**, yang berisi method:
 - pertambahan
 - pengurangan
 - perkalian
 - pembagian
- Buat Class bernama **matematikaDemo**, yang mengeksekusi method dan menampilkan:
 - Pertambahan: $20 + 20 = 40$
 - Pengurangan: $10 - 5 = 5$
 - Perkalian: $10 \times 20 = 200$
 - Pembagian: $20 / 2 = 10$

Tips Membuat Class

- 1 file bisa berisi ≥ 1 class
- 1 file hanya dapat berisi 1 public class
- Nama file harus sama dengan nama public class sifatnya adalah case sensitive
- Tips: Lebih baik dan lebih mudah dipahami apabila satu file berisi hanya satu class

Kata Kunci this

Digunakan pada pembuatan class dan digunakan untuk **menyatakan object sekarang**

```
class Mobil{  
    String warna;  
    Int tahunProduksi;  
  
    void isiData(String warnaMobil, int  
        tahunProduksiMobil){  
  
        warna = warnaMobil;  
        tahunProduksi = tahunProduksiMobil;  
    }  
}
```

```
class Mobil{  
    String warna;  
    Int tahunProduksi;  
  
    void isiData(String warna, int  
        tahunProduksi){  
  
        this.warna = warna;  
        this.tahunProduksi = tahunProduksi;  
    }  
}
```

Konstruktor -1-

- Method yang digunakan untuk memberi nilai awal pada saat object diciptakan
- Dipanggil secara otomatis ketika **new** digunakan untuk membuat instan class
- Sifat konstruktor:
 - Nama konstruktor sama dengan nama class
 - Tidak memiliki nilai balik dan tidak boleh ada kata kunci void

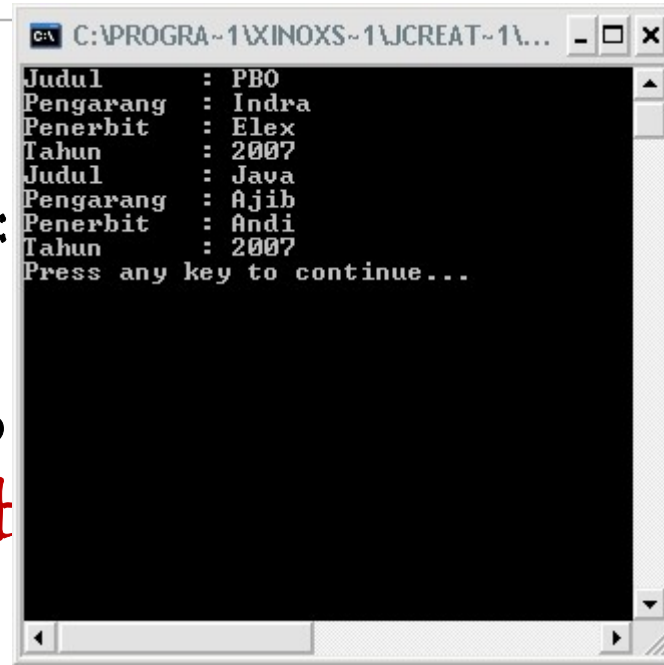
Konstruktor -2-

```
class Mobil {  
    private String warna;  
    private int tahunProduksi;  
  
    public Mobil(String warna, int tahunProduksi){  
        this.warna = warna;  
        this.tahunProduksi = tahunProduksi;  
    }  
  
    public void info(){  
        System.out.println("Warna: " + this.warna);  
        System.out.println("Tahun: " + this.tahunProduksi);  
    }  
}  
  
public class Konstruktor{  
    public static void main(String[] args){  
        Mobil mobilku = new Mobil("Merah", 2003);  
        mobilku.info();  
    }  
}
```



Tugas:

1. Buat class **Buku**
2. Tentukan variable buku:
Penerbit dan Tahun
3. Buat konstruktor class B
Judul, Pengarang, Penerbit
4. Buat method **cetakBuku**
5. Buat class **BukuDemo** dan tampilkan 2 buku:
 - Pemrograman Berbasis Objek dengan Java, Indrajani, Elexmedia Komputindo, 2007
 - Dasar Pemrograman Java, Abdul Kadir, Andi Offset, 2004



```
Judul      : PBO
Pengarang  : Indra
Penerbit   : Elex
Tahun      : 2007
Judul      : Java
Pengarang  : Ajib
Penerbit   : Andi
Tahun      : 2007
Press any key to continue...
```

Rehat Sejenak

- [Kepleset](#)
- [Password](#)

Referensi

- Object First With Java, Fifth edition, David J. Barnes & Michael Kölling, Prentice Hall / Pearson Education, 2012.
- The Java™ Tutorial,
<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/>,
Oracle, 1995–2014.
- Java SE Tutorial,
<http://www.oracle.com/technetwork/java/javase/downloads/java-se-7-tutorial-2012-02-28-1536013.html>, Oracle, 2014.
- SCJP Sun Certified Programmer for Java™ 6 Study Guide Exam (310-065), Kathy Sierra & Bert Bates, Mc Graw Hill, 2008.
- Object Oriented Programming with Java, Romi Satria Wahono, 2008.