PEMROGRAMAN BERORIENTASI OBJECT

Konsep Dasar OOP

Rules

Don't be late

Don't call (WA Only)

Monday-Friday(07.00 - 18.00 WIB)

Don't cheat

Discount 50%

Outline

- PBO vs Struktural
- Konsep dasar PBO
- UML Class Diagram

Matakuliah PBO

- Pemrograman Berbasis Objek (2 SKS/3 Jam)
- Praktikum Pemrograman Berbasis Objek (3 SKS/6 Jam)
- Capaian Pembelajaran
  - Mampu membuat program dengan menggunakan prinsip-prinsip OOP menggunakan bahasa pemrograman Java
- Software :
  - JDK
  - Netbeans

Silabus

Pertemuan Ke-	Pembahasan
1	Pengantar Konsep Dasar OOP
2	Class dan Object
3	Enkapsulasi
4	Relasi Class
5	Kuis 1
6	Inheritance
7	Overriding dan Overloading
8	Ujian Tengah Semester (UTS)
9	Abstract Class dan Interface
10	Polimorfisme
11	GUI
12	Java API
13	Kuis 2
14	GUI dan Database
15	Unit Testing
16	Tugas Besar
17	Ujian Akhir Semester (UAS)

Komponen Penilaian

- **Teori:**

KUIS	:	2	Bobot Kuis	25 %
TUGAS	:	6	Bobot Tugas	20 %
UTS	:	1	Bobot UTS	25 %
UAS	:	1	Bobot UAS	30 %

- **Praktek:**

KUIS	:	2	Bobot Kuis	20 %
TUGAS	:	12	Bobot Tugas	30 %
UTS	:	1	Bobot UTS	20 %
UAS	:	1	Bobot UAS	30 %

Referensi

- Horstmann, C. S., & Cornell, G. (2007). *Core Java Volume I–Fundamentals, Eighth Edition*. Network Circle, Santa Clara: Prentice Hall.
- Horstmann, C. S., & Cornell, G. (2008). *Core Java Volume II–Advanced Features, Eighth Edition*. Network Circle, Santa Clara: Prentice Hall.
- https://www.javatpoint.com/java-oops-concepts
- Dapat di download di http://libgen.io/

Objek Oriented vs Struktural

- Struktural
  - Program dipecah kedalam fungsi
  - Perubahan fitur → kemungkinan mengganggu keseluruhan program
- Object Oriented
  - Program dipecah kedalam object
    - Didalamnya terdapat state dan behavior
  - Perubahan fitur → tidak mengganggu keseluruhan program

Objek Oriented vs Struktural

- Contoh:
- Kita akan membuat program game simulasi sepeda, didalamnya ada karakter sepeda yang memiliki kecepatan, gear dan merk.
- Bagaimana membangun game tersebut dengan metode **konvensional**?
  - Langkah pertama kita buat variabelnya, misal kecepatan, gear, merk
  - Langkah berikutnya kita buat fungsi-fungsinya, tambah kecepatan, kurangi kecepatan.
  - Langkah berikutnya kita coba mengoperasikan sepeda tersebut secara sederhana, yaitu memanipulasi kecepatan, gear, merk nya, didalam fungsi main, kemudian kita cetak ke layar.
- Kode program →…

```java
public class SepedaStruktural
{
    public static void main(String[] args)
    {
        String merek;
        int kecepatan, gear;

        merek = "Poligone";
        kecepatan = 10;
        gear = 1;

        kecepatan = tambahKecepatan(kecepatan, 10);

        System.out.println("Merek: " + merek);
        System.out.println("Kecepatan: " + kecepatan);
    }

    public static int tambahKecepatan(int kecepatan, int increment)
    {
        kecepatan += increment;

        return kecepatan;
    }

    public static int kurangiKecepatan(int kecepatan, int decrement)
    {
        kecepatan -= decrement;

        return kecepatan;
    }
}
```

```java
public class Sepeda {
    public String merk;
    public int kecepatan,gear;

    public Sepeda(String m, int k,int g){
        merk = m;
        kecepatan = k;
        gear = g;
    }

    public int tambahKecepatan(int increment){
        kecepatan += increment;
        return kecepatan;
    }

    public int kurangKecepatan(int decrement){
        kecepatan -= decrement;
        return kecepatan;
    }

    public void info(){
        System.out.println("Merk: " + merk);
        System.out.println("Kecepatan: " + kecepatan);
    }

    public static void main(String[] args) {
        Sepeda spd1 = new Sepeda("Poligon",10,1);

        spd1.tambahKecepatan(10);
        spd1.info();
    }
}
```

Objek Oriented vs Struktural

- Bagaimana jika ada dua sepeda di game?
  - Tambahkan variabel merek2, kecepatan2, gear2
  - Coba manipulasi nilai-nilai variabelnya kemudian tampilkan ke layar
- Kode program → …

```java
public class SepedaStruktural
{
    public static void main(String[] args)
    {
        String merek, merek2;
        int kecepatan, kecepatan2, gear, gear2;

        merek = "Poligone";
        kecepatan = 10;
        gear = 1;

        merek2 = "Wiim Cycle";
        kecepatan2 = 15;
        gear2 = 3;

        kecepatan = tambahKecepatan(kecepatan, 10);
        kecepatan2 = tambahKecepatan(kecepatan2, 5);

        System.out.println("Merek: " + merek);
        System.out.println("Kecepatan: " + kecepatan);

        System.out.println("Merek: " + merek2);
        System.out.println("Kecepatan: " + kecepatan2);
    }

    public static int tambahKecepatan(int kecepatan, int increment)
    {
        kecepatan += increment;

        return kecepatan;
    }

    public static int kurangiKecepatan(int kecepatan, int decrement)
    {
        kecepatan -= decrement;

        return kecepatan;
    }
}
```

```java
public class SepedaStruktural
{
    public static void main(String[] args)
    {
        String merek, merek2;
        int kecepatan, kecepatan2, gear, gear2;

        merek = "Poligone";
        kecepatan = 10;
        gear = 1;

        merek2 = "Wiim Cycle";
        kecepatan2 = 15;
        gear2 = 3;

        kecepatan = tambahKecepatan(kecepatan, 10);
        kecepatan2 = tambahKecepatan(kecepatan2, 5);

        System.out.println("Merek: " + merek);
        System.out.println("Kecepatan: " + kecepatan);

        System.out.println("Merek: " + merek2);
        System.out.println("Kecepatan: " + kecepatan2);
    }

    public static int tambahKecepatan(int kecepatan, int increment)
    {
        kecepatan += increment;

        return kecepatan;
    }

    public static int kurangiKecepatan(int kecepatan, int decrement)
    {
        kecepatan -= decrement;

        return kecepatan;
    }
}
```

12 baris

Objek Oriented vs Struktural

- Bagaimana jika ada **sepuluh** sepeda?
  - Tambahkan variabel merek3, kecepatan3, gear3 ..... merek9, kecepatan9, gear9
  - Cukup melelahkan…
- Bagaimana dengan object oriented?
  - Buat sebuah class Sepeda yang memiliki atribut merek, kecepatan, gear.
  - Buat 10 object sepeda
- Kode program → …

```java
public class Sepeda {
    public String merk;
    public int kecepatan,gear;

    public Sepeda(String m, int k,int g){
        merk = m;
        kecepatan = k;
        gear = g;
    }

    public int tambahKecepatan(int increment){
        kecepatan += increment;
        return kecepatan;
    }

    public int kurangKecepatan(int decrement){
        kecepatan -= decrement;
        return kecepatan;
    }

    public void info(){
        System.out.println("Merk: " + merk);
        System.out.println("Kecepatan: " + kecepatan);
    }

    public static void main(String[] args) {
        // sepeda pertama
        Sepeda spd1 = new Sepeda("Poligon",10,1);

        spd1.tambahKecepatan(10);
        spd1.info();

        //sepeda kedua
        Sepeda spd2 = new Sepeda("Wim Cycle", 15,3);

        spd2.tambahKecepatan(5);
        spd2.info();
    }
}
```

```java
public class Sepeda {
    public String merk;
    public int kecepatan,gear;

    public Sepeda(String m, int k,int g){
        merk = m;
        kecepatan = k;
        gear = g;
    }

    public int tambahKecepatan(int increment){
        kecepatan += increment;
        return kecepatan;
    }

    public int kurangKecepatan(int decrement){
        kecepatan -= decrement;
        return kecepatan;
    }

    public void info(){
        System.out.println("Merk: " + merk);
        System.out.println("Kecepatan: " + kecepatan);
    }

    public static void main(String[] args) {
        // sepeda pertama
        Sepeda spd1 = new Sepeda("Poligon",10,1);

        spd1.tambahKecepatan(10);
        spd1.info();

        //sepeda kedua
        Sepeda spd2 = new Sepeda("Wim Cycle", 15,3);

        spd2.tambahKecepatan(5);
        spd2.info();
    }
}
```

6 baris

Objek Oriented vs Struktural

- Dapat kita lihat bahwa dengan OOP, untuk membuat banyak sepeda, kita tidak perlu tuliskan berulang-ulang variabel merek, kecepatan, dan gear.
- Kita cukup buat banyak objek sepeda saja, dari sebuah class sepeda yang sudah kita buat.

Konsep OOP

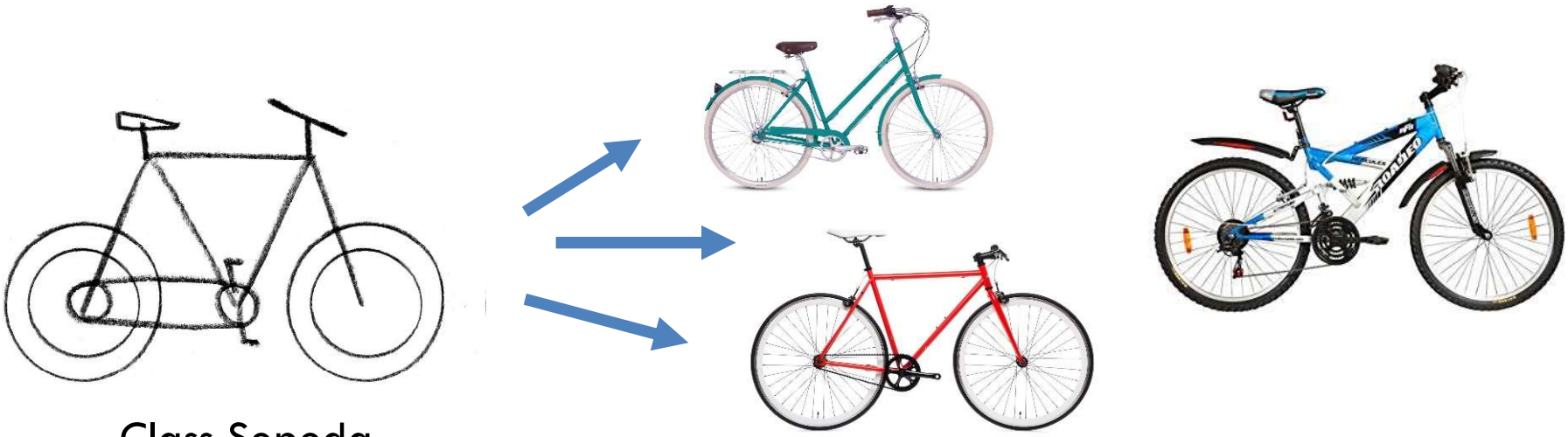
- Beberapa aspek dalam OOP:
  - Object
  - Class
  - Enkapsulasi
  - Inheritance
  - Polimorfisme

Object

- Object adalah suatu rangkaian dalam program yang terdiri dari **state** dan **behaviour**.
- Object pada software dimodelkan sedemikian rupa sehingga mirip dengan objek yang ada di dunia nyata.
- Objek memiliki state dan behaviour.
- State adalah ciri-ciri atau atribut dari objek tersebut.
  - Misal objek Sepeda, memiliki state **merek, kecepatan, gear** dan sebagainya.
- Behaviour adalah perilaku yang dapat dilakukan objek tersebut.
  - Misal pada Sepeda, behaviournya antara lain, **tambah kecepatan, pindah gear, kurangi kecepatan, belok**, dan sebagainya.

Class

- Class adalah blueprint atau prototype dari objek.
- Ambil contoh objek sepeda.
  - Terdapat berbagai macam sepeda di dunia, dari berbagai merk dan model.
  - Namun semua sepeda dibangun berdasarkan blueprint yang sama, sehingga tiap sepeda memiliki komponen dan karakteristik yang sama.
- Sepeda yang anda miliki dirumah, adalah hasil **instansiasi** dari **class** sepeda.



Class Sepeda

Object Sepeda

Inheritance

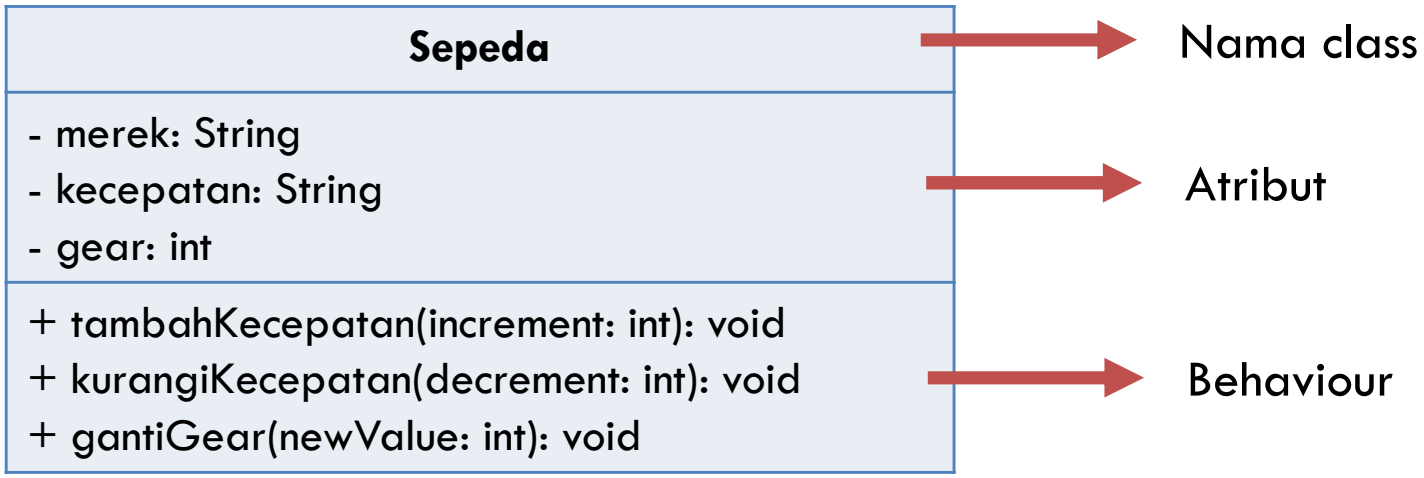
- Memungkinkan kita untuk mengorganisir struktur program dengan natural.
- Memperluas fungsionalitas program tanpa harus mengubah banyak bagian program.
- Contoh di dunia nyata:
  - Objek sepeda dapat diturunkan lagi ke model yang lebih luas, misal sepeda gunung (mountain bike) dan city bike.
  - Masing-masing dapat memiliki komponen/fitur tambahan, misal sepeda gunung memiliki **suspensi,** yang tidak dimiliki sepeda biasa. Dan city bike memiliki keranjang di bagian depannya.
  - Dalam hal ini, objek mountain bike dan road bike **mewarisi** objek sepeda.

Polimorfisme

- Polimorfisme juga meniru sifat objek di dunia nyata, dimana sebuah objek dapat memiliki bentuk
  - Atau menjelma menjadi bentuk-bentuk lain.
- Misalkan saja objek pesawat terbang.
  - Objek ini dapat diwariskan menjadi pesawat jet dan pesawat baling-baling.
  - Keduanya memiliki kemampuan untuk menambah kecepatan.
  - Namun secara teknis, metode penambahan kecepatan antara pesawat jet dengan baling-baling tentu berbeda, karena masing-masing memiliki jenis mesin yang berbeda.

UML Class Diagram

- Dalam pemrograman beroriantasi objek, rancangan class digambarkan dengan UML Class Diagram
  - UML adalah singkatan dari Unified Modelling Language
- Misal class Sepeda, yang memiliki state **merek, kecepatan, gear** dan behavior **tambahKecepatan, kurangiKecepatan, gantiGear** digambarkan dengan class diagram sebagai berikut:



Latihan

- Carilah objek apa saja di dunia nyata sebanyak 5.
- Tuliskan state dan behavior objek tersebut. Makin banyak state dan behavior makin baik. Contoh:
- Televisi
  - State:
    - Merek
    - Ukuran layar
    - Channel
    - Volume
  - Behavior:
    - Nyalakan
    - Matikan
    - Pindah channel
    - Tambah volume
    - Kurangi volume