

# 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

# Matakuliah PBO

- Pemrograman Berbasis Objek (2 SKS/4x50 menit)
- Praktikum Pemrograman Berbasis Objek (2 SKS/4x50 menit)
- Capaian Pembelajaran
  - ▣ Mampu membuat program dengan menggunakan prinsip-prinsip OOP menggunakan bahasa pemrograman Java

# 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
10	Interface
11	Polimorfisme
12	Kuis 2
13-16	OOP pada Bahasa pemrograman lain
17	Ujian Akhir Semester (UAS)



**Pemrograman  
Berorientasi Objek**

**Vs**


**Pemrograman  
Struktural**





## — PEMROGRAMAN STRUKTURAL

Paradigma pemrograman dengan konsep bahwa program merupakan urutan langkah-langkah yang dilakukan untuk menyelesaikan suatu masalah



# Pemrograman Struktural

- Contoh:
- Kita akan membuat program game simulasi sepeda, didalamnya ada objek sepeda yang memiliki kecepatan, gear dan merk.
- Bagaimana membangun game tersebut dengan metode **struktural**?
  - ▣ 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 →...



```
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;
}
```

# Pemrograman Struktural

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

```
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  
..... merek10, kecepatan10, gear10

***Algorithms + Data Structures = Programs***  
**(Niklaus Wirth, 1975, Prentice Hall)**



## — PEMROGRAMAN BERORIENTASI OBJEK

Paradigma pemrograman yang berfokus pada objek

Data + Algorithm



# Konsep OOP

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

The background features two decorative dotted lines, one on the left and one on the right, both curving upwards. There are two light blue plus signs: one in the upper right quadrant and one in the lower left quadrant.

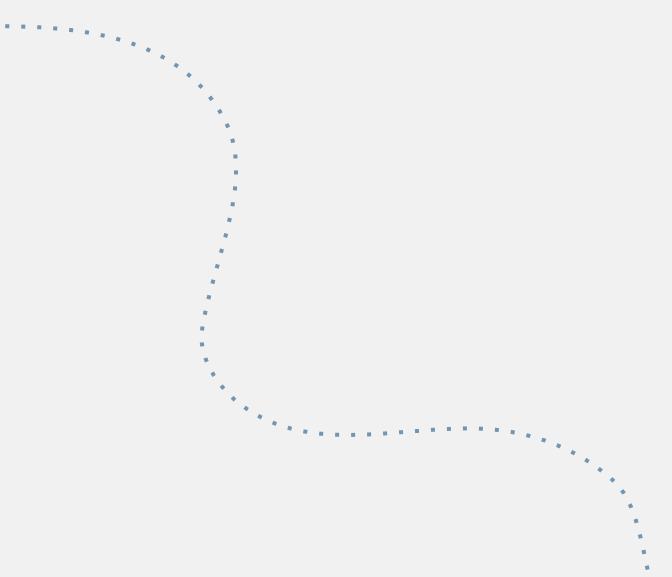
# CLASS & OBJECT





## — OBJEK

Siapa dan apa saja yang terlibat dalam business process?



# SISTEM INFORMASI AKADEMIK

Aljabar Linear



AKUNTANSI



Kalkulus 2



Analisis Numerik  
Matematika Diskrit



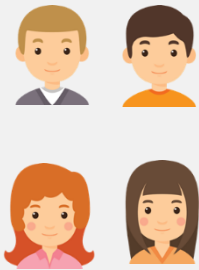
TEKNIK SIPIL  
TEKNIK KIMIA



AKUNTANSI



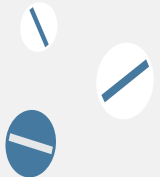
# SISTEM INFORMASI **AKADEMIK**



Analisis Numerik  
Matematika Diskrit  
Aljabar Linear  
Kalkulus 2

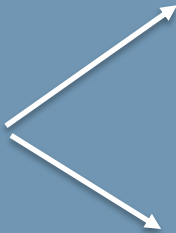


AKUNTANSI  
TEKNIK INFORMATIKA  
TEKNIK SIPIL  
TEKNIK KIMIA





**KARAKTERISTIK**



**Apa yang dimiliki?**

**Apa yang bisa dilakukan?**



Apa yang dimiliki sebuah objek mahasiswa?



- Nama
- NIM
- Tanggal Lahir
- Jenis Kelamin
- Alamat



—**ATRIBUT:**

Variabel/properti/ciri/status  
/sifat yang **dimiliki** oleh  
suatu objek



	productID	productName	categoryName	supplier
▶	1	Chai	Beverages	Specialty Biscuits, Ltd.
	2	Chang	Beverages	Exotic Liquids
	3	Aniseed Syrup	Condiments	Exotic Liquids
	4	Chef Anton's Cajun Seasoning	Condiments	New Orleans Cajun Delights
	5	Chef Anton's Gumbo Mix	Condiments	New Orleans Cajun Delights
	6	Grandma's Boysenberry Spread	Condiments	Grandma Kelly's Homestead
	7	Uncle Bob's Organic Dried Pears	Produce	Grandma Kelly's Homestead
	8	Northwoods Cranberry Sauce	Condiments	Grandma Kelly's Homestead
	9	Mishi Kobe Niku	Meat/Poultry	Tokyo Traders
	10	Ikura	Seafood	Tokyo Traders



## Apa yang bisa dilakukan oleh/terhadap objek mahasiswa?



- Memilih mata kuliah
- Melihat nilai
- Mengajukan cuti akademik

### —METHOD:

Prosedur/fungsi/perilaku/pr  
oses yang bisa **dilakukan**  
oleh/terhadap suatu objek

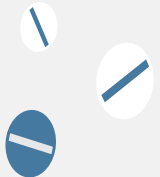
# SISTEM INFORMASI AKADEMIK



Analisis Numerik  
Matematika Diskrit  
Aljabar Linear  
Kalkulus 2



AKUNTANSI  
TEKNIK INFORMATIKA  
TEKNIK SIPIL  
MATEMATIKA

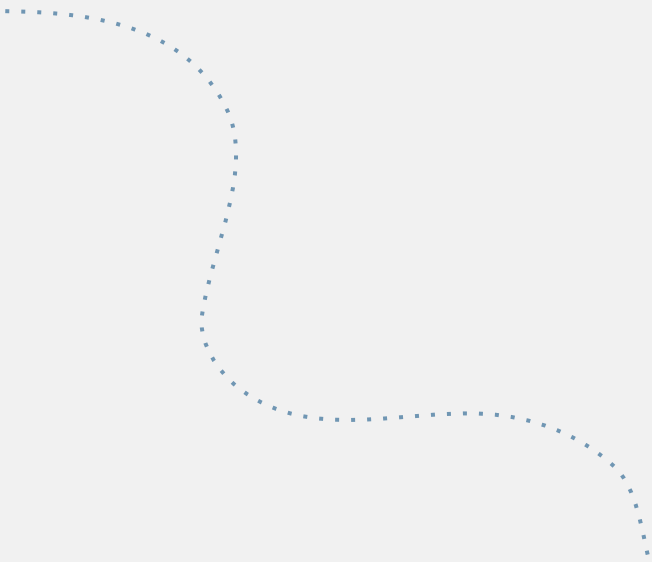






## — **CLASS**

Blueprint/template/cetakan yang mendefinisikan karakteristik (atribut dan method) objek pada class



## — OBJECT

Objek adalah representasi dari setiap entitas yang terlibat dalam sistem (baik yang nyata maupun tidak nyata)

Setiap objek akan memiliki sekumpulan variabel/ciri/status dengan nilai yang melekat padanya

# Class Donat



## 6 Objek pada Class Donat



# Class Rumah

## 3 Objek pada Class Rumah



```
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

# Kesimpulan

## □ Struktural

- ▣ Program dipecah ke dalam fungsi/prosedur
- ▣ Perubahan fitur → kemungkinan mengganggu keseluruhan program

## □ Object Oriented

- ▣ Program dipecah ke dalam object
  - Didalamnya terdapat state dan behavior
- ▣ Perubahan fitur → tidak mengganggu keseluruhan program

# Latihan

- Carilah objek apa saja di dunia nyata sebanyak 5.
- Tuliskan state/atribut dan behavior/method 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

# 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/>