

OOP dan Pustaka Standar

Dr. Bambang Purnomosidi D. P.

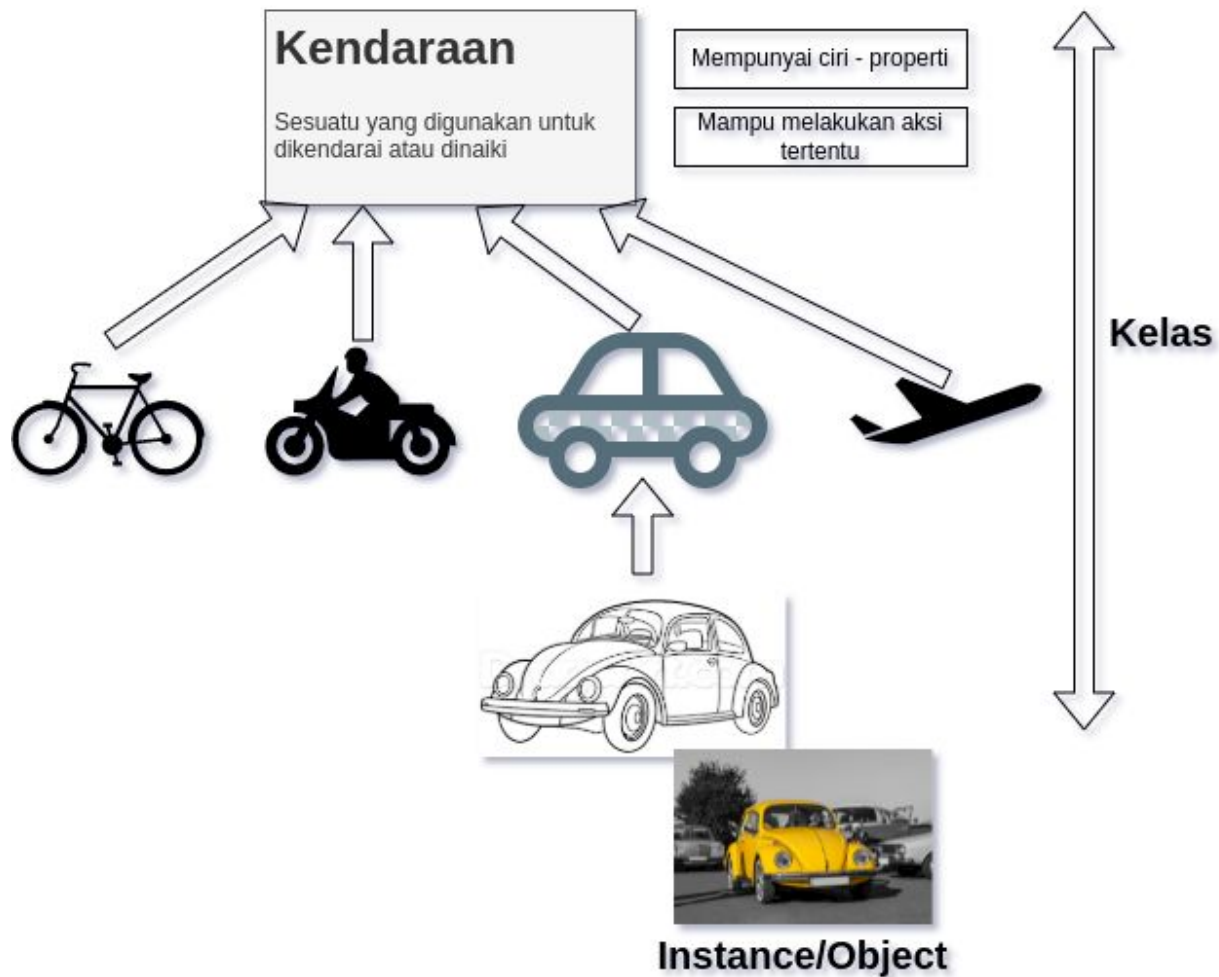


Agenda

- Object-Oriented Programming dan Pola Pikir Obyek
- *CRC Cards*
- OOP di Python
- Python Standard Library

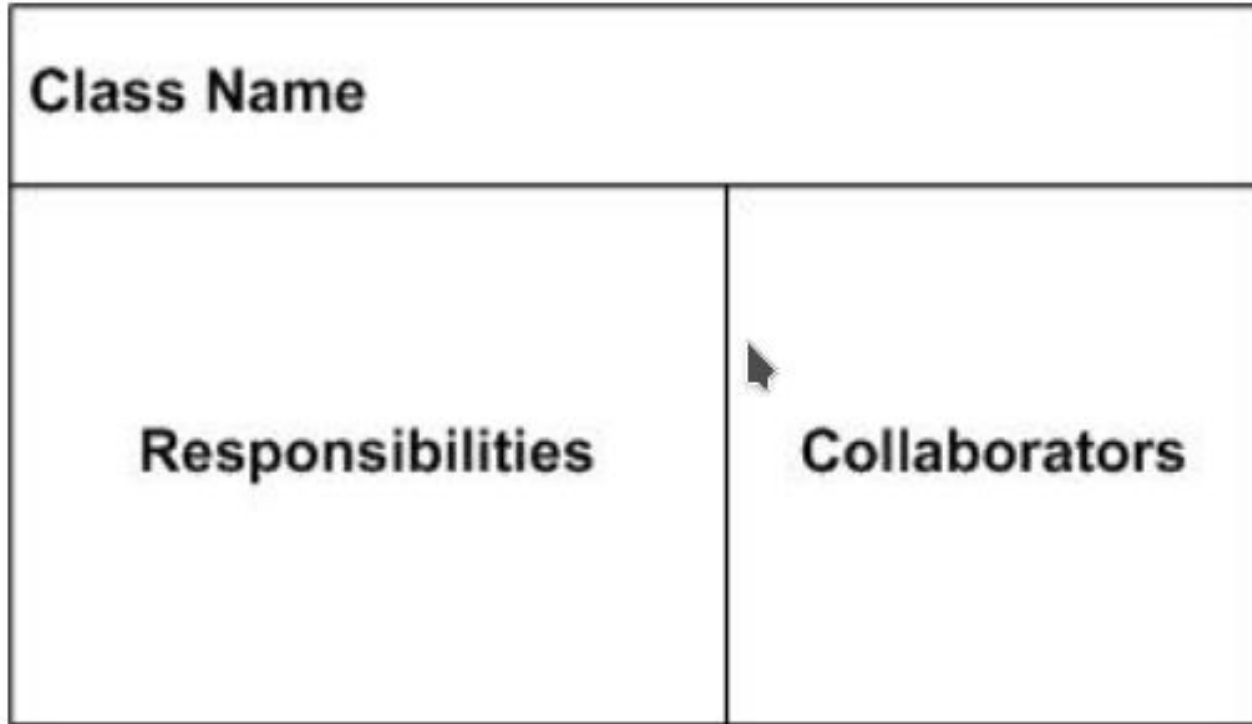
OOP dan Pola Pikir Obyek

- OOP (Object-Oriented Programming) / PBO (Pemrograman Berorientasi Obyek): merupakan salah satu **paradigma pemrograman**.
- **Paradigma**: cara pandang untuk menyelesaikan masalah
- OOP:
 - **Mengikuti pola pikir natural**: suatu sistem dibangun oleh berbagai aktor dan komponen yang mempunyai *properties* dan *action* yang bisa dilakukan oleh aktor/komponen tersebut. Pekerjaan dilakukan dengan mengirim pesan antar aktor / komponen. Komparasi: *job description* adalah *action* yang bisa dikerjakan aktor, untuk bisa mengerjakan *action*, aktor mempunyai berbagai *properties*.
 - **Menekankan pada reusability**: definisi kelas (*blueprint*) dan definisi tersebut bisa dibuat pewarisan untuk membentuk kelas baru. Komparasi: manusia -> mahasiswa -> mahasiswa informatika.



- **Kategori OOP**
 - Class-based: kelas - instance/obyek
 - Prototype-based: instance/obyek sebagai prototype - instance/obyek
- **Message Passing:** pengiriman pesan untuk meng-*invoke behaviour / action*
- **Encapsulation:** menyatukan action dan properties dalam satu unit. Berfungsi untuk menyembunyikan detail sehingga obyek lain tidak bisa mengakses *internal workings*
- **Composition:** suatu obyek terdiri atas obyek(-obyek) lain -> relasi **has-a**
- **Inheritance:** suatu obyek merupakan tipe dari obyek tertentu -> relasi **is-a**
- **Class:**
 - subclass: kelas turunan
 - superclass: kelas induk
 - multiple inheritance: suatu kelas bisa mempunyai turunan lebih dari satu induk
 - mixins: suatu kelas yang berdiri sendiri, diperlukan untuk menambahkan aksi-properti ke kelas lain.
 - abstract class: kelas yang tidak bisa dibuat obyek-nya secara langsung, harus dibuat dulu kelas turunannya.
- **Polymorphism:** satu action - banyak kemungkinan bentuk. Contoh: kelas Shape, inheritance-nya Circle dan Square. Implementasi action *draw* akan menghasilkan hal yang berbeda.

CRC Cards



Student

Student number

Name

Address

Phone number

Enroll in a seminar

Drop a seminar

Request transcripts

Seminar

OOP di Python

- Skeleton:

```
class NamaKelas:
```

```
    var = ...
```

```
# variabel utk semua instances
```

```
    __init__(self, var):
```

```
# constructor
```

```
        ...
```

```
        self.var = ....
```

```
# akses variabel untuk instance
```

```
    def method1(self, var):
```

```
# definisi method / action
```

```
        ...
```

```
        ...
```


- Inheritance dan multiple inheritance:

```
class NamaKelas1:
```

```
...
```

```
class NamaKelas2:
```

```
...
```

```
class NamaKelasBaru(NamaKelas1, NamaKelas2):
```

```
...
```

Python Standard Library

- Pahami konstruksi dasar
- Pahami *function*
- Pahami OOP
- Akses <https://docs.python.org/3/library/index.html>
- Jika akan membuat program dan belum ada function / class standar yang mendukung, maka cari pustaka pihak ketiga di <https://pypi.org/>