OBJECT & CLASS

Tujuan Instruksional :
Bagian ini berisi materi mengenai object dan class pada bahasa Python.

Kompetensi yang Diharapkan :
Mahasiswa diharapkan mampu memahami konsep OOP pada Python.

Waktu Pertemuan : 100 menit

1. Object Oriented Programming (OOP)

OOP (*Object Oriented Programming*) atau dalam bahasa indonesia dikenal dengan pemrograman berorientasikan objek (PBO) merupakan sebuah paradigma atau teknik pemrograman yang berorientesikan Objek.

Pada OOP, Fungsi dan variabel **dibungkus** dalam sebuah **objek** atau *class* yang dapat saling brinteraksi, sehingga membentuk sebuah program.

2. Class & Object

Class adalah rancangan atau *blue print* dari sebuah objek. Sedangkan objek dalam pemrograman adalah sebuah variabel yang merupakan *instance* dari *Class*.

Instance bisa diartikan sebagai wujud dari *class*.

Class berisi definisi variabel dan fungsi yang menggambarkan sebuah objek.

Dalam OOP:

- **Variabel** disebut **atribut** atau **properti**;
- **Fungsi** disebut **method**.

<Syntax>
```
class ClassName:
    statements

object = ClassName()
```
<Contoh>
```
class Orang:
      pass

gayus = Orang()
```

3. Attributes

Attributes seperti properties yang ingin kita tambahkan ke dalam class. Sebagai contoh, untuk class Orang, mari tambahkan dua attributes: nama dan sekolah, seperti di bawah ini:

<Contoh>

```
class Orang:
    nama = ''
    sekolah = ''

gayus = Orang()
gayus.nama = 'Gayus Tambunan'
gayus.sekolah = 'Pangudi Luhur'
```

4. Method

Methods sama seperti functions dalam Python, dalam hal mereka ditentukan dengan kata kunci def dan memiliki format yang sama seperti functions. Dalam class kita, mari tentukan sebuah method yang mencetak nama dan sekolah orang. Class akan tampak sebagai berikut:

<Contoh>

```
class Orang:
    nama = ''
    sekolah = ''
    def tidur(self):
        print(self.nama,'sedang tidur')

    def mandi(self, sama):
        print(self.nama,'lagi mandi sama',sama)



gayus = Orang()
gayus.nama = 'Gayus Tambunan'
gayus.sekolah = 'Pangudi Luhur'
gayus.tidur()
gayus.mandi('Lucinta Luna')
```

5. Inisialisasi

Dalam section sebelumnya, kita menginisialisasi nama dan sekolah dengan memberikan mereka sebuah nilai kosong ' '. Namun ada cara yang lebih elegan untuk menginisialisasi variabel ke nilai default, dan ini dimana *inisialisasi* berguna.

<Contoh>

```
class Orang:
    def __init__(self, name, school):
        self.nama = name
        self.sekolah = school
    def tidur(self):
        print(self.nama,'sedang tidur')

    def mandi(self, sama):
        print(self.nama,'lagi mandi sama',sama)


fadli = Orang('Fadli Zon', 'JIS')
print(fadli.name)
```