


<p>Nama: Zahwa Nur Azkia Putri</p> <p>NIM: 064002300038</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<p>MODUL 11</p> <p>Nama Dosen: Ratna Shofiati, S.Kom, M. Kom</p>
<p>Hari/Tanggal: Rabu, 13 Desember 2023</p>		<p>Nama Asisten Labratorium: 1. Yuda Hadi Prasetyo – 065002100004 2. Muhammad Hasan Husein – 065002100009</p>

Object Oriented Programming pada Python

1. Teori Singkat

Object Oriented Programming atau Pemrograman Berorientasi Objek merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk field atau dikenal juga sebagai atribut serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai method. Python telah menjadi bahasa berorientasi objek sejak bahasa Python sendiri dibuat.

Class

Class adalah prototype, atau blueprint, atau rancangan yang mendefinisikan variable dan method-method pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan di jalankan, di dalamnya berisi atribut / type data dan method untuk menjalankan suatu program. Dalam Python sendiri class didefinisikan dengan keyword class dan diikuti oleh penamaan kelas tersebut "*class nama_kelas*". Pemanggilan kelas sendiri sama seperti pemanggilan sebuah fungsi/method dalam sebuah program yaitu memanggil nama class tersebut beserta parameter classnya. Biasanya class berisi banyak method/fungsi yang merupakan turunan sifat dari kelas tersebut.

Class sendiri memiliki banyak bentuk dalam setiap Bahasa pemrograman yang berbeda, seperti abstrak class, data class dan lain sebagainya. Class juga dapat memiliki keterkaitan dengan class lainnya yang dapat disebut sebagai class turunan atau inheritance, Inheritance merupakan sebuah hubungan Parent Class (Kelas Induk) dengan Child Class (Kelas Anak) yang dimana memiliki pewarisan sifat dan pewarisan variabel turunan yang sama.



Contoh Program OOP sederhana pada Python

```
class Employee:
    'Common base class for all employees'
    empCount = 0

    def __init__(self, name, salary):
        self.name = name
        self.salary = salary
        Employee.empCount += 1

    def displayCount(self):
        print ("Total Employee %d" % Employee.empCount)

    def displayEmployee(self):
        print ("Name : ", self.name, ", Salary: ", self.salary)

# Deklarasi Objek Pertama dari Employee Class
emp1 = Employee("Zara", 2000)
# Deklarasi Objek Kedua dari Employee Class
emp2 = Employee("Manni", 5000)
emp1.displayEmployee()
emp2.displayEmployee()
print ("Total Employee %d" % Employee.empCount)
```

Output

```
Name:  Zara , Salary:  2000
Name:  Manni , Salary:  5000
Total Employee 2
```



2. Alat dan Bahan

Hardware : Laptop/PC

Software : Spyder (Anaconda Python)

3. Elemen Kompetensi

a. Latihan pertama

Buatlah sebuah program yang mengimplementasikan sebuah class yang memiliki nama class mahasiswa dan memiliki method yang dapat digunakan untuk menampilkan biodata mahasiswa yang diinputkan oleh user.

Contoh Output

```
C:\Windows\System32\cmd.exe X + v
C:\Users\Yuda Hp\Aslab Algo\Praktikum 11>python Modul11Latihan1.py
Masukkan Namamu: Yuda
Masukkan NIM kamu: 065002100004
Masukkan Tahun Angkatanmu: 2021

Nama: Yuda
Nim: 065002100004
Angkatan: 2021

Total Mahasiswa 1
C:\Users\Yuda Hp\Aslab Algo\Praktikum 11>
```

Source Code

```
class Mahasiswa:
    mhsCount = 0

    def __init__(self, name, nim, angkatan):
        self.name = name
        self.nim = nim
        self.angkatan = angkatan
        Mahasiswa.mhsCount += 1

    def displaycount(self):
        print ("total mahasiswa %d" % Mahasiswa.mhscount)

    def displaymahasiswa(self):
        print("nama: ", self.name)
        print("nim: ", self.nim)
        print("angkatan: ", self.angkatan)
```



```
name = input("masukan nama: ")
nim = input("masukan nim: ")
angkatan = input("masukan angkatan: ")

print("\n")
mhs1 = Mahasiswa(name, nim, angkatan)
mhs1.displayMahasiswa()

print("\n")
print("total Mahasiswa %d" % Mahasiswa.mhsCount)
```

Output

```
Masukkan nama: zahwa
Masukkan NIM: 38
Masukkan angkatan: 2023
Nama: zahwa
NIM: 38
Angkatan: 2023
Jumlah mahasiswa 1
```



b. Latihan Kedua

Buatlah sebuah kelas yang menerapkan method getter dan setter dimana menggunakan implementasi program percabangan serta perulangan seperti pada Latihan sebelumnya. Program menerima deklarasi nilai inputan dari user dan menampungnya dalam sebuah kelas dan variabel didalam kelas tersebut dapat dimanipulasi serta dirubah sesuai keinginan dan perubahan yang diberikan oleh user melalui inputan user itu sendiri serta dapat ditampilkan menggunakan method getter dan setter.

Contoh Output

Menambahkan & menampilkan objek

```
C:\Users\Yuda Hp\Aslab Algo\Praktikum 11>python Modul11Latihan2.py
===== Program OOP =====
1. Mendeklarasikan Objek
2. Menampilkan Objek
3. Merubah Nilai Objek
4. Menghapus Objek
5. Keluar Dari Program

Masukkan Pilihan Berupa Angka (1/2/3/4/5): 1
Masukkan Namamu: Yuda
Masukkan Nilaimu: 100
Data Berhasil Ditambahkan

===== Program OOP =====
1. Mendeklarasikan Objek
2. Menampilkan Objek
3. Merubah Nilai Objek
4. Menghapus Objek
5. Keluar Dari Program

Masukkan Pilihan Berupa Angka (1/2/3/4/5): 2

Nama: Yuda
Nilai: 100
```

Mengubah & menampilkan objek

```
Masukkan Pilihan Berupa Angka (1/2/3/4/5): 3
Apa yang ingin diubah (Nama/Nilai): Nilai
Masukkan Nilai: 95
Data Nilai Berhasil Dirubah

===== Program OOP =====
1. Mendeklarasikan Objek
2. Menampilkan Objek
3. Merubah Nilai Objek
4. Menghapus Objek
5. Keluar Dari Program

Masukkan Pilihan Berupa Angka (1/2/3/4/5): 3
Apa yang ingin diubah (Nama/Nilai): Nama
Masukkan Nama: Hasan
Data Nama Berhasil Dirubah

===== Program OOP =====
1. Mendeklarasikan Objek
2. Menampilkan Objek
3. Merubah Nilai Objek
4. Menghapus Objek
5. Keluar Dari Program

Masukkan Pilihan Berupa Angka (1/2/3/4/5): 2

Nama: Hasan
Nilai: 95
```



Menghapus & menampilkan objek

```
===== Program OOP =====
1. Mendeklarasikan Objek
2. Menampilkan Objek
3. Merubah Nilai Objek
4. Menghapus Objek
5. Keluar Dari Program

Masukkan Pilihan Berupa Angka (1/2/3/4/5): 4
Data Berhasil Dihapus

===== Program OOP =====
1. Mendeklarasikan Objek
2. Menampilkan Objek
3. Merubah Nilai Objek
4. Menghapus Objek
5. Keluar Dari Program

Masukkan Pilihan Berupa Angka (1/2/3/4/5): 2

Nama: None
Nilai: None
```

Keluar dari program

```
===== Program OOP =====
1. Mendeklarasikan Objek
2. Menampilkan Objek
3. Merubah Nilai Objek
4. Menghapus Objek
5. Keluar Dari Program

Masukkan Pilihan Berupa Angka (1/2/3/4/5): 5
Terima Kasih Sudah Menggunakan Program Saya
```

Source Code

```
print("Isi dengan Source Code kalian ya...")
```



Output

```
print("Screenshot hasil keluaran atau output dari source code kalian ya...")
```

4. File Praktikum

Github Repository:

```
print("Taruh link github repository praktikum kalian disini yaa...")
```

5. Soal Latihan

Soal:

1. Jelaskan apa itu method getter dan setter pada sebuah class dan apa kegunaan serta fungsi method getter dan setter dalam sebuah class?
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. (Isi Dengan Jawaban Kalian)
2. (Isi Dengan Jawaban Kalian)

6. Kesimpulan

- a. Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
- b. Kita dapat mengetahui... (Tolong Isi lebih dari dua baris!)



7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	✓	
2.	Latihan Kedua		

8. Formulir Umpan Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	20 Menit	Baik
2.	Latihan Kedua	... Menit	...

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang

