


Nama: Muhammad Ali Akbar Hidayatullah NIM: 064002300034	 Praktikum Algoritma & Pemrograman	MODUL 11 Nama Dosen: Ratna Shofiati, S.Kom, M. Kom
Hari/Tanggal: Rabu 13 Desember 2023		Nama Asisten Labratorium: 1. Yuda Hadi Prasetyo – 065002100004 2. Muhammad Hasan Husein – 065002100009

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

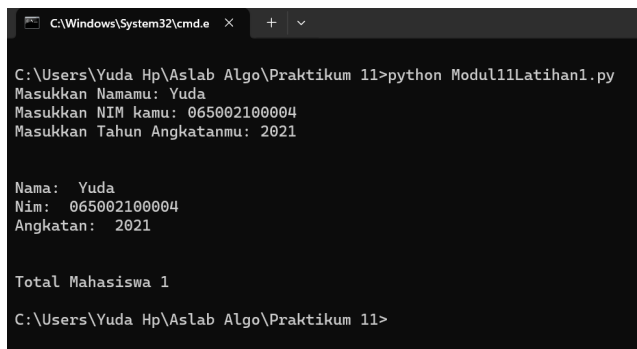
Software : Spyder (Anaconda Python)

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.e  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:
    total_mahasiswa = 0 # Variabel kelas untuk melacak total mahasiswa

    def __init__(self, nama, nim, angkatan): #sebagai konstruktor untuk menginisialisasi
atribut kelas
        self.nama = nama
        self.nim = nim
        self.angkatan = angkatan
        Mahasiswa.total_mahasiswa += 1 #menambahkan 1 setiap kali objek mahasiswa
baru dibuat.

    def tampilkan_biodata(self): #mencetak biodata mahasiswa.
        print("\nNama:", self.nama)
        print("NIM:", self.nim)
        print("Angkatan:", self.angkatan)
```



```
class mahasiswa:
    mhsCount = 0

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

    def tampilkanjumlah(self):
        print("Total Mahasiswa %d" % mahasiswa.mhsCount)

    def tampilkanmahasiswa(self):
        print("Nama: ", self.name)
        print("Nim: ", self.nim)
        print("Angkatan: ", self.angkatan)

name = input("Masukan nama: ")
nim = input("Masukan nim: ")
angkatan = input("Masukan tahun angkatan: ")

print("\n")
mhs1 = mahasiswa(name, nim, angkatan)
mhs1.tampilkanmahasiswa()

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

Output



```
Masukan nama: Muhammad Ali Akbar
Masukan nim: 064002300034
Masukan tahun angkatan: 2023

Nama: Muhammad Ali Akbar
Nim: 064002300034
Angkatan: 2023

total 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

```
class ObjekProgram:
    def __init__(self):
        self.nama = None
        self.nilai = None

    def set_nama(self, nama):
        self.nama = nama

    def get_nama(self):
        return self.nama if self.nama is not None else "None"

    def set_nilai(self, nilai):
        self.nilai = nilai

    def get_nilai(self):
        return self.nilai if self.nilai is not None else "None"

# Fungsi untuk menampilkan menu
def tampilkan_menu():
    print("\n===== Program OOP =====")
    print("1. Mendeklarasikan Objek")
    print("2. Menampilkan Objek")
    print("3. Merubah Nilai Objek")
    print("4. Menghapus Objek")
```



```
print("5. Keluar Dari Program")

# Fungsi untuk mendapatkan pilihan pengguna
def ambil_pilihan():
    return input("Masukkan Pilihan Berupa Angka (1/2/3/4/5): ")

# Fungsi utama
def main():
    objek_program = ObjekProgram()

    while True:
        tampilkan_menu()
        pilihan = ambil_pilihan()

        if pilihan == '1':
            nama_input = input("\nMasukkan Namamu: ")
            nilai_input = input("Masukkan Nilaimu: ")

            objek_program.set_nama(nama_input)
            objek_program.set_nilai(nilai_input)

            print("\nData Berhasil Ditambahkan")

        elif pilihan == '2':
            print(f"\nNama: {objek_program.get_nama()}")
            print(f"\nNilai: {objek_program.get_nilai()}")

        elif pilihan == '3':
            ubah_apa = input("\nApa yang ingin diubah (Nama/Nilai): ")
            if ubah_apa.lower() == 'nama':
                nama_input = input("Masukkan Nama Baru: ")
                objek_program.set_nama(nama_input)
                print("\nData Nama Berhasil Dirubah")
            elif ubah_apa.lower() == 'nilai':
                nilai_input = input("Masukkan Nilai Baru: ")
                objek_program.set_nilai(nilai_input)
                print("\nData Nilai Berhasil Dirubah")

        elif pilihan == '4':
```




```
        objek_program = ObjekProgram()
        print("\nData Berhasil Dihapus")

    elif pilihan == '5':
        print("\nTerima Kasih Sudah Menggunakan Program Saya")
        break

if __name__ == "__main__":
    main()
```

Output

```
===== 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: Akbar
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: Akbar
Nilai: 100
```



```
===== 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): Nilai
Masukkan Nilai Baru: 99

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): 2

Nama: Akbar
Nilai: 99

===== 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): 5

Terima Kasih Sudah Menggunakan Program Saya
```

4. File Praktikum

Github Repository:

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. Metode getter dan Setter merupakan metode dalam pemrograman Algoritma yang biasanya dipakai untuk akses dan modifikasi suatu variabel objek.
kegunaan dan fungsi dari metode getter dan setter :
kegunaan dari metode getter yaitu mampu mengambil suatu nilai atribut objek tertentu sedangkan fungsinya yaitu sebagai sumber akses read ke suatu atribut objek
kegunaan dari metode setter yaitu mampu mengubah suatu nilai atribut objek sedangkan fungsinya yaitu memberi akses tulis, modifikasi atau validasi saat ingin mengubah sebuah nilai pada atribut objek
2. diawal kita harus mendefinisikan kelas mahasiswa dengan variabel kelas mhsCount yang diinisialisasi dengan nilai 0, ditujukan untuk menghitung total jumlah lalu kita masukkan informasi mahasiswa dan terakhir untuk total mahasiswa ditampilkan setelah objek dibuat menggunakan variabel mhsCount

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 banyak wawasan di modul ini yang sedang membahas tentang objek orientasi programing termasuk juga metode getter dan metode setter keduanya sangat berguna karna bisa merubah serta menambah objek nilai.

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	30 Menit	Menarik
2.	Latihan Kedua	30 Menit	Menarik

Keterangan:

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

