



MODUL PRAKTIKUM

DASAR PEMROGRAMAN

INF1005

Penyusun :

Naufal Azmi Verdikha, M.Eng.

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2018

Praktikum 11:

Dasar Pemrograman Berorientasi Objek

Pokok Bahasan:

- ❖ Dasar Pemrograman Berorientasi Objek

Tujuan Pembelajaran:

- ✓ Mengimplementasikan dasar-dasar pemrograman berorientasi objek dari *Python 3*

Dasar Pemrograman Berorientasi Objek:

1. Kelas

Kelas merupakan suatu “cetak biru”/*blueprint* dari suatu objek. Kita dapat membayangkan suatu motor dengan merek X dengan spesifikasi dan desain tertentu merupakan suatu cetak biru sebelum diproduksi yang dapat dianggap sebagai kelas dan satu buah motor yang telah diproduksi ialah suatu objek.

Percobaan & Latihan 11.1

Jalankan Perintah Berikut :

```
class Pekerja:
    pkTotal = 0

    def __init__(self, nama, gaji):
        self.nama = nama
        self.gaji = gaji
        Pekerja.pkTotal += 1

    def displayTotal(self):
        print("Total Pekerja %d" %Pekerja.pkTotal)

    def displayPekerja(self):
        print("Nama : ", self.nama, ", Gaji : ", self.gaji)
```

Soal :

- Buatlah program sesuai perintah diatas!
- Berikan penjelasan menggunakan komentar pada tiap baris syntax
- SS syntax yang sudah diberikan penjelasan didalamnya.

2. Objek

Suatu objek merupakan instansiasi dari kelas. Suatu kelas hanya mendefinisikan deskripsi dari objek, sedangkan objek ialah “penerapan” atau perwujudan dari kelas tersebut dan terdapat pada memori.

Percobaan & Latihan 11.2

Jalankan Perintah Berikut :

```
class Pekerja:
    pkTotal = 0

    def __init__(self, nama, gaji):
        self.nama = nama
        self.gaji = gaji
        Pekerja.pkTotal += 1

    def displayTotal(self):
        print("Total Pekerja %d" %Pekerja.pkTotal)

    def displayPekerja(self):
        print("Nama : ", self.nama, ", Gaji : ", self.gaji)
```

Soal :

- a) Buatlah Objek dari Program 11.1 yang sudah dibuat dan Screenshot hasilnya!

3. Method

Dalam suatu kelas terdapat *behaviour* atau method. Method dapat disamakan dengan fungsi namun suatu method terdapat pada kelas sehingga dimiliki oleh kelas tersebut. Dalam pendeklarasian kelas juga mirip ketika mendeklarasikan fungsi, hanya saja method pada kelas wajib memiliki parameter **self** yang merujuk pada objek dari kelas itu sendiri.

Percobaan & Latihan 11.3

Jalankan Perintah Berikut :

```
class Pekerja:
    pkTotal = 0

    def __init__(self, nama, gaji):
        self.nama = nama
        self.gaji = gaji
        Pekerja.pkTotal += 1

    def displayTotal(self):
        print("Total Pekerja %d" %Pekerja.pkTotal)

    def displayPekerja(self):
        print("Nama : ", self.nama, ", Gaji : ", self.gaji)
```

Soal :

- Gunakan salah satu method yang ada didalam class **11.1** yang sudah dibuat sebelumnya!
- Jelaskan fungsi method yang digunakan!

4. Constructor

Pada pemrograman berbasis objek umumnya terdapat *constructor* yang merupakan method khusus yang akan pertamakali dijalankan ketika suatu kelas diinstansiasi (diciptakan). Pada Python suatu method *constructor* didefinisikan dengan nama `__init__()`. Method constructor memungkinkan untuk memberikan suatu nilai ketika melakukan instansiasi objek, hal ini dimungkinkan dengan memberikan parameter. Samahalnya dengan method pada umumnya terdapat parameter *self* yang bersifat wajib, dan dapat diberi parameter-parameter lainnya dengan cara yang sama dengan method pada umumnya.

Percobaan & Latihan 11.4

Jalankan Perintah Berikut :

```
class Pekerja:
    pkTotal = 0

    def __init__(self, nama, gaji):
        self.nama = nama
        self.gaji = gaji
        Pekerja.pkTotal += 1

    def displayTotal(self):
        print("Total Pekerja %d" %Pekerja.pkTotal)

    def displayPekerja(self):
        print("Nama : ", self.nama, ", Gaji : ", self.gaji)
```

Soal :

- a) Jelaskan fungsi constructor pada program diatas!

Laporan Resmi:

1. Buatlah hasil dan analisa dari Percobaan & Latihan 11.1 sampai dengan **11.4**
2. Berikan kesimpulan dari praktikum 11.