



Modul Praktikum

Algoritma dan Pemrograman

Jan Wantoro, S.T., M.Eng.

Irma Yuliana, S.T., M.M., M.Eng.

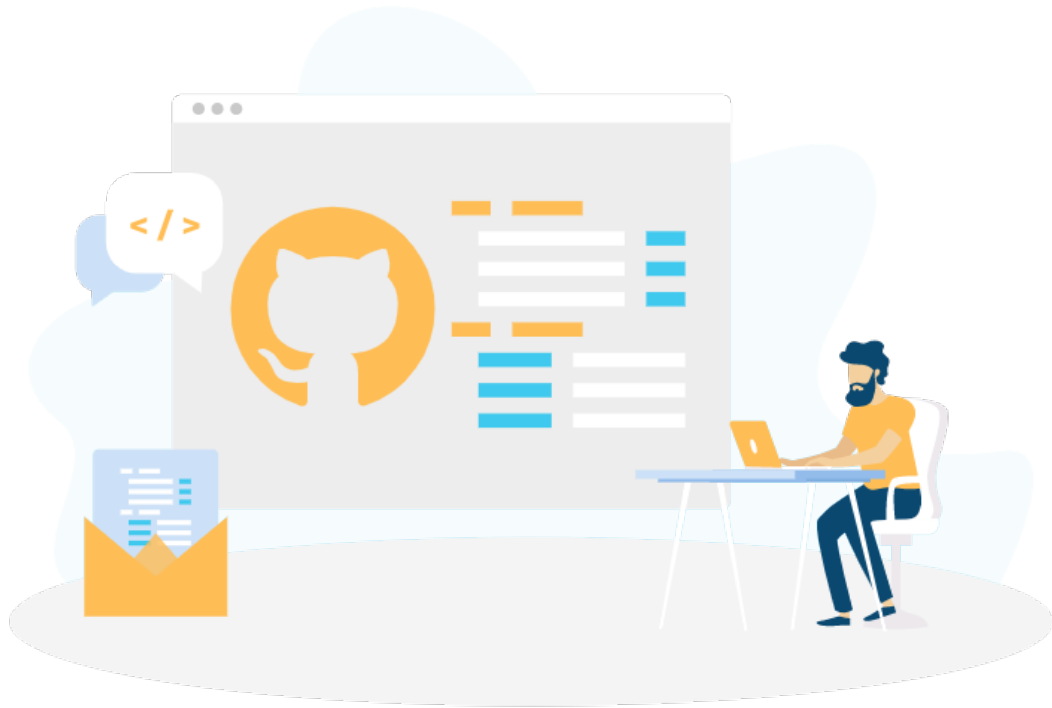
Arif Setiawan, S.Kom., M.Eng.

Program Studi Pendidikan Teknik Informatika

Fakultas Keguruan dan Ilmu Pendidikan Universitas Muhammadiyah Surakarta

BAB 9

Class dan Object



9.1 Tujuan

1. Dapat menjelaskan class dan object pada bahasa python.
2. Dapat mengimplementasikan class dan object pada pemrograman bahasa python

9.2 Pengantar

Pemrograman Berorientasi Object (PBO) merupakan salah satu paradigma yang diterapkan hampir di seluruh bahasa pemrograman. Konsep dasar dari PBO adalah mengumpulkan data dan fungsi yang memiliki hubungan kedalam suatu pulau informasi. Pulau ini disebut dengan object.

Jika dibandingkan dengan bahasa pemrograman prosedural, PBO akan melihat suatu masalah secara keseluruhan. Dalam bahasa pemrograman prosedural suatu masalah akan dipecahkan dengan cara memanggil prosedur yang biasa disebut dengan function. Dalam PBO, alih-alih berurusan dengan data secara langsung PBO akan memahami data mana yang akan digunakan dengan cara melakukan modeling. Untuk melakukan modeling ini ada beberapa istilah yang perlu dipahami yaitu class dan object

9.2.1 Pengenalan Class dan Object

Class merupakan sebuah blueprint dari object yang akan kita buat. Class berarti cetakannya sedangkan object (instance) adalah hasil dari cetakan tersebut. Untuk memahami secara lebih jelas perhatikan tabel berikut ini.

Tabel 8.1 Contoh Class dan Object

Class	Object/Instance
Car	Honda Jazz, Toyota Avanza, Suzuki Jimny
Cat	Persia, Siam, Kucing Bengal
Coffe	Americano, Capucino, Late
Dog	Labrador, Husky, Bulldog, Doberman

Untuk membuat sebuah class kita dapat menggunakan kode berikut ini. Nama sebuah kelas harus diawali dengan huruf kapital.

```
1. class NamaKelas:  
2.     # atribut atau metode yang digunakan di class ini
```

Contoh dalam pembuatan class sebuah Dog

```
1. class Dog:
2.     def __init__(self, nama, umur):
3.         self.nama = nama
4.         self.umur = umur
5.
6.     def duduk(self):
7.         print(f"{self.nama} sekarang duduk")
8.
9.     def berdiri(self):
10.        print(f"{self.nama} sekarang berdiri")
11.
```

Kode diatas akan dijelaskan pada sub bab dibawah ini

9.2.2 Method `__init__`

Fungsi yang berada dalam sebuah class dinamakan dengan method. Semua aturan fungsi yang sudah kita pelajari di bab sebelumnya berlaku juga pada pembuatan method. Jika kita lihat pada baris ke-2 terdapat method `__init__`, method ini merupakan spesial method yang secara otomatis akan berjalan setiap kali object/instance dari class Dog dibuat. Penulisan method ini diawali dengan dua kali underscore (`_`) dan diakhiri juga dengan dua kali underscore (`_`). Tanpa penulisan dua kali underscore di awal dan di akhir ini maka python tidak akan menjalankan method ini secara otomatis.

9.2.3 Parameter `self`

Pada method `__init__` terdapat tiga parameter yaitu `self`, `nama` dan `umur`. Parameter `self` merupakan sebuah parameter yang harus ada didalam pembuatan method, dan harus ditulis di awal sebelum parameter lainnya. Jika kita lihat pada method `duduk` dan `berdiri`, parameter `self` harus ditulis walaupun tidak ada parameter lain di dalam method tersebut.

Parameter `self` ini berfungsi untuk mendapatkan akses secara internal terhadap atribut atau method didalam sebuah class saat kita membuat object/instance.

9.2.4 Membuat Object/Instance dari class Dog

Jika sebuah class adalah blueprint maka object adalah hasil cetakannya. Mari kita buat object dari class Dog

```
1. class Dog:
2.     def __init__(self, nama, umur):
3.         self.nama = nama
4.         self.umur = umur
5.
6.     def duduk(self):
7.         print(f"{self.nama} sekarang duduk")
8.
9.     def berdiri(self):
10.        print(f"{self.nama} sekarang berdiri")
11.
12. my_dog = Dog("Labrador", 6)
13.
14. print(f"anjingku bernama {my_dog.nama}")
15. print(f"anjingku berumur {my_dog.umur} tahun ")
```

Baris ke 12 merupakan cara pembuatan object my_dog berdasarkan class Dog. Pembuatan object harus menggunakan aturan lowercase. Pada pembuatan object my_dog kita mengirim dua variabel yaitu Labrador dan 6. Parameter ini disesuaikan dengan kebutuhan method __init__ pada class Dog. Ingat parameter self akan diproses secara otomatis sehingga kita hanya perlu mengirim variabel untuk parameter nama dan umur. Jika baris ke 12 ini dijalankan maka python akan membuat sebuah object bernama my_dog dengan nama = Labrador dan umur = 6

9.2.5 Mengakses Attribute

Untuk mengakses atribut pada sebuah objek/instance kita menggunakan bantuan titik (.) perhatikan baris ke 14 dan 15 dari kode diatas. Saat akan mengakses atribut nama dari object my_dog penulisannya menjadi

```
1. {my_dog.nama}
```

Dengan bantuan titik (.) disini maka python akan mencari objek dengan nama my_dog kemudian mencari attribute nama di object my_dog tersebut. Jika baris 14 dan 15 dijalankan maka akan menghasilkan output seperti berikut

```
anjingku bernama Wili  
anjingku berumur 6 tahun
```

9.2.6 Memanggil Method

Jika kita sudah membuat object dari suatu class maka kita dapat menggunakan method yang ada pada class tersebut. Perhatikan kode dibawah ini

```
1. class Dog:  
2.     def __init__(self, nama, umur):  
3.         self.nama = nama  
4.         self.umur = umur  
5.  
6.     def duduk(self):  
7.         print(f"{self.nama} sekarang duduk")  
8.  
9.     def berdiri(self):  
10.        print(f"{self.nama} sekarang berdiri")  
11.  
12. my_dog = Dog("Wili",6)  
13.  
14. print(f"anjingku bernama {my_dog.nama}")  
15. print(f"anjingku berumur {my_dog.umur} tahun")  
16.  
17. my_dog.duduk()  
18. my_dog.berdiri()
```

Perhatikan baris ke 17 dan 18, hampir mirip seperti pengaksesan atribut, pemanggilan method juga menggunakan bantuan titik (.) Kita tulis nama object terlebih dahulu, beri tanda titik, diakhiri dengan nama method yang akan dipanggil. Jika kode tersebut dijalankan maka akan menghasilkan output seperti berikut :

```
anjingku bernama Wili  
anjingku berumur 6 tahun  
Wili sekarang duduk  
Wili sekarang berdiri
```

9.3 Kegiatan Praktikum

9.3.1 Kegiatan Praktikum 1 : Class dengan default Attribute

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. class Car:
2.     def __init__(self, merk, model, tahun):
3.         self.merk = merk
4.         self.model = model
5.         self.tahun = tahun
6.         self.odometer = 0
7.
8.     def keterangan(self):
9.         print(f"Mobil baru saya {self.merk} {self.model} tahun
10.             {self.tahun} kilometernya masih {self.odometer}")
11. mobil_baru = Car('Honda', 'City', 2021)
12. mobil_baru.keterangan()
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 1 ini

9.3.2 Kegiatan Praktikum 2 : Multiple Object

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. class Kucing():
2.     def __init__(self, ras, umur):
3.         self.ras = ras
4.         self.umur = umur
5.
6.     def tampil(self):
7.         print(f"Kucingku berjenis {self.ras} dengan umur
8.             {self.umur} bulan")
9. kucing_saya = Kucing("persia", 2)
10. kucing_dia = Kucing("anggora", 3)
11.
12. kucing_saya.tampil()
13. kucing_dia.tampil()
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 2 ini

9.3.3 Kegiatan Praktikum 3 : Attribute Class yang merupakan Instance dari Class lain

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. class Kucing():
2.     def __init__(self, ras, umur):
3.         self.ras = ras
4.         self.umur = umur
5.
6.
7. class Pemilik():
8.     def __init__(self, nama, alamat, kucing):
9.         self.nama = nama
10.        self.alamat = alamat
11.        self.kucing = kucing
12.
13.    def tampil(self):
14.        print(f"Halo, aku {self.nama} berasal dari
15.        {self.alamat}")
16.        print(f"Kucingku berjenis {self.kucing.ras} dengan umur
17.        {self.kucing.umur} bulan")
18.
19.
20. arif = Pemilik(
21.     nama = "Arif",
22.     alamat = "Solo",
23.     kucing = Kucing(
24.         ras = "Persia",
25.         umur = 7
26.     )
27. )
28. arif.tampil()
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 3 ini

9.4 Tugas

1. Buatlah sebuah class dengan nama Mahasiswa. Buat 4 atribut dengan nama "nim", "nama", "angkatan", dan "prodi". Buatlah sebuah method dengan nama

"kartu_mahasiswa" yang akan menampilkan data mahasiswa tersebut dan method "selamat" yang menampilkan ucapan " Selamat datang {nama} di kampus UMS".

Buatlah minimal 3 objek dari class tersebut dan panggil 2 method tersebut untuk tiap objeknya

DAFTAR PUSTAKA

- Hunt, J. (2019). *A Beginners Guide to Python 3 Programming*. In Springer
- Romano, Fabrizio. (2015). *Learning Python*. Packt Publishing.
- Swastika, W. (2019). *Pengantar Algoritma dan Penerapannya pada Python*. Ma Chung Press.
- Wadi, H. *Pemrograman Python untuk Mahasiswa dan Pelajar*. TR Publisher
- Python File Handling Tutorial: How to Create, Open, Read, Write, Append*. (n.d.). Retrieved September 7, 2020, from <https://www.softwaretestinghelp.com/python/python-file-reading-writing/>
- Learn Python Programming*. (n.d.). Retrieved September 7, 2020, from <https://www.programiz.com/python-programming>
- Tutorial Pemrograman Python*. (n.d.). Retrieved September 5 , 2020, from <https://www.petanikode.com/tutorial/python/>
- PY4E - Python for Everybody*. (n.d.). Retrieved September 2, 2020, from <https://www.py4e.com/>
- Google's Python Class | Python Education | Google Developers*. (n.d.). Retrieved September 9, 2020, from <https://developers.google.com/edu/python/>
- Learn Python the Hard Way*. (n.d.). Retrieved September 2, 2020, from <https://learnpythonthehardway.org/book/>
- Python Programming Tutorials*. (n.d.). Retrieved September 9, 2020, from <https://pythonprogramming.net/python-fundamental-tutorials/>

LAMPIRAN

Laporan Sementara

Laporan sementara dilakukan dengan langkah sebagai berikut:

1. Buatlah screenshot atau gambar dari hasil kegiatan yang dilakukan.
2. Ganti nama file dengan format: bab-kegiatan-no.gambar-nim.jpg Misalnya file adalah gambar ke 1 pada Bab 1, Kegiatan 1, dan nim anda adalah A40009001, maka nama filenya adalah **1-1-1-A40009001.jpg**
3. Simpan file tersebut kemudian tempatkan pada folder sesuai instruksi dosen/asisten praktikum.

Laporan Praktikum

Laporan ditulis dalam kertas putih ukuran A4. Sedangkan urutan susunan laporan adalah sebagai berikut:

1. Cover depan: Berwarna sama dengan cover modul praktikum
2. Halaman Cover: Contoh dapat di-download di <http://bit.ly/lap-alpro>
3. Kata Pengantar
4. Daftar isi
5. Laporan tiap modul (1-10) sesuai dengan format terlampir
6. Penulis: berisi biodata penulis (disertai foto), pesan dan kesan, kritik dan saran demi kemajuan praktikum berikutnya.

Format Laporan Tiap Bab

Matakuliah : Algoritma dan Pemrograman	Acc
NIM :	
Nama :	
Tgl. Prakt.:	Tgl:

BAB I **Judul**

1. Dasar Teori

300 sampai dengan 350 kata

2. Tujuan

3. Analisa Hasil

3.1. Kegiatan 1: ...

Tampilkan hasil praktikum berupa kode yang dibuat atau hasil output, kemudian berikan analisisnya. Jika terdapat gambar, berikan juga nomor gambar.

3.2. Kegiatan 2: ...

4. Penyelesaian Tugas

Jika terdapat tugas yang dikerjakan, tuliskan disini langkah pengerjaan dan hasilnya.

5. Kesimpulan

Berikan kesimpulan yang didapatkan setelah anda menyelesaikan praktikum