

# Class dan Object

Tl2014 - Pemrograman Berorientasi Objek

Andik Yulianto

### Membuat Class



```
Attributes

Class <NamaKelas>:
    data1 = value1
    ...
    dataM = valueM

def <namafunction1>(self,arg1,...,argK):
    <block>

def <namafunctionN>(self,arg1,...,argK):
    <block>
```

CATATAN: Menurut konvensi penulisan nama class menggunakan huruf besar diawal kata, namun tidak boleh ada spasi. Contoh: jika ingin membuat nama "Kendaraan roda empat" maka ditulis "Kendaraan Roda Empat"

# Mendefinisikan fungsi dalam Class



- Sebuah Class dapat terdiri dari banyak fungsi (Function)
- Fungsi-fungsi tersebut merepresentasikan sebuah perilaku (behαvior) yang berkaitan dengan Class tersebut.
- Contoh:

```
class Mobil:
    def drive(self):
        print("berjalan")

    def brake(self):
        print("mengurangi kecepatan")
```

```
class Math:
    def subtract(self, a, b):
        return a-b

    def add(self, x, y):
    return x+y
```

Argumen "Self" merujuk ke objek itu sendiri

# Bagaimana membuat objek dan memanggilnya?



```
class Mobil:
    def __init__(self, name, model, color, speed):
        self.name= name
        self.model = model
        self.color = color
        self.speed = speed

    def drive(self):
        print("berjalan")

    def brake(self):
        print("mengurangi kecepatan")
```

1 Inisialisasi Class

```
pickup = Mobil("Toyota", "pickup", "red", '100km/jam')
truk = Mobil("Hino", "truck", "blue", "80km/jam")
```

2 Pembuatan objek

```
pickup.drive()
truck.brake()
```

Memanggil/eksekusi method sebuah objek

# Bagaimana membuat objek dan memanggilnya?



```
class Math:
    def subtract(self, a, b):
        return a-b

    def add(self, x, y):
        return x+y
```

```
1 Inisialisasi Class
```

```
pickup = Mobil("Toyota", "pickup", "red", '100km/jam')
truk = Mobil("Hino", "truck", "blue", "80km/jam")
```

2 Pembuatan objek

```
pickup.drive()
truck.brake()
```

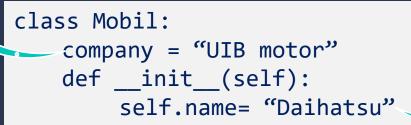
Memanggil/eksekusi method sebuah objek

# Attributes



#### Class Attribute

Didefinisikan di bawah nama class



```
>>sedan = Mobil()
>>truck = Mobil()
>>sedan.name = "Toyota"
>>print(truck.name)
Daihatsu
```

```
>>sedan = Mobil()
truck = Mobil()
>>sedan.company = "UK motor"
>>print(truck.company)
UK motor
```

#### Instance Attribute

Didefinisikan di dalam fungsi class.



Mengubah Instance Attribute hanya berpengaruh pada instance yang bersangkutan



Mengubah Class Attribute berpengaruh pada semua instance

### Constructor



- Saat sebuah instance dari Class dibuat, fungsi contructor secara otomatis dipanggil
- 2. Contructor selalu dituliskan dengan \_\_init\_\_()
- 3. Constructor terdiri dari kode program yang akan dieksekusi ketika sebuah objek dibuat (*creαte*)

```
class Mobil:
    def __init__(self, name, model, color, speed):
        self.name= name
        self.model = model
        self.color = color
        self.speed = speed
```

truk = Mobil("Hino", "truck", "blue","80km/jam")

## Methods



- Merupakan fungsi yang didefinisikan di dalam class dan hanya dapat dipanggil dari instance dari class tersebut.
- Karakteristik method:
  - dapat mengembalikan sebuah nilai (bisa juga tidak)
  - Dapat menerima paramereter

```
class Mobil:
    def __init__(self, name, color, speed):
        self.name= name
        self.color = color
        self.speed = speed

def drive(self):
        print("berjalan")

def warning(self):
        return "Peringatan kecepatan"
```

```
>>truck = Mobil('Toyota', 'blue', '100km/jam')
>>truck.drive()
berjalan

>>print(truck.warning())
Peringatan kecepatan
```

Instance mengembalikan sebuah nilai ketika dipanggil

### Kasus



- Sebuah departemen HR (human resources) PT. ABC ingin membuat perangkat lunak untuk pengelolaan pegawai mereka.
- 2. UIB membutuhkan perangkat lunak untuk mengelola inventory IT. Mulai dari memasukkan nama barang hingga

Buatlah paling tidak 2 class yang berhubungan dengan perangkat lunak tersebut diatas.