

Encapsulation

Program python berorientasi objek, Anda dapat membatasi akses ke metode dan variabel. Ini dapat mencegah data dari dimodifikasi secara tidak sengaja dan dikenal sebagai enkapsulasi.

Python tidak memiliki kata kunci pribadi, tidak seperti beberapa bahasa berorientasi objek lainnya, tetapi enkapsulasi dapat dilakukan. Sebagai gantinya, ia bergantung pada konvensi: variabel kelas yang tidak boleh diakses secara langsung harus diawali dengan garis bawah. (__)

```
In [3]: class Robot(object):
        def __init__(self):
            self.__version = 22

        def getVersion(self):
            print(self.__version)

        def setVersion(self, version):
            self.__version = version

obj = Robot()
obj.getVersion()
obj.setVersion(23)
obj.getVersion()
#print(obj._A_myAnotherPrivateMethod()) supaya bisa dapat diakses
print(obj._Robot__version)

22
23
23
```

Private Variable

Variabel dapat bersifat pribadi yang dapat berguna pada banyak kesempatan. Variabel pribadi hanya dapat diubah dalam metode kelas. Objek dapat menyimpan data penting untuk aplikasi Anda dan Anda tidak ingin data itu dapat diubah dari mana pun dalam kode.

```
In [4]: class Car:
        __maxspeed = 0
        __name = ""
        def __init__(self):
            self.__maxspeed = 200
            self.__name = "Supercar"
        def drive(self):
            print(
                'driving. maxspeed ' + str(self.__maxspeed))
redcar = Car()
redcar.drive()
redcar.__maxspeed = 10 # tidak berubah karena variable privat
redcar.drive()

driving. maxspeed 200
driving. maxspeed 200
```

Jika Anda ingin mengubah nilai variabel pribadi, gunakan metode set. Ini hanyalah sebuah metode yang menetapkan nilai variabel pribadi.

```
In [5]: class Car:
        __maxspeed = 0
        __name = ""
        def __init__(self):
            self.__maxspeed = 200
            self.__name = "Supercar"
        def drive(self):
            print(
                'driving. maxspeed ' + str(self.__maxspeed))
        def setMaxSpeed(self, speed):
            self.__maxspeed = speed
redcar = Car()
redcar.drive()
redcar.setMaxSpeed(320)
redcar.drive()

driving. maxspeed 200
driving. maxspeed 320
```

Berikut ringkasannya :

Type	Description
public methods	Accessible from anywhere
private methods	Accessible only in their own class. starts with two underscores
public variables	Accessible from anywhere
private variables	Accesible only in their own class or by a method if defined. starts with two underscores

Bahasa pemrograman lain juga memiliki metode kelas yang dilindungi, tetapi Python tidak. Enkapsulasi memberi Anda kontrol lebih besar terhadap dalam kode Anda, memungkinkan kelas untuk mengubah implementasinya tanpa mempengaruhi bagian lain dari kode.

```
In [ ]:
```