Pengertian Class dalam Pemrograman Berbasis Objek

Class adalah ‘*cetak biru*’ atau ‘*blueprint’* dari **object**. Class berfungsi untuk membuat kerangka dasar

**class** bisa diibaratkan dengan *komputer*  Kita tahu bahwa *komputer* memiliki ciri-ciri seperti *merk*, memiliki *keyboard*, memiliki *processor*, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah *komputer.* Selain memiliki ciri-ciri, sebuah laptop juga bisa dikenakan tindakan, seperti: *menghidupkan laptop* atau *mematikan laptop*.

*Class*  Di dalam pemrograman nantinya, contoh class seperti: *koneksi\_database* dan *profile\_user*.

Contoh:

class Person:  
  def \_\_init\_\_(self, name, age):  
    self.name = name  
    self.age = age  
  
p1 = Person("John", 36)  
  
print(p1.name)  
print(p1.age)

Pengertian atribut dalam Pemrograman Berbasis Objek

**Property** (atau disebut juga dengan *atribut*) adalah data yang terdapat dalam sebuah **class,**lanjut analogi tentang *komputer*, **atribut** dari komputer bisa berupa *merk, warna, jenis processor, ukuran layar*, dan lain-lain.

Jika anda sudah terbiasa dengan program Phyton, **property** ini sebenarnya hanyalah *variabel*yang terletak di dalam *class*. Seluruh aturan dan tipe data yang biasa diinput kedalam *variabel*, bisa juga diinput kedalam *property*. Aturan tata cara penamaan *property* sama dengan aturan penamaan *variabel*.

Contoh penulisan class dengan penambahan atribut :

class Person:  
  def \_\_init\_\_(self, name, age):  
    self.name = name  
    self.age = age  
  
p1 = Person("John", 36)  
  
print(p1.name)  
print(p1.age)

Dari contoh diatas, **person, nama**dan**umur** adalah *atribut* dari **class person**.

Pengertian Method dalam Pemrograman Berbasis Objek

**Method** adalah tindakan yang bisa dilakukan didalam *class*. Jika menggunakan analogi **class person**, maka contoh method adalah: ***menghidupkan komputer***, ***mematikan komputer***.

Method pada dasarnya adalah ***function*** yang berada di dalam **class**. Seluruh fungsi dan sifat function bisa diterapkan kedalam method, seperti argumen/parameter, mengembalikan nilai (dengan keyword *return*), dan lain-lain.

Contoh:

class Person:

def greet(name):

"""This function greets to

the person passed in as

parameter"""

print("Hello, " + name + ". Good morning!")

.

Pengertian Object dalam Pemrograman Berbasis Objek

**Object** atau **Objek** adalah *hasil cetak* dari **class**, atau hasil ‘*konkrit’* dari **class**. Jika menggunakan analogi **class komputer**, maka objek dari **class komputer** bisa berupa: *komputer\_andi, komputer\_anto, komputer\_duniailkom,* dan lain-lain. Objek dari **class komputer** akan memiliki seluruh ciri-ciri **komputer**, yaitu *property* dan *method*-nya.

Proses ‘*mencetak’* objek dari **class** ini disebut dengan ‘**instansiasi’** (atau ***instantiation*** dalam bahasa *inggris*).

Contoh:

class Person:  
  def \_\_init\_\_(self, name, age):  
    self.name = name  
    self.age = age  
  
p1 = Person("John", 36)  
  
print(p1.name)  
print(p1.age)

PEMBELAJARAN DI KELAS

