

Nama: Sedri Sella Jumeni

Nim: 211001073

Kelas : Pemrograman Python D

Object Oriented Programming

Object Oriented Programming (OOP) merupakan suatu konsep pemrograman yang menekankan pada paradigma atau cara pandang terhadap suatu masalah berdasarkan "object".

1. Istilah - Istilah Dalam OOP:

- Kelas adalah cetak biru atau prototipe dari object dimana kita mendefinisikan atribut dari suatu object
- Variabel Kelas adalah variabel yang dishare atau dibagi oleh semua instance (turunan) dari kelas..
- Data member adalah variabel yang menyimpan data yang berhubungan dengan kelas dan objeknya.
- Overloading Fungsi adalah fungsi yang memiliki nama yang sama di dalam kelas, tapi dengan jumlah dan tipe argumen yang berbeda sehingga dapat melakukan beberapa hal yang berbeda.
- Overloading Operator adalah pembuatan beberapa fungsi atau kegunaan untuk suatu operator.
- Variabel Instansiasi adalah variabel yang didefinisikan di dalam suatu metode dan hanya menjadi milik dari instance kelas.
- Pewarisan/Inheritance (Inheritance) adalah pewarisan karakteristik sebuah kelas ke kelas lain yang menjadi turunannya.
- Instance adalah istilah lain dari objek suatu kelas. Sebuah objek yang dibuat dari prototipe kelas Lingkaran misalnya disebut sebagai instance dari kelas tersebut.
- Instansiasi adalah pembuatan instance/objek dari suatu kelas.
- Metode adalah fungsi yang didefinisikan di dalam suatu kelas.
- Objek adalah instansiasi atau perwujudan dari sebuah kelas. Bila kelas adalah prototipenya, dan objek adalah barang jadinya.

2. Perkenalan Kelas & Objec

3. Kelas dan Object Sederhana

4. Variabel Kelas dan Object

5. Method

6. Game dengan OOP

TASK 4 MODUL 5

Contoh:

```
1 print("====MODUL/Tema 6====")
2 print("====SEDRi SELLA JUMENI====")
3 #Object Oriented Programming
4 #1. Perkenalan Kelas & Object
5 #kelas:adalah cetak biru atau prototipe dari object dimana kita mendefinisikan atribut dari suatu object
6 class python: #penamaan kelas, diikuti dengan titik dua (;)
7     pass #pass pada baris ke-3 artinya adalah pernyataan pass tidak melakukan apapun.
8 #object:adalah instansiasi atau perwujudan dari sebuah kelas. Bila kelas adalah prototipenya, dan objek adalah barang jadinya.
9 python1=python()#pembuatan object.
10 python2=python()#pembuatan object.
11 python3=python()#pembuatan object.
12
13 python1.name ="kelas D"
14 python1.mahasiswa="30"
15 python2.name ="kelas C"
16 python2.mahasiswa="31"
17 python3.name ="kelas B"
18 python3.mahasiswa="32"
19
20 #pemanggilan
21 print(python1)
22 print(python1.__doc__)
23 print(python1.name)#pemanggilan atribut.
24 #2.Kelas dan object sederhana
25 class marvel:
26     def __init__(self,inputname,inputhealth,inputpower,inputarmor):# __init__() adalah metode konstruktor
27         #untuk menginisialisasi pembuatan objek dari kelas tersebut.
28         self.name=inputname #self digunakan untuk mengakses atribut.
29         self.health=inputhealth
30         self.power=inputpower
31         self.armor=inputarmor
32
33 marvel1=marvel("iron man",100,10,90)
34 marvel2=marvel("thor",90,15,100)
35 marvel3=marvel("captain america",80,5,70)
36 print(marvel1.name)
37 print(marvel2.health)
38
39 #3.Variabel kelas dan object
40 class marvel: #class variabel
41     jumlah=0
42     def __init__(self,inputname,inputhealth,inputpower,inputarmor):#instance variabel:
43         self.name=inputname#variabel object
44         self.health=inputhealth
45         self.power=inputpower
46         self.armor=inputarmor
47         marvel.jumlah+=1
48         print("hero marvel dengan nama :"+inputname)
49 marvel1=marvel("iron man",1000,900,800)
50 print(marvel.jumlah)
51 marvel2=marvel("thor",900,1000,900)
52 print(marvel.jumlah)
53 marvel3=marvel("captain america",800,700,600)
54 print(marvel.jumlah)
55 #4.Method
56 class marvel:
57     def __init__(self,inputname,inputhealth,inputpower,inputarmor):# __init__() adalah metode konstruktor
58         #untuk menginisialisasi pembuatan objek dari kelas tersebut.
59         self.name=inputname #self digunakan untuk mengakses atribut.
60         self.health=inputhealth
61         self.power=inputpower
62         self.armor=inputarmor
63 #void function method tanpa return
64     def siapa(self):
65         print("namaku adalah :"+self.name)
66         #method dengan argumen
67     def healthtambah(seft,tambah):
68         self.health += tambah
69
```

TASK 4 MODUL 5

```
70     #method dengan return
71     def gethealth(self):
72         return self.health
73
74     marvel1=marvel('iron man',1000,900,800)
75     marvel2=marvel("thor",900,1000,900)
76     marvel3=marvel("iron man",800,700,600)
77
78     #pemanggilan method
79     marvel1.siapa()
80     #pemakaian method dengan argumen
81     marvel1.healthtambah(10)
82     print(marvel1.health)
83     #mengembalikan nilai dengan method
84     print(marvel.gethealth())
85     #5.Game dengan OOP
86     class marvel:
87         def __init__(self,name,health,attackpower,armornumber):
88             self.name=name
89             self.health=health
90             self.attackpower=attackpower
91             self.armornumber=armornumber
92         def serang(self,lawan):
93             print(self.name+"menyerang"+lawan.name )
94             lawan.diserang(self,self.attackpower)
95         def diserang (self,lawan,attackpower lawan):
96             print (self.name +"diserang"+lawan.name)
97             attact_diterima=attackpower lawan
98             print("serangan terasa:"+str(attact_diterima))
99             self.health-=attact_diterima
100             print("Darah"+self.name+"tersisa"+str(self.health))
101     ironman=marvel("iron man",100,10,5)
102     thor=marvel("thor",95,15,10)
103     #ironman.serang()
104     ironman.serang(thor)
105
106
```