KOMINFO digitalent

PRO Professional Academy

Digital Talent Scholarship 2022

Python - 2

Lead a sprint through the Machine Learning with Tensorflow Track

Agenda

- Introduction Class and Object, OOP, Methods
- Writing the script

KOMINFO digitalent

Are your students ML-ready?

Introduction to Class and Object

Apa itu Class ?
Class merupakan hal yang paling dasar dalam Python.

Class bersinggungan langsung dengan OOP

KOMINFO digitalent

Apa itu OOP ?

Dikarenakan python merupakan Bahasa pemrograman multi-paradigma. OOP mendukung pendekatan pemrograman yang berbeda.

Pendekatan popular yang digunakan untuk memecahkan masalah adalah dengan membuat objek. Yang dikenal sebagai OOP (Pemrograman berorientasi objek)

Objek memiliki 2 karakteristik :

1. Atribut
2. Pelaku

Contoh :

Beo adalah sebuah objek dengan beberapa property dibawah ini :

1. Atribut -> nama, umur, warna
2. Pelaku -> menyanyi, berbicara

OOP juga berfokus pada pembuatan kode yang bisa dipakai kembali, konsep OOP adalah DRY(Don’t repeat Yourself)

Konsep OOP adalah :

1. Class
2. Objek
3. Metode
4. Inheritance (turunan)
5. Enkapsulasi
6. polymorphisme

Class atau Kelas

Class is a blueprint for the object.

Contoh kita membuat class dari beo dimana terdiri dari nama, warna, ukuran, dll.

Contoh

```
class beo :
    pass
```

Object

Objek/ instance adalah sebuah instantiasi dari sebuah kelas.

Ketika kelas didefinisikan, hanya deskripsi untuk objek yang didefinisikan. Sehingga tidak ada memori atau penyimpanan yang dialokasikan.

Contoh :

obj = beo() #obj merupakan objek dari kelas beo

Method

Metode merupakan fungsi yang didefinisikan didalam badan kelas.

Kita lihat implementasi codenya setelah slide ini!

```
class beo :
        #instance atribut
        def __init__(self.nama, umur):
                self.nama = nama
                self.umur = umur
        #instance metod
        def nyanyi(self, song):
                return”{} menyanyi lagu {}”.format(self.name,song)

blu = beo(“bulu”,10)
print(blu.nyanyi(“”Happy””))
```

Turunan (Inheritance)

Turunan adalah cara membuat kelas baru untuk menggunakan detail kelas yang ada tanpa memodifikasinya.

Contohnya :

```
class buah :
    def __init__(self):
        print(“Ini buah”)
    def apaIni(self):
        print(“buah”)
```

```
#turunan
class apel(buah):
        def __init__(self):
                super().__init__()
                print(“apel adalah buah”)
        def apaIni(self):
                print(“apel”)
        def warna(self):
                print(“warna merah”)

A = apel()
A.apaIni()
A.warna()
A.__init__()
```

Mulai Bingung ?

Enkapsulasi

Membatasi akses ke metode dan variable. Berfungsi untuk mencegah data dari modifikasi langsung agar data tetap secure. Atribut pribadi menggunakan garis bawah sebagai awalan yaitu _ tunggal atau ganda __.

Contohnya :

```
class komputer :
        def__init__(self):
                self.__maxprice = 900
        def sell(self,price):
                self.__maxprice = price
c =komputer()
c.sell(1000)
```

Polimorfisme

Polimorfisme merupakan kemampuan untuk menggunakan antarmuka umum untuk berbagai bentuk.

Contohnya : kita perlu mewarnai beberapa bentuk seperti (bulat, persegi, persegi Panjang). Namun kita bisa menggunakan metode warna untuk semua bentuk yang ada hal ini disebut sebagai polimorfisme.

Contoh Polimorfisme

```
class beo :
        def terbang(self):
                print(“beo terbang”)
        def warna (self):
                print(“beo merah”)
class merpati :
        def terbang(self):
                print(“merpati terbang”)
        def warna (self):
                print(“merpati putih”)
def terbang(burung):
        burung.terbang()
```

```
Beo=beo()
Merpati = merpati()

terbang(Beo)
terbang(Merpati)
```

KOMINFO digitalent

Yang Perlu diingat !

- OOP membuat program lebih mudah dipahami serta efisien
- Code dapat digunakan Kembali
- Data aman dan terjamin dengan abstraksi data
- Polimorfisme memungkinkan antarmuka yang sama untuk objek yang berbeda sehingga bisa menulis kode dengan efisien.

Overloading

Overloading merupakan operator bawaan yang sama menunjukkan perilaku yang berbeda untuk objek dari kelas yang berbeda.

Contohnya :

```
class Point():
  def __init__(self, x= 0,y=0):
    self.x = x
    self.y = y
    hasil  = x+y
    print(hasil)



p1 = Point(1,2)
p2 = Point(2, 3)
```

Overriding

Overriding merupakan kemampuan pemogramman subclass atau kelas anak untuk menyediakan implementasi spesifik dari metode yang sudah ada oleh salah satu kelas induk.

KOMINFO digitalent

Sudah Mulai Pusing ?
Quick Recap
Apa itu Inheritance ?

Iterator

Kita menggunakan syntax next() untuk berpindah ke item lain yang ada pada list.

Contohnya :

```
listku = [1,2,3,4,5]
myIter = iter(listku)
next(myIter)
next(myIter)
```

Apakah Iterator bisa kita jalankan melalui Looping ?

Closure

x	y

def print_msg(msg):
  # This is the outer enclosing function
 	
  def printer():
  # This is the nested function
  print(msg)
 	
  printer()
 	
# We execute the function 	
# Output: Hello 	
print_msg("Hello")
	def print_msg(msg):
	  # This is the outer enclosing function
	
	  def printer():
	  # This is the nested function
	  print(msg)
	
	  return printer # returns the nested function
	
	# Now let's try calling this function.
	# Output: Hello
	another = print_msg("Hello")
	another()

Manakah table diatas yang merupakan Closure ? Dan sebutkan alasannya!

Kriteria yang harus dipenuhi untuk membuat Closure

- Harus memiliki fungsi perulangan di dalam perulangan (nested loop)
- Fungsi nested harus merujuk pada nilai yang ditentukan dalam fungsi
- Fungsi harus mengembalikan ke fungsi nested

Getter and Setter

Fungsi getter berasal dari kata get yang berarti mendapatkan, mendapatkan dalam python misalnya, mendapatkan nilai.

Fungsi setter berasal dari kata set yang berarti menetapkan dalam python misalnya, menetapkan nilai untuk sebuah fungsi.

Ayuk kita lihat contohnya!

Contoh:

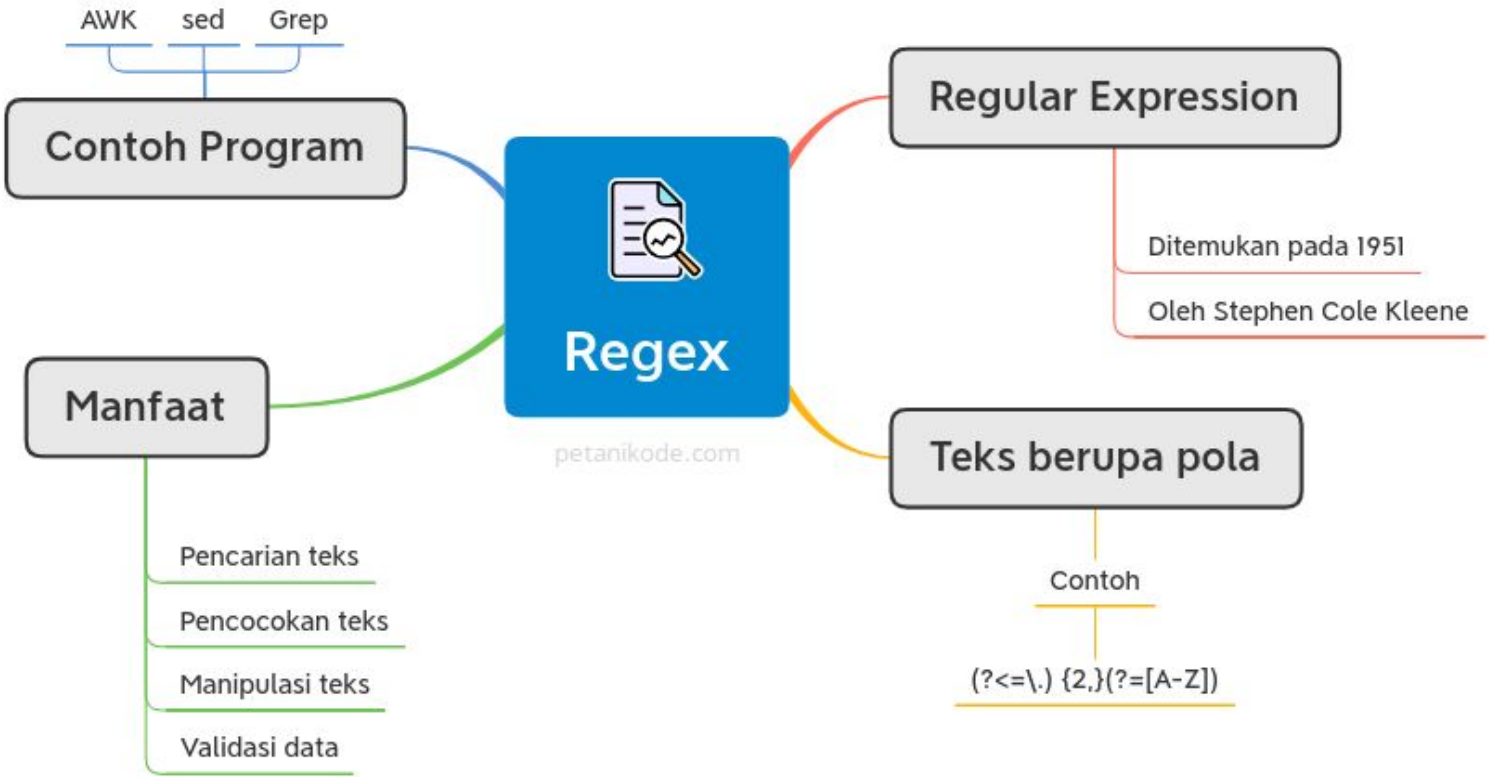
```
class lemari :
  def __init__(self, barang="baju"):
   self.set_barang(barang)
  def get_barang(self):
   return self._barang
  def set_barang(self, barang):
   self._barang = barang
   print(barang)

d = lemari()
d.get_barang()
d.set_barang('pita')
```

KOMINFO digitalent

RegEx

Expresi	String	Match ?
^a…s$	abs	No Match
	alias	Match
	abyss	Match
	Alias	No Match
	An abacus	No Match

Jadi RegEx atau bisa disebut sebagai Regular Expression merupakan sebuah sekuens dari karakter yang mendefinisikan Pattern dari pencarian.


AWK
sed
Grep
Contoh Program
Regex
petanikode.com
Regular Expression
Ditemukan pada 1951
Oleh Stephen Cole Kleene
Manfaat
Pencarian teks
Pencocokan teks
Manipulasi teks
Validasi data
Teks berupa pola
Contoh
(?<=\.) {2,}(?=[A-Z])

Karakter apa saja sih yang bisa dipakai untuk pencariannya ?

Ada : []\.^$*+?{}()\|

Kita memakai [] untuk menspesifikasi karakter yang ingin dicari

Contohnya :

[abc]

Test case:

a # cocok 1

ac #cocok 2

hkimh #tidak ada yang cocok

ShortCut RegEx

[a-z] -> Berarti dari a sampai z

[1-7] -> berarti angka 1 sampai 7

[^bvc] -> berarti karakter lainnya selain b,v,c

Silahkan mencari shortcut lainnya dan jelaskan !

FindAll

Digunakan untuk mendapatkan list yang terdiri dari semua yang kita cari.

Contohnya :

```
import re

string = ‘haii 15. Nomor kamu 34'
pattern = '\d+'

result = re.findall(pattern, string)
print(result)
```

Hasilnya :

[15,34]

Split()

Ini digunakan untuk membagi string menjadi beberapa bagian.

Contohnya :

```
import re

string = ‘haii 15. Nomor kamu 34'
pattern = '\d+'

result = re.split(pattern, string)
print(result)
```

Hasilnya :

[‘haii’,’’. Nomor kamu ‘]

KOMINFO digitalent

Saatnya mencari RegEx lainnya selain yang sudah kita pelajari apakah ada yang lain ? Jika iya, sebutkan 2 dan jelaskan !

Fungsi Lambda

Kekuatan lambda lebih baik ditampilkan ketika kita menggunakannya sebagai fungsi anonim di dalam fungsi lain.

Contohnya :

```
def coba(n):
  return lambda a : a * n
test1= coba(2)
print(test1(12))
```

Hasilnya :

24

Scripting

Buatlah pemrograman Kalkulator sederhana
Terapkanlah apa yang sudah dipelajari sebelumnya dan presentasikan
Yang wajib ada pada kalkulator :
1. Fungsi penambahan
2. Fungsi Pengurangan
3. Fungsi Perkalian
4. Fungsi Pembagian

Semangat! Ditunggu (30 Menit)

KOMINFO digitalent

Terima kasih