

Python (Chapter 2)



Tri Angga Dio Simamora

NIM : 1.18.4.047

Politeknik Pos Indonesia

Applied Bachelor Program of Informatics Engineering

Bandung 2019

‘Jika Kamu tidak dapat menahan lelahnya belajar,
Maka kamu harus sanggup menahan perihnya Kebodohan.’
Imam Syafi’i

Contents

1	Teori	1
1.1	Pemahaman Teori	1
1.1.1	Fungsi	1
1.1.2	Paket	1
1.1.3	Class, Object, Atribute, and Method	2
1.1.4	Pemanggilan Class	2
1.1.5	Penggunaan Fungsi	2
1.1.6	Pemanggilan Fungsi dari Folder	3
1.1.7	Pemanggilan Class dari Folder	3
2	Keterampilan Pemrograman	4
2.1	Listing Program Code	4

List of Figures

Listings

src/kode_tugas.py	4
-----------------------------	---

Chapter 1

Teori

1.1 Pemahaman Teori

1.1.1 Fungsi

Fungsi adalah sebuah block method jika dijalankan maka fungsi itu akan mengembalikan nilai. Fungsi memiliki parameter yang didalamnya adalah sebuah isian variabel yang akan diisi. Fungsi dapat dipanggil berulang-ulang sesuai yang kita inginkan. contoh kode :

```
def nambahinAngka(angka1, angka2):  
    hasil = angka1 + angka2  
    return hasil
```

Jika kita input $angka1 = 1$ dan $angka2 = 2$ maka akan mengembalikan nilai sebesar 3. Bisa juga sebuah fungsi untuk mengolah string contoh :

```
def rangkaiKata(nama, umur):  
    hasil = "halo nama saya " + nama + ", dan umur saya " + umur  
    return hasil
```

Jika kita input $nama = "angga"$ dan $umur = "20"$, maka hasil dari return akan menjadi seperti "halo nama saya angga, dan umur saya 20".

1.1.2 Paket

Paket adalah sebuah bungkus yang membungkus semua isi yang ada didalamnya. untuk cara pemanggilan paket atau library cukup mudah.

```
import selenium
```

Ketika kita mengetikkan perintah diatas maka paket telah kita panggil dan siap digunakan.

1.1.3 Class, Object, Attribute, and Method

Class atau Kelas adalah sebuah kerangka dari objek yang akan kita inisiasikan nantinya. Objek adalah sebuah wujud yang dapat kita lakukan perintah sesuai dengan method-nya. Atribut ada 2 yaitu kelas atribut dan instansi atribut, perbedaannya hanya di letak, kalau kelas atribut ada di bawah kelas, dan instansi atribut ada didalam fungsi, atribut itu sebuah variabel yang dimiliki oleh parentnya seperti fungsi atau class.

```
class Auahelap(object):  
    iniNamanyaKelasAtribut = "paksayaganteng"  
  
    def sayaGanteng(self, kamujelek):  
        self.iniNamanyaIntansiAtribut = kamujelek
```

1.1.4 Pemanggilan Class

Cara pemanggilan class dari instansiasi pemanggilan sebuah paket caranya gini,

```
from filegelap import Auahelap
```

from itu tuh manggil si filenya, abis itu *import* itu kita panggil si classnya.

1.1.5 Penggunaan Fungsi

```
from kalkulator import Penambahan
```

```
Penambahan(1,2)
```

Jika perintah diatas dilakukan maka fungsi Penambahan akan dijalankan dimana fungsi Penambahan yaitu menambahkan parameter ke-1 dan parameter ke-2. contoh kode lain :

```
from sayaganteng import gantengBanget
```

```
ganteng("angga")
```

1.1.6 Pemanggilan Fungsi dari Folder

```
from folderKu import kalkulator
```

```
a,b = 10, 15
```

```
run = kalkulator.Ngitung(a,b)
```

```
run.Penambahan()
```

Sebuah fungsi Penambahan ada didalam file kalkulator dan file kalkulator itu ada di folder yang lain jadi begitulah cara pemanggilannya.

1.1.7 Pemanggilan Class dari Folder

```
from folderKu import kalkulator
```

```
run = kalkulator.Ngitung()
```

Sebuah kelas Perhitungan ada didalam file kalkulator dan file kalkulator itu ada di folder yang lain jadi begitulah cara pemanggilannya.

Chapter 2

Keterampilan Pemrograman

2.1 Listing Program Code

Fungsi init sebagai constructor dari variabel npm

```
def __init__(self, npm):
    self.npm = npm

def nomor1(self):
    print("Soal_nomor_1")
    nPm = int(self.npm)
    print(nPm % 3)

    print("###_###_#####_###_###_#####_###_###_")
    print("###_###_#####_###_###_#####_###_###_")
    print("###_###_###_###_###_###_###_###_###_###_")
    print("###_###_###_###_###_###_###_###_###_###_")
    print("###_###_#####_#####_###_###_#####_")
    print("###_###_###_###_###_###_###_###_###_###_")
    print("###_###_###_###_###_###_###_###_###_###_")
    print("###_###_#####_###_#####_###_###_")
    print("###_###_#####_###_#####_###_###_")

def nomor2(self):
    # nomor 2
    print("soal_nomor_2")

    nPm = int(self.npm[-2:])

    for i in range(nPm):
        print("Halo, " + self.npm + " _apa_kabar?")

    # nomor 3
    print("soal_nomor_3")
```

```

nPm = self.npm[-3:]

b = 0

for i in nPm:
    c = int(i) + b
    b = c

for i in range(b):
    print("Halo , " + nPm + " _apa_kabar?")

def nomor4(self):
    # nomor 4
    print("soal_nomor_4")

nPm = self.npm[-3]

for i in nPm:
    print("Halo , " + nPm + " _apa_kabar?")

def nomor5(self):
    # nomor 5
    print("soal_nomor_5")

string = "abcdefg"

index = 0

for i in string:
    print(i + " _=" + self.npm[index])
    index += 1

def nomor6(self):
    # nomor 6
    print("soal_nomor_6")

a = 0
b = 0
for i in self.npm:
    c = int(i) + b
    b = c
    a += 1

print(b)

```

```

def nomor7(self):
    # nomor 7
    print("soal_nomor_7")

    a = 0
    b = 0
    for i in self.npm:
        c = int(i) * b
        b = c
        a += 1

    print(b)

def nomor8(self):
    # nomor 8
    print("soal_nomor_8")

    for i in self.npm:
        print(i)

def nomor9(self):
    # nomor 9
    print("soal_nomor_9")

    for i in self.npm:
        a = int(i)

        if a != 0:
            if a % 2 == 0:
                print(i, end=" ")

def nomor10(self):
    # nomor 10
    print("soal_nomor_10")

    for i in self.npm:
        a = int(i)

        if a != 0:
            if a % 2 == 1:
                print(i, end=" ")

def nomor11(self):
    # nomor 11
    print("soal_nomor_11")
    for i in self.npm:

```

```
number = int(i)

if number > 1:

    for i in range(2, number):
        if (number % i) == 0:
            break
    else:
        print(number)
```