

Tugas Chapter 3

Rayhan Yuda Lesmana

27 Oktober 2019

Teori

Fungsi

Fungsi dalam python merupakan satu blok program yang terdiri dari nama fungsi, input variabel, dan variabel kembalian. Nama fungsi di python diawali dengan def dan setelahnya tanda titik dua. Fungsi input() dan raw_input digunakan untuk mengambil data angka dan teks. Cara mengembalikan nilai dari sebuah fungsi dengan menggunakan kata kunci return lalu diikuti dengan nilai atau variabel yang akan dikembalikan.

Inputan fungsi(parameter) adalah inputan sebuah fungsi bertujuan menyimpan sebuah nilai.

kembalian fungsi(return) berfungsi untuk mengembalikan sebuah nilai, dan bisa juga mengakhiri sebuah eksekusi fungsi.

Contoh syntax:

```
def function(a,b):  
    c=a*b  
    return c
```

Paket dan Library

Package adalah folder yang menyimpan syntax code. Semisal kita membuat folder yang di dalamnya terdapat syntax program kita dengan nama remote.py.

Cara memanggil package:

```
from remote import batre
```

Kelas, Objek, Atribut dan methode

Class Class adalah cetak biru (blueprint) dari sebuah onject.

Contoh syntax:

```
class Name: def _init_(self,name): self.name = name def hayname(self):  
print("Haii", name)
```

Obejct Obejct adalah hasil cetak dari sebuah class.

Contoh syntax:

```
import kelas3lib
```

```
cobakelas=kelas3lib.aku(npm)  
hasil=cobakelas.NPM2()
```

Atribut Atribut adalah variabel yang dimiliki oleh sebuah class.

Contoh syntax:

```
class Name:  
def _init_(self,nama):  
self.nama = nama
```

Method Method fungsi dari sebuah class.

Contoh syntax:

```
class Name:  
def _init_(self,name):  
self.name = name  
def name(self):  
print("hayy",name)
```

Library

Membuat folder library dengan nama try:

```
def Name():  
print("Rayhan yuda")
```

Contoh memanggil fungsi dari library:

```
import try  
try.Name()
```

Penggunaan package from kalkulator import perkalian

Contoh lainnya:

from motor import bensin

Pemanggilan library dalam folder Contoh syntax:

from motor import bensin

Memakai library bensin.

Pemanggilan class dalam folder Contoh syntax:

from motor import bensin

Memakai class bensin.

Keterampilan pemograman

1. Question1

```
def NPM1():  
    print("****  
    print("****  
    print("****  
    print("****  
    print("****
```

2. Question2

```
def NPM2():  
    NPM=int(input("masukan NPM:"))  
    Tld=NPM%100  
    for i in range(Tld):  
        print("Halo ", NPM, " apa kabar ?")
```

3. Question3

```
def NPM3():  
    NPM=input("Masukan Npm kamu: ")  
  
    X=int(NPM[4])  
    Y=int(NPM[5])  
    Z=int(NPM[6])  
  
    hitung1 = X + Y + Z  
    hitung2 = X + Y + Z  
  
    while hitung1 > 0:  
        print("Halo ,", NPM[4:7], " Apa kabar ?")  
        hitung1 = hitung1 -1  
    print(" ... ",str(hitung2), " kali (",str(X),"+",str(Y),"+",str
```

4. Question4

```
def NPM4():  
    NPM = input("Npm_kamu: ")  
  
    print(" Halo , " ,NPM[4] , " _Apa_kabar?" )
```

5. Question5

```
def NPM5():  
  
    i=0  
    NPM = input("Npm: ")  
    while i<1:  
        if len(NPM)<7:  
            print("Npm_kurang_dari_7!")  
            NPM = input("Npm: ")  
        elif len(NPM)>7:  
            print("Npm_lebih_dari_7!")  
            NPM = input("Npm: ")  
        else:  
            i=1  
  
    A=NPM[0]  
    B=NPM[1]  
    C=NPM[2]  
    D=NPM[3]  
    E=NPM[4]  
    F=NPM[5]  
    G=NPM[6]  
  
    for this in A,B,C,D,E,F,G:  
        print(this)
```

6. Question6

```
def NPM6():  
    i=0  
    NPM = input("Npm: ")  
    while i<1:  
        if len(NPM)<7:  
            print("Npm_kurang_dari_7!")
```

```

        NPM = input("Npm: ")
    elif len(NPM)>7:
        print("Npm lebih dari 7!")
        NPM = input("Npm: ")
    else:
        i=1

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]

X=0

for this in A,B,C,D,E,F,G:
    X+=int(this)
print(X)

```

7. Question7

```

def NPM7():
    i=0
    NPM = input("Npm: ")
    while i<1:
        if len(NPM)<7:
            print("Npm kurang dari 7!")
            NPM = input("Npm: ")
        elif len(NPM)>7:
            print("Npm lebih dari 7!")
            NPM = input("Npm: ")
        else:
            i=1

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]

```

```
F=NPM[5]
G=NPM[6]
```

```
X=1
```

```
for this in A,B,C,D,E,F,G:
    X*=int(this)
print(X)
```

8. Question8

```
def NPM8():
    i=0
    NPM = input("Npm: ")
    while i<1:
        if len(NPM)<7:
            print("Npm kurang dari 7!")
            NPM = input("Npm: ")
        elif len(NPM)>7:
            print("Npm lebih dari 7!")
            NPM = input("Npm: ")
        else:
            i=1
```

```
A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]
```

```
X=1
```

```
for this in A,B,C,D,E,F,G:

    if int(this)%2==0:
        if int(this)==0:
            this=""
        print(this,end=" ")
```

9. Question9


```

def NPM9():
    i=0
    NPM = input("Npm: ")
    while i < 1:
        if len(NPM) < 7:
            print("Npm kurang dari 7!")
            NPM = input("Npm: ")
        elif len(NPM) > 7:
            print("Npm lebih dari 7!")
            NPM = input("Npm: ")
        else:
            i=1

```

```

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]

```

```

X=1

```

```

for this in A,B,C,D,E,F,G:

```

```

    if int(this)%2==1:
        print(this,end=" ")

```

10. Question10

```

def NPM10():
    i=0
    NPM = input("Npm: ")
    while i < 1:
        if len(NPM) < 7:
            print("Npm kurang dari 7!")
            NPM = input("Npm: ")
        elif len(NPM) > 7:
            print("Npm lebih dari 7!")
            NPM = input("Npm: ")
        else:

```

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]

11. Question11

9

```

NPM = npm
while i < 1:
    if len(NPM) < 7:
        print("Npm kurang dari 7!")
        NPM = input("Npm: ")
    elif len(NPM) > 7:
        print("Npm lebih dari 7!")
        NPM = input("Npm: ")
    else:
        i = 1
A = NPM[0]
B = NPM[1]
C = NPM[2]
D = NPM[3]
E = NPM[4]
F = NPM[5]
G = NPM[6]
for this in A,B,C,D,E,F,G:
    print(this)
def NPM6(npm):
    i = 0
    NPM = npm
    while i < 1:
        if len(NPM) < 7:
            print("Npm kurang dari 7!")
            NPM = input("Npm: ")
        elif len(NPM) > 7:
            print("Npm lebih dari 7!")
            NPM = input("Npm: ")
        else:
            i = 1
    A = NPM[0]
    B = NPM[1]
    C = NPM[2]
    D = NPM[3]
    E = NPM[4]
    F = NPM[5]
    G = NPM[6]
    X = 0
    for this in A,B,C,D,E,F,G:
        X += int(this)

```

```

    print(X)
def NPM7(npm):
    i=0
    NPM = npm
    while i<1:
        if len(NPM)<7:
            print("Npm_kurang_dari_7!")
            NPM = input("Npm:_")
        elif len(NPM)>7:
            print("Npm_lebih_dari_7!")
            NPM = input("Npm:_")
        else:
            i=1
    A=NPM[0]
    B=NPM[1]
    C=NPM[2]
    D=NPM[3]
    E=NPM[4]
    F=NPM[5]
    G=NPM[6]
    X=1
    for this in A,B,C,D,E,F,G:
        X*=int(this)
    print(X)
def NPM8(npm):
    i=0
    NPM = npm
    while i<1:
        if len(NPM)<7:
            print("Npm_kurang_dari_7!")
            NPM = input("Npm:_")
        elif len(NPM)>7:
            print("Npm_lebih_dari_7!")
            NPM = input("Npm:_")
        else:
            i=1
    A=NPM[0]
    B=NPM[1]
    C=NPM[2]
    D=NPM[3]
    E=NPM[4]

```

```

F=NPM[5]
G=NPM[6]
X=1
for this in A,B,C,D,E,F,G:
    if int(this)%2==0:
        if int(this)==0:
            this=""
        print(this,end=" ")
def NPM9(npm):
    i=0
    NPM = npm
    while i<1:
        if len(NPM)<7:
            print("Npm_kurang_dari_7!")
            NPM = input("Npm_: ")
        elif len(NPM)>7:
            print("Npm_lebih_dari_7!")
            NPM = input("Npm_: ")
        else:
            i=1
A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]
X=1
for this in A,B,C,D,E,F,G:
    if int(this)%2==1:
        print(this,end=" ")
def NPM10(npm):
    i=0
    NPM = npm
    while i<1:
        if len(NPM)<7:
            print("Npm_kurang_dari_7!")
            NPM = input("Npm_: ")
        elif len(NPM)>7:
            print("Npm_lebih_dari_7!")
            NPM = input("Npm_: ")

```

```

        else :
            i=1
A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]
X=1
for x in A,B,C,D,E,F,G:
    if int(X) > 1:
        for i in range(2,int(X)):
            if (int(X) % i) == 0:
                break
        else :
            print (int(X),end ="" ),

```

12. Question12

```

import lib3
class aku:
    def __init__(self ,npm):
        self.npm = npm
    def NPM1(self):
        return lib3.NPM1()
    def NPM2(self):
        return lib3.NPM2(self.npm)
    def NPM3(self):
        return lib3.NPM3(self.npm)
    def NPM4(self):
        return lib3.NPM4(self.npm)
    def NPM5(self):
        return lib3.NPM5(self.npm)
    def NPM6(self):
        return lib3.NPM6(self.npm)
    def NPM7(self):
        return lib3.NPM7(self.npm)
    def NPM8(self):
        return lib3.NPM8(self.npm)
    def NPM9(self):

```

```
        return lib3.NPM9(self.npm)
def NPM10(self):
    return lib3.NPM10(self.npm)
```

Keterampilan penanganan error

Penanganan error error:

Tipe error: `_ _ init_ _` missing 1 required positional argument: "npm"

Penyelesaian:

Menambahkan para meter.

Syntax:

```
def perkalian(a,b): c=a*b return c
```

```
    d=int(input("Angka")) e=int(input("Angka")) try: print(perkalian(d,e))
```

```
except: print("Tidak boleh 0")
```