

Tugas Praktikum Pemrograman II

(Chapter 3)



Disusun oleh :

Dimas Aqila Maulana

D4 TI 2C

1.18.4.081

PROGRAM DIPLOMA IV POLITEKNIK POS INDONESIA

POLITEKNIK POS INDONESIA

BANDUNG

2019

1 Fungsi

1.1 pemahaman teori

1. fungsi

Fungsi adalah blok kode yang akan dieksekusi ketika dipanggil dalam sebuah program.

2. Parameter

Parameter adalah inputan sebuah fungsi yang bertujuan untuk menyimpan suatu nilai.

3. Return

Return berfungsi untuk mengembalikan suatu nilai yang telah di proses dalam suatu fungsi dan mengakhiri sebuah fungsi.

```
def fungsi (x , y):  
    z=x+y  
    return z
```

4. Item Paket

Item Paket adalah sebuah direktori dengan suatu file python dan file dengan nama `_init_.py`. jadi sebuah direktori yang didalamnya ada sebuah python dengan nama `_init_.py`, dan dianggap sebagai paket oleh python tersebut, untuk memanggil sebuah paket atau library adalah dengan cara menekan *import* nama paket atau library tersebut lalu paket atau library tersebut dapat digunakan.

```
from kampus import mahasiswa
```

5. Class

Class adalah sebuah blueprint dari suatu objek yang akan di bangun.

```
class World:  
    def __init__(self , World):
```

```

        self.World = World
    def helloWorld( self ):
        print ( "Helo" , World )

```

6. Objek memiliki suatu variabel dan kode yang saling terhubung. objek di buat dengan adanya suatu class.

```

#import kelas terlebih dahulu
import kelas3lib
#membuat object
cobakelas=kelas3lib . Kelas3ngitung (npm)
hasilkelas=cobakelas . npm1 ()

```

7. Attribut

Attribut adalah sebuah tempat tampungan dari sebuah data atau perintah yang berhubungan dengan attribut tersebut.

```

Class Kelas3ngitung :
    #pendefinisian attribute
    def __init__( self , World ):
        self.World = World

```

8. Method

Method adalah sebuah fungsi yang ada didalam suatu class.

```

class world :
    def __init__( self , world ):
        self.world = world
    #Pembuatan method pada class
    def world( self ):
        print ( "hello" , world , " , apa_kabar_?" )

```

9. Contoh membuat sebuah library, contoh disini kita membuat pada folder library:

```
def hello():  
    print("Hello world")
```

10. Contoh jika kita ingin memanggil sebuah fungsi dari suatu library pada main program kita harus terlebih dahulu melakukan import:

```
#import library yang telah dibuat  
import library  
#pemanggilan fungsi pada library  
library.hello()
```

11. Pemakaian package from kalkulator import penambahan

```
from kalkulator import penambahan
```

kode diatas merupakan perintah program yang memanggil sebuah package terlebih dahulu baru menambahkan source code penambahan, kode diatas dapat dibaca seperti ini "import penambahan dari folder kalkulator"

12. Pemanggilan library dalam sebuah folder

untuk mengakses sebuah library dalam sebuah folder kita perlu menuliskan foldernya terlebih dahulu, setelah itu kita mengimport nama librarynya, contoh:

```
from mahasiswa import kampus
```

artinya dalam package mahasiswa akan memakai atau menggunakan suatu library kampus

13. Pemanggilan class dalam sebuah folder

untuk mengakses sebuah class dalam sebuah folder kita perlu menuliskan foldernya terlebih dahulu lalu kita mengimport nama class nya, contoh :

```
from mahasiswa import kampus
```

artinya dalam package mahasiswa akan memakai atau menggunakan suatu class kampus

1.2 Ketrampilan Pemrograman

```
1. def npm1():
```

```
print(" _+++_+++_+++_+++_++++_+++_+++_++++_++++_+++_+++")
```

```
print(" _+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_")
```

[illegible][illegible]

```
print(" _+++_+++_+_+++_+++++_____+++_++_+++++_+_+_+++_++++_+_+_++_")
```

```
2. # -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Tue Nov 12 20:17:23 2019
```

```
@author: dimas
```

```
"""
```

```
def npm2(npm):
```

```
    npm=int(npm)
```

```
    TwoLastDigit=abs(npm)%100
```

```
    for i in range(TwoLastDigit):
```

```
        print("Hallo , ", npm, " apa kabar ?")
```

```
def npm3(npm):
```

```
    for i in range(int(str(npm)[4])+int(str(npm)[5])+int(str(npm)[6])):
```

```
        print("Hallo , "+str(npm)[4]+str(npm)[5]+str(npm)[6]+" a
```

```
i=0
```

```
npm=input("Masukan NPM : ")
```

```
while i<1:
```

```
    if len(npm) < 7:
```

```
        print("NPM Kurang dari 7 digit")
```

```
        npm=input("Masukan NPM : ")
```

```
    elif len(npm) > 7:
```

```
        print("NPM lebih dari 7 digit")
```

```
        npm=input("Masukan NPM : ")
```

```
    else:
```

```
        i=1
```

```
npm3(npm)
```

```

3. def npm4(npm):
    key=npm%1000
    str_key=str(key)
    print("Hallo , "+str_key[0]+"_apa_kabar_?")

i=0
npm=input("Masukan _NPM_: ")
while i < 1:
    if len(npm) < 7:
        print("NPM_Kurang_dari_7_digit")
        npm=input("Masukan _NPM_: ")
    elif len(npm) > 7:
        print("NPM_lebih_dari_7_digit")
        npm=input("Masukan _NPM_: ")
    else:
        i=1
npm4(npm)

def npm5(npm):
    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]

    for x in a,b,c,d,e,f,g:
        print(x)

```

```

i=0
npm=input("Masukan NPM : ")
while i < 1:
    if len(npm) < 7:
        print("NPM Kurang dari 7 digit")
        npm=input("Masukan NPM : ")
    elif len(npm) > 7:
        print("NPM lebih dari 7 digit")
        npm=input("Masukan NPM : ")
    else :
        i=1
npm5(npm)

```

6. **def** npm6(npm):

```

a=npm[0]
b=npm[1]
c=npm[2]
d=npm[3]
e=npm[4]
f=npm[5]
g=npm[6]
y=0

for x in a,b,c,d,e,f,g:
    y+=int(x)
print(y)

```

```

i=0
npm=input("Masukan NPM : ")

```



```

while i < 1:
    if len(npm) < 7:
        print("NPM_Kurang_dari_7_digit")
        npm=input("Masukan_NPM_: ")
    elif len(npm) > 7:
        print("NPM_lebih_dari_7_digit")
        npm=input("Masukan_NPM_: ")
    else:
        i=1

```

```
npm6(npm)
```

```
def npm7(npm):
```

```

    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]
    conv=1

    for x in a,b,c,d,e,f,g:
        conv*=int(x)
    print(conv)

```

```
i=0
```

```
npm=input("Masukan_NPM_: ")
```

```

while i < 1:
    if len(npm) < 7:

```

```

        print("NPM_Kurang_dari_7_digit")
        npm=input("Masukan_NPM_: ")
    elif len(npm) > 7:
        print("NPM_lebih_dari_7_digit")
        npm=input("Masukan_NPM_: ")
    else :
        i=1
npm7(npm)

```

```

8. def npm8(npm):
    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]
    for x in a,b,c,d,e,f,g:
        if int(x)%2==0:
            if int(x)==0:
                x=""
            print(x,end=" ")

```

```

i=0
npm=input("Masukan_NPM_: ")
while i<1:
    if len(npm) < 7:
        print("NPM_Kurang_dari_7_digit")
        npm=input("Masukan_NPM_: ")

```

```

    elif len(npm) > 7:
        print("NPM lebih dari 7 digit")
        npm=input("Masukan NPM: ")
    else:
        i=1
npm8(npm)

def npm9(npm):
    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]
    for x in a,b,c,d,e,f,g:

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

i=0
npm=input("Masukan NPM: ")
while i<1:
    if len(npm) < 7:
        print("NPM Kurang dari 7 digit")
        npm=input("Masukan NPM: ")
    elif len(npm) > 7:
        print("NPM lebih dari 7 digit")
        npm=input("Masukan NPM: ")
    else:

```

```

        i=1
    npm9(npm)

```

```

19. def npm10(npm):
    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]
    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 =""),

    i=0
    npm=input("Masukan NPM : ")
    while i<1:
        if len(npm) < 7:
            print("NPM Kurang dari 7 digit")
            npm=input("Masukan NPM : ")
        elif len(npm) > 7:
            print("NPM lebih dari 7 digit")
            npm=input("Masukan NPM : ")
        else:
            i=1

```

[illegible]

```

        for x in a,b,c,d,e,f,g:
            print(x)

def npm6(npm):

    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]
    y=0

    for x in a,b,c,d,e,f,g:
        y+=int(x)
    print(y)
def npm7(npm):

    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]
    conv=1

    for x in a,b,c,d,e,f,g:

```

```

        conv*=int(x)
    print(conv)
def npm8(npm):
    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]
    for x in a,b,c,d,e,f,g:
        if int(x)%2==0:
            if int(x)==0:
                x=""
            print(x,end=" ")
def npm9(npm):
    a=npm[0]
    b=npm[1]
    c=npm[2]
    d=npm[3]
    e=npm[4]
    f=npm[5]
    g=npm[6]
    for x in a,b,c,d,e,f,g:

        if int(x)%2==1:
            print(x,end=" ")
def npm10(npm):
    a=npm[0]

```

```

b=npm[1]
c=npm[2]
d=npm[3]
e=npm[4]
f=npm[5]
g=npm[6]
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. **import lib3**

```

class Kelas3lib:
    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)

import lib3
class Kelas3lib:
    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)
```