

Tugas Praktikum Pemrograman II

(Chapter 3)



Disusun oleh :

Dinda Anik Masruro

D4 TI 2C

1.18.4.003

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 adalah inputan sebuah fungsi yang bertujuan untuk menyimpan sebuah nilai.
3. return berfungsi untuk mengembalikan nilai yang telah di proses dalam suatu fungsi dan mengakhiri sebuah fungsi

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

4. item paket adalah sebuah direktory dengan file python dan file dengan nama `_init_.py`. jadi sebuah direktori didalam sebuah python dengan nama `_init_.py`, akan dianggap sebagai paket oleh python. untuk memanggil sebuah paket atau library adalah dengan cara *import* nama paket atau library tersebut lalu paket atau library tersebut dapat di gunakan.

```
from kampus import mahasiswa
```

5. class adalah sebuah blueprint dari sebuah objek yang akan di bangun.

```
class World:  
    def __init__(self,World):  
        self.World = World  
    def heloWorld(self):  
        print("Helo",World)
```

6. objek memiliki variable dan kode yang saling terhubung. objek di buat dengan class.

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

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

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

8. method adalah sebuah fungsi dalam 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 libra :

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

10. contoh jika kita ingin memanggil fungsi dari library pada main program kita harus terlebih dahulu 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 berarti program 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 lalu mengimport nama librarynya, contoh :

```
from mahasiswa import kampus
```

artinya dalam package mahasiswa akan memakai library kampus

13. pemanggilan class dalam sebuah folder

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

```
from mahasiswa import kampus
```

artinya dalam package mahasiswa kita akan memakai class kampus

1.2 Ketrampilan Pemrograman

```
1. # -*- coding: utf-8 -*-
    """
```

Created on Fri Nov 8 23:07:18 2019

@author: *Dinda Anik Masruro*

” ” ”

```
def npm1():
```

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

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

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

```
print("_+++_+++_+++____++++_____+___+__+__+__+__+__+__+__")
```

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

```
# -*- coding: utf-8 -*-
"""
```

Created on Fri Nov 8 23:09:05 2019

@author: Dinda Anik Masruro

” ” ”

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

```
npm= i n t ( npm )
```

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

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

```
print("Hallo ,_", npm, "_apa_kabar_?")
```

```
3. # -*- coding: utf-8 -*-
    """
```

Created on Fri Nov 8 23:10:11 2019

@author: Dinda Anik Masruro

"""

```
def npm3(npm):  
    for i in range(int(str(npm)[4])+int(str(npm)[5])+int(str(npm)[6])):  
        print("Halo , "+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)
```

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

"""

Created on Fri Nov 8 23:11:56 2019

@author: Dinda Anik Masruro

"""

```
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)

```

4. # -*- coding: utf-8 -*-

"""

Created on Fri Nov 8 23:12:54 2019

@author: Dinda Anik Masruro

"""

```

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)

# -*- coding: utf-8 -*-
"""
Created on Fri Nov 8 23:14:01 2019

@author: Dinda Anik Masruro
"""

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)

```

```

6. # -*- coding: utf-8 -*-
"""

```

Created on Fri Nov 8 23:15:21 2019

@author: Dinda Anik Masruro

```

"""
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)

# -*- coding: utf-8 -*-

```

"""

Created on Fri Nov 8 23:16:04 2019

@author: Dinda Anik Masruro

"""

```
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)

8. # -*- coding: utf-8 -*-
    """
    Created on Fri Nov 8 23:16:45 2019

    @author: Dinda Anik Masruro

    """
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)

# -*- coding: utf-8 -*-
"""
Created on Fri Nov  8 23:17:33 2019

@author: Dinda Anik Masruro

"""
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 =""),

```



```

        for i in range(TwoLastDigit):
            print("Halo , ", npm, " _apa _kabar _?")
def npm3(npm):
    for i in range(int(str(npm)[4])+int(str(npm)[5])+int(str(npm)[6])):
        print("Halo , "+str(npm)[4]+str(npm)[5]+str(npm)[6]+" _apa _kabar _?")
    return None
def npm4(npm):
    key=npm%1000
    str_key=str(key)
    print("Halo , "+str_key[0]+" _apa _kabar _?")
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)

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 ="" ) ,

# -*- coding: utf-8 -*-
"""

Created on Fri Nov 8 23:02:37 2019

@author: Dinda Anik Masruro
"""

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)

```

13. # -*- coding: utf-8 -*-

"""

Created on Fri Nov 8 23:05:37 2019

@author: Dinda Anik Masruro

"""

```

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)
```