

Laporan Tugas

Chapter III



Muhammad Wahyu Ardi Ismail

NPM : 1.18.4.059

Program Diploma IV Teknik Informatika

Applied Bachelor Program of Informatics Engineering

Bandung 2019

Chapter 1

Fungsi dan Kelas

1.1 Teori

1.1.1 Fungsi

1. Fungsi

Fungsi yaitu sebuah blok kode yang akan dieksekusi ketika dipanggil dalam suatu program

2. Parameter

Parameter yaitu inputan sebuah fungsi bertujuan sebagai menyimpan sebuah nilai

3. return

Digunakan untuk mengembalikan sebuah nilai, bisa juga untuk mengakhiri eksekusi sebuah fungsi

```
def fungsi(a,b):  
    c=a+b  
    return c
```

1.1.2 Package

1. Package

package yaitu sebuah folder yang menyimpan source code, misalnya pada tempat kita menyimpan main program, kita membuat sebuah folder motor dan di dalamnya kita membuat sebuah source code dengan nama mesin.py.

1.1.3 Class

1. Class

class merupakan blueprint dari sebuah object, jika diibaratkan membuat sebuah kue, class merupakan cetakan kuenya, contoh kode :

```
class Nama:
    def __init__(self,nama):
        self.nama = nama
    def helonama(self):
        print("Helo",nama)
```

2. Object

object merupakan hasil cetakan dari sebuah class contoh kode :

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

3. Attribute

attribute merupakan variabel global yang dimiliki oleh sebuah class

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

4. Method

method merupakan fungsi-fungsi dalam sebuah class

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

import kelas3lib
cobakelas=nama.Nama(nama)
#pemanggilan method pada program
hasilkelas=cobakelas.nama()
```

1.1.4 Penggunaan library

contoh membuat sebuah library, contoh disini kita membuat pada folder libra :

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

contoh jika kita ingin memanggil fungsi dari library pada main program kita harus terlebih dahulu import :

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

1.1.5 pemakaian package from kalkulator import penambahan

```
from kalkulator import penambahan
```

kode diatas yaitu program memanggil sebuah package terlebih dahulu, setelah itu menambahkan source code penambahan, kode diatas dapat dibaca seperti ini "import penambahan dari folder kalkulator", contoh lainnya :

```
from dapur import memasak
```

1.1.6 pemanggilan library dalam sebuah folder

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

```
from me import libheart
```

artinya dalam package me kita akan memakai library libheart

1.1.7 pemanggilan class dalam sebuah folder

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

```
from me import clheart
```

artinya dalam package me kita akan memakai class cheart

Chapter 2

Keterampilan pemrograman

1. Soal 1

```
def NPM1():
    print ("###_###_#####_###_###_#####_#####_#####")
    print ("###_###_###_###_###_###_###_###_###_###_###_###")
    print ("###_###_#####_#####_###_###_#####_#####")
    print ("###_###_###_###_###_###_###_###_###_###_###_###")
    print ("###_###_#####_###_###_#####_#####_#####")
```

2. Soal 2

```
def npm2(npm):
    npm = input("masukan npm: ")
    key = int(npm)%100
    for i in range(key):
        print("halo", npm, "apa kabarmu?")
```

3. Soal 3

```
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 ")

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)

```

4. Soal 4

```

def npm4(npm):
    key=npm%1000
    str_key=str(key)
    print("Halo , "+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)

```

5. Soal 5

```

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. Soal 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)

```

7. Soal 7

```

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. Soal 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)

```

9. Soal 9

```

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)

```

10. Soal 10

```

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
npm10(npm)

```

11. Soal 11

```

def npm1():
    print("#####_#####_#####_#####_#####_#####_#####_#####")
    print("#####_#####_#####_#####_#####_#####_#####_#####_#####_#####")
    print("#####_#####_#####_#####_#####_#####_#####_#####_#####_#####")
    print("#####_#####_#####_#####_#####_#####_#####_#####_#####_#####")
    print("#####_#####_#####_#####_#####_#####_#####_#####_#####_#####")
def npm2(npm):
    npm=int(npm)
    TwoLastDigit=abs(npm)%100
    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 ="" ),

```

12. Soal 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)

```

13. main.py

```

import kelas3lib
import lib3

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

#Contoh pemanggilan fungsi pada class
cobakelas=kelas3lib.Kelas3lib(npm)
hasilkelas=cobakelas.npm1()

print("")

#Contoh pemanggilan fungsi pada library
lib3.npm3(npm)

```

Keterampilan penanganan error

2.0.1 penanganan error

error :

TypeError: __init__() missing 1 required positional argument: 'npm'

solusi :

menambahkan parameter pada fungsi

Try except

```
def pembagian(a,b):  
    c=a/b  
    return c  
  
d=int(input("angka pertama : "))  
e=int(input("angka kedua : "))  
try:  
    print(pembagian(d,e))  
except:  
    print("jangan masukan angka 0")
```