

CHAPTER 3

**Diajukan untuk memenuhi kelulusan matakuliah
Pemrograman II pada Program Studi DIV Teknik
Informatika**

O l e h :

Echa Dwiifanka

1.18.4.022



PROGRAM DIPLOMA IV TEKNIK INFORMATIKA

POLITEKNIK POS INDONESIA

BANDUNG

2019

BAB I

PEMAHAMAN TEORI

1.1 Fungsi,parameter,dan return

fungsi

Fungsi adalah blok kode terorganisir dan dapat digunakan kembali yang digunakan untuk melakukan sebuah tindakan/action. Fungsi memberikan modularitas yang lebih baik untuk aplikasi Anda dan tingkat penggunaan kode yang tinggi.

parameter

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

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
```

1.2 paket

paket atau biasa disebut dengan package adalah sebuah tempat atau wadah yang nantinya akan digunakan untuk tempat penyimpanan sebuah kode program, dan untuk menjalankannya kita nanti hanya memanggilnya saja.misalnya kita membuat sebuah kode program yang akan diberi nama dengan *Bangun datar*, dan untuk memanggilnya dapat dilihat sebagai berikut:

```
from Segitiga import bangun datar
```

1.3 Class,objek,Atribute,dan Method

Class

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

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

Objek

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()
```

Atribute

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
```

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 ?")
```

1.4 contoh membuat library

1. pertama-tama kita akan membuat sebuahkode program yang akan disimpan pada folder library:

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

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

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

1.5 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"

1.6 contoh pemakaian paket fungsi dan 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 segitiga import bangun datar
```

artinya dalam package segitiga akan memakai atau menggunakan suatu library bangun datar.

1.7 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 segitiga import bangun data
```

artinya dalam package segitiga akan memakai atau menggunakan suatu class bangun datar.

BAB II

Ketrampilan Pemrograman

soal 1

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

soal 2

```
def npm2(npm):
    npm=int(npm)
    TwoLastDigit=abs(npm)%100
    for i in range(TwoLastDigit):
        print("Halo, ", npm, " _apa_kabar_?")
```

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)
```

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)
```

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)
```

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)
```

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)

```

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)

```

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)

```

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:

```



```

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 = ""),

```

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)
```

BAB III

Keterampilan Penanganan Error

A Penanganan error

error:

TypeError: init ()missing 1 required postional argument:'npm'

solusi:

menambahkan parameter pada fungsi

B Try Except

```
def pembagian(a,b):
```

```
    c=a/b
```

```
    return c
```

```
d=int(input("pembilang:"))
```

```
e=int(input("penyebut:"))
```

```
try:
```

```
    print(pembagian(d,e))
```

```
except:
```

```
    print("jangan masukan angka 0")
```