

# **CERDAS MENGUASAI PYTHON**



---

# CERDAS MENGUASAI PYTHON

## Dalam 24 Jam

---

**Rolly M. Awangga**  
Informatics Research Center



**Kreatif Industri Nusantara**

***Penulis:***

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

***Desain sampul dan Tata letak:***

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.’  
Imam Syafi’i*

# CONTRIBUTORS

---

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1</b>	<b>SEJARAH DAN KARAKTERISTIK PYTHON</b>	<b>1</b>
<b>2</b>	<b>Judul Bagian Kedua</b>	<b>3</b>
<b>3</b>	<b>Judul Bagian Ketiga</b>	<b>5</b>





# DAFTAR ISI

---

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xix
Kata Pengantar	xxi
Acknowledgments	xxiii
Acronyms	xxv
Glossary	xxvii
List of Symbols	xxix
Introduction	xxxi
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
<b>1 SEJARAH DAN KARAKTERISTIK PYTHON</b>	<b>1</b>
<b>2 Judul Bagian Kedua</b>	<b>3</b>
<b>3 Judul Bagian Ketiga</b>	<b>5</b>
	ix

3.1	D. Irga B. Naufal Fakhri	5
3.1.1	Pemahaman Teori	5
3.1.2	Keterampilan Pemograman	8
3.1.3	Ketrampilan Penanganan Error	11
3.2	Nurul Izza Hamka   1174062   Teori	11
3.3	Fanny Shafira Damayanti   1174069	16
3.4	Teori	16
3.5	keterampilan pemrograman	18
3.6	penanganan Error	21
3.7	1174054   Aulyardha Anindita	21
3.8	Pemahaman Teori	21
3.8.1	Fungsi	21
3.8.2	Paket	22
3.8.3	Kelas, Objek, Atribut, dan Method	22
3.8.4	Cara Pemanggilan Library Kelas dari Instansiansi	23
3.8.5	Pemakaian Paket dengan Perintah from dan import	23
3.8.6	Pemakaian Paket Fungsi	23
3.8.7	Pemakaian Paket Kelas	23
3.9	Keterampilan Pemrograman	24
3.10	Penanganan Error	27
3.11	Dini Permata Putri	27
3.12	ainul filiani	31
3.13	Chandra Kirana Poetra	34
3.13.1	Pemahaman Teori	34
3.13.2	Praktek	37
3.13.3	Praktek	40
	Daftar Pustaka	41
	Index	43

# DAFTAR GAMBAR

---



# DAFTAR TABEL

---



# Listings

---

3.1	Contoh kode fungsi inputan ke fungsi.	5
3.2	Contoh kode fungsi outputan ke fungsi.	5
3.3	Contoh import package atau library.	6
3.4	Contoh import package atau library.	6
3.5	Contoh package atau library.	7
3.6	Contoh import package atau library.	7
3.7	Contoh import package atau library.	7
3.8	Contoh import package atau library didalam folder.	7
3.9	Contoh import package atau library didalam folder.	7
	src/1174066.py	8
	src/1174066.py	8
	src/1174066.py	8
	src/1174066.py	9
	src/1174066.py	9
	src/1174066.py	9
	src/1174066.py	9
	src/1174066.py	9
	src/1174066.py	9



src/1174066.py	10
src/main_1174066.py	10
src/main_1174066.py	10
src/main_1174066.py	11
src/1174062.py	12
src/kelas3lib.py	12
src/1174062.py	12
src/1174062.py	13
src/1174062.py	13
src/1174062.py	13
src/1174062.py	13
src/1174062.py	14
src/1174062.py	15
src/1174062.py	15
src/1174062.py	15
src/1174062.py	15
src/1174062.py	16
src/1174062.py	16
src/1174062.py	16
src/1174062.py	16
src/1174069/1174069.py	17
src/1174069/1174069.py	17
src/1174069/1174069.py	17
src/1174069/1174069.py	17
src/1174069/1174069.py	18
src/1174069/main_1174069.py	18
src/1174069/main_1174069.py	18
src/1174069/1174069.py	18
src/1174069/1174069.py	19
src/1174069/1174069.py	19
src/1174069/1174069.py	20
src/1174069/1174069.py	20
src/1174069/1174069.py	20
src/1174069/1174069.py	20
src/1174069/1174069.py	20
src/1174069/1174069.py	20
src/1174069/main_1174069.py	21

src/1174069/main_1174069.py	21
src/1174069/1174069.py	21
src/1174054/1174054.py	21
src/1174054/1174054.py	22
src/1174054/1174054.py	22
src/1174054/1174054.py	22
src/1174054/1174054.py	22
src/1174054/1174054.py	23
src/1174054/1174054.py	23
src/1174054/main_1174054.py	23
src/1174054/main_1174054.py	23
src/1174054/1174054.py	24
src/1174054/1174054.py	25
src/1174054/1174054.py	25
src/1174054/1174054.py	25
src/1174054/1174054.py	26
src/1174054/1174054.py	26
src/1174054/1174054.py	26
src/1174054/1174054.py	26
src/1174054/1174054.py	26
src/1174054/1174054.py	26
src/1174054/main_1174054.py	27
src/1174054/main_1174054.py	27
src/1174054/1174054.py	27
src/1174079/1174079.py	34
src/1174079/1174079.py	34
src/1174079/1174079.py	35
src/1174079/1174079.py	35
src/1174079/1174079.py	35
src/1174079/1174079.py	36
src/1174079/1174079.py	36
src/1174079/1174079.py	36
src/1174079/1174079.py	36
src/1174079/1174079.py	37
src/1174079/1174079.py	37
src/1174079/1174079.py	38
src/1174079/1174079.py	38
src/1174079/1174079.py	38

src/1174079/1174079.py	38
src/1174079/1174079.py	38
src/1174079/1174079.py	39
src/1174079/1174079.py	39
src/1174079/1174079.py	39
src/1174079/main.py	39
src/1174079/main.py	40
src/1174079/1174079.py	40

# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat  
Februari, 2019*



# ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.





# ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association



# GLOSSARY

---

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald



# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
  
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$





# BAB 1

---

## SEJARAH DAN KARAKTERISTIK PYTHON

---



## BAB 2

---

## JUDUL BAGIAN KEDUA

---



## BAB 3

---

# JUDUL BAGIAN KETIGA

---

### 3.1 D. Irga B. Naufal Fakhri

#### 3.1.1 Pemahaman Teori

##### 1. Fungsi

Fungsi adalah blok blok kode yang terorganisir yang dapat digunakan kembali didalam program yang digunakan untuk melakukan suatu perintah yang telah diberikan. untuk membuat fungsi kita harus menggunakan def kemudian nama fungsinya dan (variable)nya diakhiri oleh tanda :

```
1 #Contoh fungsi inputan
2 def printteks( str ):
3     print ( str )
4     return
5
6 printteks("Ini adalah Contoh fungsi")
```

**Listing 3.1** Contoh kode fungsi inputan ke fungsi.

Fungsi juga berguna untuk melemparkan variable contohnya

```

1 #Contoh fungsi outputan
2 def output( str ):
3     full = "Ini adalah contoh dari " + str
4     return full
5
6 print( output("fungsi outputan"))

```

**Listing 3.2** Contoh kode fungsi outputan ke fungsi.

## 2. Paket(Package) atau Library

Paket atau yang biasa disebut dengan library adalah kumpulan kode-kode fungsi atau method pada python yang dapat dipanggil kedalam program python yang kita buat. Package berada di file terpisah dari main program cara memanggil package: Pastikan file package ada didalam folder yang sama lalu ditambah import dengan nama filenya tanpa extensi (.py)

```

1 #Contoh import library atau package
2 import fungsi_1174066
3
4 input( fungsi_1174066.hello("Dirga"))

```

**Listing 3.3** Contoh import package atau library.

## 3. Kelas (Class), Objek (Object), Atribut (Attribute), dan Method

Kelas(Class) adalah sebuah blueprint(cetakan) dari sebuah objek. Objek(Object) adalah hasil cetakan dari sebuah kelas(class). Atribut(Attribute) adalah nilai data yang ada didalam sebuah object. Method adalah sesuatu yang bisa dilakukan oleh object.

```

1 #Contoh Class
2 class Mahasiswa:
3     def __init__( self , nama ,npm, kelas ) :
4         self.nama = nama
5         self.npm = npm
6         self.kelas = kelas
7
8     def datadiri( self ):
9         print ( "Nama: ", self.nama, ", NPM: ", self.npm, ",
10             Kelas: ", self.kelas )
11
12 #Pembuatan Object
13 mhs1 = Mahasiswa("D. Irga",11174066,"D4TI2C")
14 mhs1.datadiri() ;

```

**Listing 3.4** Contoh import package atau library.

## 4. Cara memanggil library dari instansiasi

Cara memanggilnya:

- Pertama kita import filenya
- kemudian buat variabelnya jika menggunakan variable untuk menampung data

- Kemudian panggil nama classnya(file) dan panggil fungsinya
- Kemudian menggunakan perintah print untuk menampilkan data

```

1 #Contoh package nomor 4
2 def Tambah(x,y):
3     z = x + y
4     return z

```

**Listing 3.5** Contoh package atau library.

```

1 #Contoh cara memanggil library atau package
2 import fungsi_1174066
3
4 x = 3
5 y = 7
6 print(fungsi_1174066.Tambah(x,y))

```

**Listing 3.6** Contoh import package atau library.

5. Contoh pemakaian paket dengan perintah from kalkulator import Penambahan  
Pemakaian package(paket) dengan perintah from namafilenya import berfungsi untuk memanggil fungsi dari nama filenya

```

1 #Contoh cara memanggil library atau package dari from import
2 from fungsi_1174066 import Tambah
3
4 x = 3
5 y = 7
6 print(Tambah(x,y))

```

**Listing 3.7** Contoh import package atau library.

6. Jelaskan dengan contoh kode, pemakaian paket fungsi didalam folder  
Jika file paket ada didalam folder maka kita harus menambahkan lokasi filenya ada didalam folder apa dengan cara menggunakan namafolder.namafile

```

1 #Contoh cara memanggil library atau package dari folder
2 from 1174066.fungsi_1174066 import Tambah
3
4 x = 3
5 y = 7
6 print(Tambah(x,y))

```

**Listing 3.8** Contoh import package atau library didalam folder.

7. Jelaskan dengan contoh kode, pemakaian paket fungsi didalam folder  
Jika file paket ada didalam folder maka kita harus menambahkan lokasi filenya ada didalam folder apa dengan cara menggunakan namafolder.namafile

```

1 #Contoh cara memanggil library atau package dari folder
2 from 1174066.fungsi_1174066 import Tambah
3
4 x = 3

```



```

5 y = 7
6 print(Tambah(x,y))

```

**Listing 3.9** Contoh import package atau library didalam folder.

### 3.1.2 Keterampilan Pemograman

#### 1. Jawaban nomor 1

```

1 #Jawaban nomor 1
2 def npm(npm):
3     text = str(npm)
4     from PIL import Image, ImageDraw, ImageFont
5     import numpy as np
6     myfont = ImageFont.truetype("verdanab.ttf", 12)
7     size = myfont.getsize(text)
8     img = Image.new("1",size,"black")
9     draw = ImageDraw.Draw(img)
10    draw.text((0, 0), text, "white", font=myfont)
11    pixels = np.array(img, dtype=np.uint8)
12    if int(text) % 3 == 0:
13        chars = np.array([' ', '*'], dtype="U1")[pixels]
14    elif int(text) % 3 == 1:
15        chars = np.array([' ', '#'], dtype="U1")[pixels]
16    elif int(text) % 3 == 2:
17        chars = np.array([' ', '+'], dtype="U1")[pixels]
18    strings = chars.view('U' + str(chars.shape[1])).flatten()
19    print( "\n".join( strings ))

```

#### 2. Jawaban nomor 2

```

1 #Jawaban nomor 2
2 def loop(npm):
3     i = 1
4     while(i <= int(npm[5:])):
5         print("Hallo, "+str(npm)+" apa kabar?")
6         i += 1
7
8 loop(input("Masukan NPM : "))

```

#### 3. Jawaban nomor 3

```

1 #Jawaban nomor 3
2 def looptambah(npm):
3     i = 1
4     akhir = int(npm[4:5]) + int(npm[5:6]) + int(npm[6:7])
5     while(i <= akhir):
6         print("Hallo, "+str(npm[4:7])+" apa kabar?")
7         i += 1
8
9 looptambah(input("Input: "))

```

#### 4. Jawaban nomor 4

```
1 #Jawaban nomor 4
2 def hallo(npm):
3     print("Halo, "+str(npm[-3])+ " apa kabar?")
4
5 hallo(input("Input: "))
```

## 5. Jawaban nomor 5

```
1 #Jawaban nomor 5
2 def array(npm):
3     for x in npm:
4         print(x)
```

## 6. Jawaban nomor 6

```
1 #Jawaban nomor 6
2 def jumlah(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil += x
7     print("Hasil perjumlahan dari npm anda adalah: " + str(hasil))
8
9 jumlah(input("NPM: "))
```

## 7. Jawaban nomor 7

```
1 #Jawaban nomor 7
2 def kali(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil *= x
7     print("Hasil perkalian dari npm anda adalah: " + str(hasil))
8
9 kali(input("NPM: "))
```

## 8. Jawaban nomor 8

```
1 #Jawaban nomor 8
2 def printgenap(npm):
3     npm = list(map(int, npm))
4     for x in npm:
5         if(x % 2 == 0):
6             if(x != 0):
7                 print(x, end = " ")
8
9 printgenap(input("NPM: "))
```

## 9. Jawaban nomor 9

```
1 #Jawaban nomor 9
2 def printganjil(npm):
3     npm = list(map(int, npm))
```

```

4     for x in npm:
5         if(x % 2 != 0):
6             print(x, end = "")
7
8     printganjil(input("NPM: "))

```

## 10. Jawaban nomor 10

```

1  #Jawaban nomor 10
2  def printprima(npm):
3      npm = list(map(int, npm))
4      prima = []
5      for n in npm:
6          bilPrima = True
7          if n == 0 or n == 1:
8              bilPrima = False
9          for x in range(2, n):
10             if n % x == 0:
11                 bilPrima = False
12             if bilPrima:
13                 prima.append(n)
14
15     for p in prima:
16         print(p, end = "")
17
18     printprima(input("NPM: "))

```

## 11. Jawaban nomor 11

```

1  #Pemanggilan library 3lib
2  from lib_1174066 import *
3
4  npm(1174066)
5  loop(input("Masukan NPM : "))
6  looptambah(input("Input: "))
7  hallo(input("Input: "))
8  array(input("NPM: "))
9  jumlah(input("NPM: "))
10 kali(input("NPM: "))
11 printgenap(input("NPM: "))
12 printganjil(input("NPM: "))
13 printprima(input("NPM: "))

```

## 12. Jawaban nomor 12

```

1  #Pemanggilan kelas 3lib
2  from kelas3lib_1174066 import kelas3lib
3  npm = "1174066"
4  kelas = kelas3lib(npm)
5
6  kelas.npmprint()
7  kelas.loop()
8  kelas.looptambah()
9  kelas.hallo()
10 kelas.array()
11 kelas.jumlah()

```

```

12 kelas.kali()
13 kelas.printgenap()
14 kelas.printganjil()
15 kelas.printprima()

```

### 3.1.3 Ketrampilan Penanganan Error

- Syntax Errors

Syntax Errors adalah kesalahan pada penulisan syntax atau kode. Solusinya adalah memperbaiki penulisan syntax atau kode

- Zero Division Error

ZeroDivisonError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.

- Name Error

NameError adalah exception saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi atau tidak ada. Solusinya adalah memastikan variabel atau function yang akan dipanggil ada didalam program atau tidak salah mengetikannya.

- Type Error

TypeError adalah exception saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkoversi variabelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

```

1 #Pemanggilan kelas 3lib
2 from kelas3lib_1174066 import kelas3lib
3 npm = "1174066"
4 kelas = kelas3lib(npm)
5
6 kelas.npmprint()
7 kelas.loop()
8 kelas.looptambah()
9 kelas.hallo()
10 kelas.array()
11 kelas.jumlah()
12 kelas.kali()
13 kelas.printgenap()
14 kelas.printganjil()
15 kelas.printprima()

```

## 3.2 Nurul Izza Hamka | 1174062 | Teori

1. Fungsi pada python menggunakan kata kunci def. setelah menulis kata def, kita tulis lagi nama fungsi kemudian diikuti dengan parameter yang diberi tanda

kurung dan diakhirnya diberi tanda titi dua (:). Setelah itu kita menulis lagi fungsi yang akan di panggil untuk di jalankan. Inputan fungsi adalah untuk memanggil fungsi dengan menuliskan nama fungsi. Kembalian fungsi adalah keluaran fungsi dan untuk kembali ke baris selanjutnya untuk memanggil fungsi tadi.

```
1 def Izzah() :
2     print ("Apa Kabar Izzah")
3 Izzah()
```

2. Paket adalah sebuah file yang berisi kode program python yang bisa digunakan berulang ketika sebuah paket itu dipanggil.

```
1 #from kelas3lib import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))
```

3. Apa itu kelas, apa itu objek, apa itu atribut, apa itu method :

- Kelas adalah sebuah objek yang di dalamnya terdapat sebuah metode atau seperangkat atribut.
- Objek adalah struktur data yang di definisikan dalam kelas, objek ini memiliki atribut dan juga aksi /behaviour.
- Atribut adalah data dari variable kelas dan juga method.
- Method adalah sebuah kode yang di gunakan untuk melakukan perintah.

```
1 #class Employee:
2 #     'Common base class for all employees'
3 #     empCount = 0
4
5 #     def __init__(self, name, salary):
6 #         self.name = name
7 #         self.salary = salary
8 #         Employee.empCount += 1
9
10 #     def displayCount(self):
11 #         print ("Total Employee %d" % Employee.empCount)
12
13 #     def displayEmployee(self):
14 #         print ("Name : ", self.name, ", Salary : ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Izza", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Manni", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)
```

#### 4. Pemanggilan library kelas dari instansiasi

- Pertama, lakukan import file.
- Membuat variable untuk menyimpan data.
- Lakukan pemanggilan data dan class.
- Ketikkan print untuk melihat hasilnya.

```

1 #import belajar
2 #a = 10
3 #b = 5
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

#### 5. Penggunaan paket from kalkulator import untuk penambahan berfungsi untuk memanggil file yang di masukkan dan juga fungsinya.

```

1 #import belajar
2 #a = 10
3 #b = 5
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

#### 6. Pemakaian paket fungsi adalah sebuah kumpulan fungsi-fungsi.

#### 7. Pemakaian paket kelas apabila file library ada di dalam folder.

```

1 # Employee.empCount += 1

```

#### 8. No1

```

1 #No 1
2 def penulisan(npm):
3     npm = list(str(npm))
4
5     angka1 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
##### ", "4": " ###   ## ", "5": " ##### ", "6": " ##### ", "7": "
##### ", "8": " ##### ", "9": " ##### "}
6     angka2 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
##### ", "4": " ###   ## ", "5": " ##### ", "6": " ##### ", "7": "
##### ", "8": " ##### ", "9": " ##### "}
7     angka3 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
##### ", "4": " ###   ## ", "5": " ##### ", "6": "   ##   ", "7": "
   ##   ", "8": " ##### ", "9": " ##### "}
8     angka4 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
##### ", "4": " ###   ## ", "5": " ##### ", "6": "   ##   ", "7": "
   ##   ", "8": " ##### ", "9": " ##### "}

```

```

9     angka5 = {"0": " ##      ## ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
10    angka6 = {"0": " ##      ## ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
11    angka7 = {"0": " ##      ## ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
12    angka8 = {"0": " ##      ## ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
13    angka9 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
14    angka10 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}

15
16    hasil1 = []
17    hasil2 = []
18    hasil3 = []
19    hasil4 = []
20    hasil5 = []
21    hasil6 = []
22    hasil7 = []
23    hasil8 = []
24    hasil9 = []
25    hasil10 = []
26
27    for x in npm:
28        hasil1.append(angka1[x])
29        hasil2.append(angka2[x])
30        hasil3.append(angka3[x])
31        hasil4.append(angka4[x])
32        hasil5.append(angka5[x])
33        hasil6.append(angka6[x])
34        hasil7.append(angka7[x])
35        hasil8.append(angka8[x])
36        hasil9.append(angka9[x])
37        hasil10.append(angka10[x])
38
39    print(*hasil1, sep=' ')
40    print(*hasil2, sep=' ')
41    print(*hasil3, sep=' ')
42    print(*hasil4, sep=' ')
43    print(*hasil5, sep=' ')
44    print(*hasil6, sep=' ')
45    print(*hasil7, sep=' ')
46    print(*hasil8, sep=' ')
47    print(*hasil9, sep=' ')
48    print(*hasil10, sep=' ')

```

## 9. no2

```

2 def perulangan(npm):
3     hitung = 0
4     while(hitung < 62):
5         print("Halo, "+str(npm)+" apa kabar?")
6         hitung = hitung +1
7
8 perulangan(int(input("Masukan NPM : ")))

```

## 10. no3

```

1 #No 3
2 def perulangan_3_digit(npm):
3     hitung = 0
4     npm = str(npm)
5     bil = npm[4:7]
6
7     while(hitung < 8):
8         print("Halo, "+bil+" apa kabar?")
9         hitung = hitung +1
10
11 perulangan_3_digit(int(input("Masukan NPM : ")))

```

## 11. no4

```

1 #No 4
2 def perulangan_3_digit_terakhir(npm):
3     npm = str(npm)
4     bil = npm[-3]
5     print("Halo, "+bil+" apa kabar?")
6
7 perulangan_3_digit_terakhir(int(input("Masukan NPM : ")))

```

## item no5

```

1 #No 5
2 def down(npm):
3     for i in npm:
4         print(i)
5
6 down(input("Masukan NPM : "))

```

## 12. no6

```

1 #No 6
2 def penjumlahan(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah += int(i)
6     print(str(jumlah)+" Adalah hasil penjumlahan")
7
8 penjumlahan(input("Masukan NPM : "))

```

## 13. no7



```

1 #No 7
2 def perkalian(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah *= int(i)
6     print(str(jumlah)+" Adalah hasil perkalian")
7
8 perkalian(input("Masukan NPM : "))

```

## 14. no8

```

1 #No 8
2 def genap():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i % 2) == 0:
6             print("Bilangan Genapnya : "+str(i))
7 genap()

```

## 15. no9

```

1 #No 9
2 def ganjil():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i%2)==1:
6             print("Bilangan Ganjilnya : "+str(i))
7 ganjil()

```

## 16. no10

```

1 #No 10
2 def prima(npm):
3     npm = str(npm)
4     bil = npm[2]
5     num = int(bil)
6     if num > 1:
7         for i in range(2,num):
8             if (num%i)==0:
9                 print("Bukan Bilangan Prima")
10                break
11            else:
12                print("Bilangan Primanya : "+str(num))
13    else:
14        print("Tidak Ada Bilangan Prima")
15    prima(int(input("Masukan NPM : ")))

```

## 3.3 Fanny Shafira Damayanti | 1174069

## 3.4 Teori

1. fungsi adalah sebuah program untuk melakukan tugas tertentu secara berulang. Fungsi di dalam Python di tandai dengan def, yang artinya definition. Inputan

fungsi berarti memanggil fungsi yang ditulis oleh user dan dikembalikannya dalam bentuk string.

```
1 def nyapa():
2     print("anyeong Fanny")
3
4 nyapa()
```

2. paket adalah kumpulan fungsi yang siap untuk di pakai. Paket digunakan untuk memudahkan programmer agar tidak menuliskan kembali kode-kodenya.

```
1 #from fungsi_fanny import *
2 #print(penulisan(int(input("Masukan NPM kamu :"))))
```

3. objek adalah instansi dari kelas. Kelas adalah cetakan biru yang berisikan variable dan method. Method adalah fungsi dari suatu objek. Atribut adalah nilai dari suatu objek.

```
1 #class Employee:
2 #     'Common base class for all employees'
3 #     empCount = 0
4
5 #     def __init__(self, name, salary):
6 #         self.name = name
7 #         self.salary = salary
8 #         Employee.empCount += 1
9
10 #     def displayCount(self):
11 #         print("Total Employee %d" % Employee.empCount)
12
13 #     def displayEmployee(self):
14 #         print("Name : ", self.name, ", Salary : ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Fanny", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Shafira", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print("Total Employee %d" % Employee.empCount)
```

4. cara pemanggilan library yaitu :

- Import file yang akan di panggil
- Buat variable nya
- Panggil nama class dan methodnya
- Print untuk menampilkan outputnya

```
1 #import latihan
2 #a = 30
3 #b = 10
```

5. import berfungsi untuk memanggil fungsi dari kelas lain.

```
1 # 'Common base class for all employees'
2 # empCount = 0
```

6. pemakaian paket fungsi apabila ke library ada di dalam folder

```
1 from fungsi_fanny import penulisan
```

7. pemakaian paket kelas apabila ke library ada di dalam folder

```
1 #from 3lib_1174069 import penulisan
```

### 3.5 keterampilan pemrograman

1. No 1

```
1 def penulisan(npm):
2     npm = list(str(npm))
3
4     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
5     angka2 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
6     angka3 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
7     angka4 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
8     angka5 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
9     angka6 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
10    angka7 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
11    angka8 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
12    angka9 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
13    angka10 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
14
15    hasil1 = []
16    hasil2 = []
17    hasil3 = []
```

```

18     hasil4 = []
19     hasil5 = []
20     hasil6 = []
21     hasil7 = []
22     hasil8 = []
23     hasil9 = []
24     hasil10 = []
25
26     for x in npm:
27         hasil1.append(angka1[x])
28         hasil2.append(angka2[x])
29         hasil3.append(angka3[x])
30         hasil4.append(angka4[x])
31         hasil5.append(angka5[x])
32         hasil6.append(angka6[x])
33         hasil7.append(angka7[x])
34         hasil8.append(angka8[x])
35         hasil9.append(angka9[x])
36         hasil10.append(angka10[x])
37
38     print(*hasil1, sep=' ')
39     print(*hasil2, sep=' ')
40     print(*hasil3, sep=' ')
41     print(*hasil4, sep=' ')
42     print(*hasil5, sep=' ')
43     print(*hasil6, sep=' ')
44     print(*hasil7, sep=' ')
45     print(*hasil8, sep=' ')
46     print(*hasil9, sep=' ')
47     print(*hasil10, sep=' ')
48
49     penulisan(int(input("Masukan NPM nyaa :")))

```

## 2. N0 2

```

1 def perulangan(npm):
2     hitung = 0
3     while(hitung < 69):
4         print("Halo, "+str(npm)+" anyeong?")
5         hitung = hitung +1

```

## 3. N0 3

```

1 def perulangan(npm):
2     hitung = 0
3     npm = str(npm)
4     bil = npm[4:7]
5
6     while(hitung < 9):
7         print("Halo, "+bil+" watsapp?")
8         hitung = hitung +1
9
10    perulangan(int(input("Masukan NPM nyaa : ")))

```

## 4. N0 4

```

1 def perulangan2(npm):
2     npm = str(npm)
3     bil = npm[-3]
4     print("Halo, "+bil+" wattsup?")
5
6 perulangan2(int(input("Masukan NPM nyaa : ")))

```

## 5. N0 5

```

1 def down(npm):
2     for i in npm:
3         print (i)

```

## 6. N0 6

```

1 def penjumlahan(npm):
2     jumlah = 0
3     for i in npm:
4         jumlah += int(i)
5     print(str(jumlah)+" hasil penjumlahannya adalah ")

```

## 7. N0 7

```

1 def perkalian(npm):
2     jumlah = 0
3     for i in npm:
4         jumlah *= int(i)
5     print(str(jumlah)+" hasil perkaliannya adalah ")
6
7 perkalian(input("Masukan NPM nyaa : "))

```

## 8. N0 8

```

1 def genap():
2     npm = [1,1,7,4,0,6,9]
3     for i in npm:
4         if (i % 2) == 0:
5             print("Bilangan Genapnya : "+str(i))
6 genap()

```

## 9. N0 9

```

1 def ganjil():
2     npm = [1,1,7,4,0,6,9]
3     for i in npm:
4         if (i%2)==1:
5             print("Bilangan Ganjilnya : "+str(i))
6 ganjil()

```

## 10. N0 10

```

1 def prima(npm):
2     npm = str(npm)
3     bil = npm[2]
4     num = int(bil)

```

```

5     if num > 1:
6         for i in range(2,num):
7             if (num%i)==0:
8                 print("Bukan Bilangan Prima")
9                 break
10            else:
11                print("Bilangan Primanya :"+str(num))
12    else:
13        print("Tidak Ada Bilangan Prima")
14    prima(int(input("Masukan NPM nyaa : ")))

```

## 11. NO 11

```

1 from fungsi_fanny import penulisan

```

## 12. NO 12

```

1 #from 3lib_1174069 import penulisan

```

## 3.6 penanganan Error

```

1 #def penanganan_error(a,b):
2 #     try :
3 #         c = a+b
4 #         print(c)
5 #     except TypeError:
6 #         print("We Are Different")

```

## 3.7 1174054 | Aulyardha Anindita

## 3.8 Pemahaman Teori

### 3.8.1 Fungsi

Fungsi adalah bagian dari suatu sub program yang terdiri dari nama fungsi itu sendiri dan variabel yang dapat digunakan ulang dan nama tersebut dapat dipanggil dimana-pun dalam suatu program. Fungsi dalam python menggunakan kata kunci 'def'. Dan setelah 'def' biasanya terdapat nama pengenalan fungsi yang diikuti oleh parameter yang diapit oleh tanda kurung dan diakhiri dengan tanda titik dua (:). Dan baris berikutnya adalah blok fungsi yang akan dijalankan jika fungsi dipanggil.

Contoh fungsi :

```

1 # Contoh fungsi
2 def dita():
3     print("Assalamualaikum, dita!")
4
5 dita()

```

Inputan fungsi adalah memanggil fungsi dengan fungsi yang telah dibuat dari inputan user dan mengembalikannya dalam bentuk string.

Contoh :

```

1 # Contoh inputan fungsi
2 def test(nama):
3     print ("Nama ku adalah : "+str(nama))
4
5 test(input("Masukkan Nama Kamu :"))

```

Kembalian fungsi adalah keluar dari suatu fungsi dan kembali ke baris selanjutnya dimana suatu fungsi dipanggil.

Contoh :

```

1 #Contoh kembalian fungsi
2 def test_return(a,b):
3     r = a + b
4     return r
5
6 a = 15
7 b = 25
8 c = test_return(a,b)
9 print(c)

```

### 3.8.2 Paket

Paket adalah suatu teknik pengumpulan dari beberapa file-file modul. Paket memudahkan programmer dalam mengelompokkan dan mengorganisasikan modul yang telah dibuat. Ringkasnya, kita tidak perlu membuat script untuk beberapa kasus, namun kita bisa mengelompokkannya dalam 1 file tiap kasus dan memanggilnya dalam satu program.

Cara memanggil paket adalah menggunakan kata kunci 'import' untuk mengimport file yang telah dibuat.

Contoh paket :

```

1 # Contoh 2 paket
2 #from fungsi_dita import *
3 #print(penulisan(int(input("Masukkan NPM anda : "))))

```

### 3.8.3 Kelas, Objek, Atribut, dan Method

Kelas adalah suatu entitas atau struktur data yang biasa digunakan yang terdiri dari objek, atribut, dan method didalamnya. Dengan kata lain kelas adalah sebuah cetak biru atau blueprint dari sebuah objek (instans)

Objek adalah suatu entitas yang biasanya memiliki variabel dan method didalamnya dengan kata lain memiliki keadaan (state) dan kelakuan (behavior).

Atribut adalah suatu entitas atau berupa fungsi-fungsi yang dimiliki oleh kelas atau objek. Biasanya atribut berisi variabel-variabel yang telah dideklarasikan.

Method adalah suatu fungsi yang melekat pada sebuah objek atau instan kelas untuk merepresentasikan suatu behavior (kelakuan).

Contoh :

```

1 # Contoh 3 Kelas
2 #class Employee:

```

```

3 # 'Common base class for all employees'
4 #   empCount = 0
5
6 #   def __init__(self, name, salary):
7 #       self.name = name
8 #       self.salary = salary
9 #       Employee.empCount += 1
10
11 #   def displayCount(self):
12 #       print ("Total Employee %d" % Employee.empCount)
13
14 #   def displayEmployee(self):
15 #       print ("Name : ", self.name, " , Salary : ", self.salary)
16
17
18 #This would create first object of Employee class"
19 #emp1 = Employee("Zara", 2000)
20 #This would create second object of Employee class"
21 #emp2 = Employee("Manni", 5000)
22 #emp1.displayEmployee()
23 #emp2.displayEmployee()
24 #print ("Total Employee %d" % Employee.empCount)

```

### 3.8.4 Cara Pemanggilan Library Kelas dari Instansiansi

Instansiansi adalah suatu pembuatan instance atau objek dari suatu kelas. Untuk memanggil nama kelas yaitu dengan menggunakan fungsi `init()` pada saat kita mendefinisikannya.

Contoh :

```

1 # Contoh pemanggilan library kelas dari instansiansi
2 #import tugas
3 #a = 25
4 #b = 30
5
6 #c = tugas.penambahan(a,b)
7 #print(c)

```

### 3.8.5 Pemakaian Paket dengan Perintah `from` dan `import`

```

1 # Contoh 2 paket
2 #from fungsi_dita import *
3 #print(penulisan(int(input("Masukkan NPM anda : "))))

```

### 3.8.6 Pemakaian Paket Fungsi

```

1 from kelas3lib_1174054 import penulisan

```

### 3.8.7 Pemakaian Paket Kelas

```

1 #from 3lib_1174054 import penulisan

```



### 3.9 Keterampilan Pemrograman

#### 1. Jawaban Soal No.1

```

1 #No 1
2 def penulisan(npm):
3     npm = list(str(npm))
4
5     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
6     angka2 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
7     angka3 = {"0": "##   ## ", "1": "### ", "2": "### ", "3": "##### ", "4": "###   ### ", "5": "###   ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
8     angka4 = {"0": "##   ## ", "1": "### ", "2": "### ", "3": "##### ", "4": "###   ### ", "5": "###   ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
9     angka5 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
10    angka6 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
11    angka7 = {"0": "##   ## ", "1": "### ", "2": "### ", "3": "##### ", "4": "##### ", "5": "###   ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
12    angka8 = {"0": "##   ## ", "1": "### ", "2": "### ", "3": "##### ", "4": "##### ", "5": "###   ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
13    angka9 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
14    angka10 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
15
16    hasil1 = []
17    hasil2 = []
18    hasil3 = []
19    hasil4 = []
20    hasil5 = []
21    hasil6 = []
22    hasil7 = []
23    hasil8 = []
24    hasil9 = []
25    hasil10 = []
26
27    for x in npm:
28        hasil1.append(angka1[x])
29        hasil2.append(angka2[x])
30        hasil3.append(angka3[x])
31        hasil4.append(angka4[x])
32        hasil5.append(angka5[x])

```

```

33     hasil6.append(angka6[x])
34     hasil7.append(angka7[x])
35     hasil8.append(angka8[x])
36     hasil9.append(angka9[x])
37     hasil10.append(angka10[x])
38
39     print(*hasil1, sep=' ')
40     print(*hasil2, sep=' ')
41     print(*hasil3, sep=' ')
42     print(*hasil4, sep=' ')
43     print(*hasil5, sep=' ')
44     print(*hasil6, sep=' ')
45     print(*hasil7, sep=' ')
46     print(*hasil8, sep=' ')
47     print(*hasil9, sep=' ')
48     print(*hasil10, sep=' ')
49
50     penulisan(int(input("Masukkan NPM Anda :")))

```

## 2. Jawaban Soal No.2

```

1  #No 2
2  def perulangan(npm):
3      hitung = 0
4      while(hitung < 54):
5          print("Halo, "+str(npm)+" apa kabar?")
6          hitung = hitung +1
7
8  perulangan(int(input("Masukan NPM Anda : ")))

```

## 3. Jawaban Soal No.3

```

1  #No 3
2  def perulangan_3_digit(npm):
3      hitung = 0
4      npm = str(npm)
5      bil = npm[4:7]
6
7      while(hitung < 9):
8          print("Halo, "+bil+" apa kabar?")
9          hitung = hitung +1
10
11  perulangan_3_digit(int(input("Masukan NPM Anda : ")))

```

## 4. Jawaban Soal No.4

```

1  #No 4
2  def perulangan_3_digit_terakhir(npm):
3      npm = str(npm)
4      bil = npm[-3]
5      print("Halo, "+bil+" apa kabar?")
6
7  perulangan_3_digit_terakhir(int(input("Masukan NPM Anda : ")))

```

## 5. Jawaban Soal No.5

```

1 #No 5
2 def down(npm):
3     for i in npm:
4         print (i)
5
6 down(input("Masukan NPM Anda : "))

```

## 6. Jawaban Soal No.6

```

1 #No 6
2 def penjumlahan(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah += int(i)
6     print(str(jumlah)+" merupakan hasil dari penjumlahan")
7
8 penjumlahan(input("Masukkan NPM Anda : "))

```

## 7. Jawaban Soal No.7

```

1 #No 7
2 def perkalian(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah *= int(i)
6     print(str(jumlah)+" merupakan hasil dari perkalian")
7
8 perkalian(input("Masukan NPM Anda : "))

```

## 8. Jawaban Soal No.8

```

1 #No 8
2 def genap():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i % 2) == 0:
6             print("Bilangan Genapnya adalah : "+str(i))
7
8 genap()

```

## 9. Jawaban Soal No.9

```

1 #No 9
2 def ganjil():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i%2)==1:
6             print("Bilangan Ganjilnya adalah : "+str(i))
7
8 ganjil()

```

## 10. Jawaban Soal No.10

```

1 #No 10
2 def prima(npm):
3     npm = str(npm)
4     bil = npm[2]

```

```

5     num = int(bil)
6     if num > 1:
7         for i in range(2,num):
8             if (num%i)==0:
9                 print("Bukan Bilangan Prima")
10                break
11            else:
12                print("Bilangan Prima :"+str(num))
13        else:
14            print("Tidak Ada Bilangan Prima")
15    prima(int(input("Masukan NPM Anda : ")))

```

### 11. Jawaban Soal No.11

```

1 from kelas3lib_1174054 import penulisan

```

### 12. Jawaban Soal No.12

```

1 #from 3lib_1174054 import penulisan

```

## 3.10 Penanganan Error

```

1 # Contoh inputan fungsi
2 def test(nama):
3     print ("Nama ku adalah : "+str(nama))
4
5 test(input("Masukkan Nama Kamu :"))

```

### 3.11 Dini Permata Putri

1. Apa itu fungsi, inputan fungsi dan kembalian fungsi dengan contoh kode program lainnya.

fungsi / function adalah satu blok kode yang melakukan tugas tertentu atau satu blok kode yang melakukan tugas tertentu atau satu blok instruksi yang eksekusi ketika dipanggil dari bagian lain dalam sebuah program. tujuan pembuatan fungsi adalah : memudahkan dalam pembuatan program.

contoh kodenya :

```

def function_name(parameters):
    """function_docstring"""
    statement(s)
    return [expression]

```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya.

paket digunakan untuk mengelompokkan kelas-kelas yang mempunyai kemiripan fungsi (related class). kelas-kelas java yang akan digunakan didalam program, terlebih dahulu harus diimpor beserta dengan nama paket dimana kelas tersebut berada. cara memanggilnya :

```
penampung = namaClass()
penampung.namaMetode()
```

3. Jelaskan apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing.

- class adalah salah satu cara bagaimana kita membuaat sebuah kode yang mempunyai behaviour tertentu dan lebih mudah dalam mengprganisasi berbagai fungsi dan state-nya. dalam sebuah class kamu dapat menyimpan sebuah state tanpa harus membuat banyak state bila tidak menggunakan class.

contohnya :

Class Product:

```
vendor.message = "Ini adalah rahasia"
name = ""
price = ""
size = ""
unit = ""
```

```
def init_(self, name):
print "ini adalah consuctor"
self.name = name
self.unit = "ml"
self.size = 250
```

```
def get.vendor.message(self):
print self.vendor.message
```

```
def set_price(Self, price):
self.price = price
```

- objek adalah instansi atau perwujudan dari sebuah kelas. bila kelas adalah prototypenya, dan objek adalah barang jadinya.

contohnya :

```
obj=Karyawan("K001", "Dini", "Teknisi")
obj.infoKaryawan()
```

```
#tambah karyawan baru
obj2=Karyawan("K002", "Ayu", "Akunting")
obj2.infoKaryawan()
#tampilkan total karyawan
print "Total Karyawan "
```

- atribut adalah instance spesifik untuk setiap objek, atribut class sama untuk semua contoh – yang dalam hal ini adalah semua dog.

contohnya :

```
class Dog:
#class Attribute
```

```
species = 'mammal'
```

```
#Initializer / Instance Attributes
```

```
def.init(self, name, age):
```

```
self.name = name
```

```
self.age = age
```

jadi, sementara setiap Dog memiliki nama dan umur yang unik, setiap Dog akan menjadi mamalia.

Method digunakan untuk melakukan operasi dengan atribut objek, seperti init metodenya, argumen pertama selalu self:

contohnya :

```
class dog:
```

```
# instance method def description(self):
```

```
return "() is () years old".format(self.name, self.age)
```

```
# instance method
```

```
def speak(self, sound):
```

```
return "() says {}".format(self.name, sound)
```

```
#instantiate the Dog object
```

```
mikey = Dog("Mikey", 6)
```

```
#call our instance methods
```

```
print(mikey.description())
```

```
print(mikey.speak("Gruff Gruff"))
```

4. Jelaskan cara pemanggilan library kelas dari instansiasi dan pemakaiannya dengan contoh program lainnya.

library adalah salah satu cara yang paling efisien untuk menghemat waktu ketika membangun aplikasi. untuk memanggil file library digunakan perintah include() atau require(), selain include() dan require() ada juga include.once() dan require.once(), perbedaannya adalah jika suatu include ke suatu file dilakukan selama lebih dari 1 kali dalam suatu file, maka akan menghasilkan error karena dianggapnya ada deklarasi ulang (redeclare), tetapi jika menggunakan include.once() atau require.once() maka kejadian tersebut dihindari.

contohnya :

```
$autoload
```

```
'libraries'] = array('form_validation','database','session'); $thisloadlibrary('nama.li
```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import penambahan disertai dengan contoh kode lainnya.

paket dengan perintah from kalkulator import penambahan pertama yaitu tentukan nama fungsi, variabel, dan inputannya apa saja, setiap penulisan harus menggunakan () dan : dan identasi (jaraknya harus sama)

contoh :

```
def penambahan (a+b):
```

```
r=(a+b)
```

```
return
```

a=5

b=6

anu=penambahan(a,b)

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila file library ada di dalam folder.

```
def tambah(bil1,bil2):
```

```
total = bil1+bil2
```

```
return total
```

```
def kurang(bil1,bil2):
```

```
total = bil1-bil2
```

```
return total
```

```
def kali(bil1,bil2):
```

```
total = bil1*bil2
```

```
return total
```

```
def bagi(bil1,bil2):
```

```
total = bil1/bil2
```

```
return total
```

```
def nilai(n1,n2):
```

```
hasil = n1+n2
```

```
if 80 <= hasil <= 100:
```

```
print("nilai anda adalah A")
```

```
elif 70 <= hasil <= 80:
```

```
print("nilai anda adalah B")
```

```
elif 60 <= hasil <= 70:
```

```
print("nilai anda adalah C")
```

```
elif 50 <= hasil <= 60:
```

```
print("nilai anda adalah D, Silahkan Mengulang kembali")
```

```
else: print("anda GAGAL")
```

7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila file library ada di dalam folder

mendefinisikan sebuah class dengan menggunakan kata kunci class diikuti oleh nama class tersebut.

```
class ClassName: """class docstring"""
```

```
class.body
```

class memiliki docstring atau string dokumentasi yang bersifat opsional artinya bisa atau tidak. Docstring bisa diakses menggunakan format

```
ClassName.doc
```

### 3.12 ainul filiani

1. Apa itu fungsi, inputan fungsi dan kembalian fungsi dengan contoh kode program lainnya.

fungsi / function adalah satu blok kode yang melakukan tugas tertentu atau satu blok kode yang melakukan tugas tertentu atau satu blok instruksi yang eksekusi ketika dipanggil dari bagian lain dalam sebuah program. tujuan pembuatan fungsi adalah : memudahkan dalam pembuatan program.

contoh kodenya :

```
def function_name(parameters):
    """function_docstring"""
    statement(s)
    return [expression]
```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya.

paket digunakan untuk mengelompokkan kelas-kelas yang mempunyai kemiripan fungsi (related class). kelas-kelas java yang akan digunakan didalam program, terlebih dahulu harus diimpor beserta dengan nama paket dimana kelas tersebut berada. cara memanggilnya :

```
penampung = namaClass()
penampung.namaMetode()
```

3. Jelaskan apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing.

- class adalah salah satu cara bagaimana kita membuat sebuah kode yang mempunyai behaviour tertentu dan lebih mudah dalam mengorganisasi berbagai fungsi dan state-nya. dalam sebuah class kamu dapat menyimpan sebuah state tanpa harus membuat banyak state bila tidak menggunakan class.

contohnya :

Class Product:

```
vendor.message = "Ini adalah rahasia"
name = ""
price = ""
size = ""
unit = ""
```

```
def init_(self, name):
    print "ini adalah constructor"
    self.name = name
    self.unit = "ml"
    self.size = 250
```



```
def get.vendor.message(self):
print self.vendor.message
```

```
def set_price(Self, price):
self.price = price
```

- objek adalah instansi atau perwujudan dari sebuah kelas. bila kelas adalah prototipnya, dan objek adalah barang jadinya.

contohnya :

```
obj=Karyawan("K001", "Dini", "Teknisi")
obj.infoKaryawan()
```

```
#tambah karyawan baru
obj2=Karyawan("K002", "Ayu", "Akunting")
obj2.infoKaryawan()
```

```
#tampilkan total karyawan
print "Total Karyawan "
```

- atribut adalah instance spesifik untuk setiap objek, atribut class sama untuk semua contoh – yang dalam hal ini adalah semua dog.

contohnya :

```
class Dog:
#class Attribute
species = 'mammal'
#Initializer / Instance Attributes
def.init(self, name, age):
self.name = name
self.age = age
```

jadi, sementara setiap Dog memiliki nama dan umur yang unik, setiap Dog akan menjadi mamalia.

Method digunakan untuk melakukan operasi dengan atribut objek, seperti init metodenya, argumen pertama selalu self:

contohnya :

```
class dog:
# instance method def description(self):
return "() is () years old".format(self.name, self.age)
# instance method
def speak(self, sound):
return "() says {}".format(self.name, sound)
#instantiate the Dog object
mikey = Dog("Mikey", 6)
#call our instance methods
print(mikey.description())
print(mikey.speak("Gruff Gruff"))
```

4. Jelaskan cara pemanggilan library kelas dari instansiasi dan pemakaiannya dengan contoh program lainnya.

library adalah salah satu cara yang paling efisien untuk menghemat waktu ketika membangun aplikasi. Untuk memanggil file library digunakan perintah `include()` atau `require()`, selain `include()` dan `require()` ada juga `include.once()` dan `require.once()`, perbedaannya adalah jika suatu `include` ke suatu file dilakukan selama lebih dari 1 kali dalam suatu file, maka akan menghasilkan error karena dianggapnya ada deklarasi ulang (`redeclare`), tetapi jika menggunakan `include.once()` atau `require.once()` maka kejadian tersebut dihindari.

contohnya :

\$autoload

```
'libraries'] = array('form_validation','database','session'); $this->loadlibrary('nama.li
```

5. Jelaskan dengan contoh pemakaian paket dengan perintah `from` kalkulator import penambahan disertai dengan contoh kode lainnya.

paket dengan perintah `from` kalkulator import penambahan pertama yaitu tentukan nama fungsi, variabel, dan inputannya apa saja, setiap penulisan harus menggunakan `()` dan `:` dan identasi (jaraknya harus sama)

contoh :

def penambahan (a+b):

  r=(a+b)

  return

  a=5

  b=6

  anu=penambahan(a,b)

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila file library ada di dalam folder.

def tambah(bil1,bil2):

  total = bil1+bil2

  return total

    def kurang(bil1,bil2):

      total = bil1-bil2

      return total

    def kali(bil1,bil2):

      total = bil1\*bil2

      return total

    def bagi(bil1,bil2):

      total = bil1/bil2

      return total

    def nilai(n1,n2):

      hasil = n1+n2

```

if 80 <= hasil <= 100:
    print("nilai anda adalah A")
elif 70 <= hasil <= 80:
    print("nilai anda adalah B")
elif 60 <= hasil <= 70:
    print("nilai anda adalah C")
elif 50 <= hasil <= 60:
    print("nilai anda adalah D, Silahkan Mengulang kembali")
else: print("anda GAGAL")

```

7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila file library ada di dalam folder mendefinisikan sebuah class dengan menggunakan kata kunci class diikuti oleh nama class tersebut.

```

class ClassName: """class docstring"""
class.body

```

class memiliki docstring atau string dokumentasi yang bersifat opsional artinya bisa atau tidak. Docstring bisa diakses menggunakan format ClassName.doc

### 3.13 Chandra Kirana Poetra

#### 3.13.1 Pemahaman Teori

1. Apa itu fungsi, inputan fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi merupakan suatu fitur yang dapat digunakan untuk membuat suatu program lebih mudah dimengerti dan juga agar program tersusun rapih serta mengurangi redudansi kode yang sama, misalkan kita ingin membuat suatu fungsinya penjumlahan maka, kita membuat fungsi penjumlahan lalu ketika kita ingin menghitung penjumlahan, maka yang perlu kita lakukan tinggal memanggil fungsinya itu sendiri agar fungsi itu lah yang akan melakukan eksekusi perhitungan. fungsi pada python biasanya dimulai dengan mengetikan kata kunci def yang selanjutnya diikuti dengan nama fungsinya itu sendiri contohnya seperti dibawah ini :

```

1 def masukannamafungsidisini(inputannyadisini):

```

Inputan fungsi sendiri merupakan suatu input/masukan data yang kita berikan ke program

```

1 outputnya = masukannamafungsidisini("Kembalian Fungsi")

```

Pengembalian fungsi atau return merupakan suatu fitur yang digunakan untuk mengembalikan nilai yang telah di proses oleh fungsi, penulisan return sendiri yaitu return diikuti dengan variabel mana yang nantinya ingin nilainya dikembalikan

```
1 return inputannyadisini
```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Library atau paket sendiri merupakan suatu kumpulan dari fungsi fungsi yang ditulis oleh berbagai macam developer yang biasanya digunakan untuk menghemat waktu, dikarenakan misalkan untuk fungsi koneksi, biasanya orang sudah menyediakan librarynya, oleh karena itu dikarenakan sudah ada yang membuat, maka akan lebih cepat jika kita menggunakan library yang dibuat oleh orang lain dibandingkan dengan kita harus membuat lagi dari awal.

```
1 #soal no 2
2 import math
3 print("Nilai I adalah : ", math.pi)
```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. Kelas merupakan blueprint atau template dari suatu objek yang nantinya akan di munculkan menjadi instance objek objek merupakan sebuah instance dari suatu kelas, kelas merupakan rancangan yang mendeskripsikan sesuatu hal didunia nyata secara umum misalkan kita punya class mobil, maka kita bisa membuat objek mobil sedan, mobil sport, mobil offroad dan lain lain seperti itulah contoh class dan objek, atribut adalah sesuatu hal yang melekat pada suatu objek atau kelas sedangkan method sendiri merupakan suatu fungsi yang ada pada suatu kelas yang cara penggunaanya dengan dipanggil

```
1 #soal no 3
2 class Mahasiswa:
3     jumlahMahasiswa = 0
4
5     def __init__(self, npm,nama):
6         self.npm = npm
7         self.nama = nama
8         Mahasiswa.jumlahMahasiswa +=1
9
10    def tampilprofile(self):
11        print("npm : ", self.npm)
12        print("NIK : ", self.nama)
13        print()
14
15    mahasiswa1 = Mahasiswa("1174079","Chandra Kirana Poetra")
16    mahasiswa2 = Mahasiswa("1174069","Bakti qilan")
17
18    mahasiswa1.tampilprofile()
19    mahasiswa2.tampilprofile()
20
21    print("total mahasiswa adalah", Mahasiswa.jumlahMahasiswa)
```

4. Jelaskan cara pemanggilan library kelas dari instansiasi dan pemakaianya dengan contoh program lainnya. cara pemanggilannya yaitu dengan import terdahulu filenya, lalu buat variabel untuk menampung data dan setelah itu panggil nama class dan panggil method yang ada di class itu.

```
1 #soal no 4
2
3 from file.Mahasiswa import Mahasiswa
4
5 mahasiswa1 = Mahasiswa("1174079","Chandra Kirana Poetra")
6 mahasiswa2 = Mahasiswa("1174069","Bakti qilan")
7 mahasiswa1.tampilprofile()
8 mahasiswa2.tampilprofile()
9 print("total mahasiswa adalah", Mahasiswa.jumlahMahasiswa)
```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. contohnya sebagai berikut :

```
1 #soal no 5
2 from file import Kalkulator
3
4 hasil = Kalkulator.penambahan(7,6)
5 print(hasil)
```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila file library ada di dalam folder. pemakaian paket sendiri adalah kumpulan fungsi fungsi contoh dari kodenya sebagai berikut :

```
1 #soal no 6
2 from file import Kalkulator
3
4 a = 120
5 b = 60
6
7 hasil1 = Kalkulator.penambahan(a,b)
8 hasil2 = Kalkulator.pengurangan(a,b)
9 hasil3 = Kalkulator.perkalian(a,b)
10 hasil4 = Kalkulator.pembagian(a,b)
11
12 print(hasil1)
13 print(hasil2)
14 print(hasil3)
15 print(hasil4)
```

7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila file library ada di dalam folder. pemakaian paket sendiri adalah kumpulan fungsi fungsi contoh dari kodenya sebagai berikut :

```
1 #soal no 7
2
3 from file.Mahasiswa import Mahasiswa
4
5 mahasiswa1 = Mahasiswa("1174079","Chandra Kirana Poetra")
6 mahasiswa2 = Mahasiswa("1174069","Bakti qilan")
7 mahasiswa1.tampilprofile()
8 mahasiswa2.tampilprofile()
9 print("total mahasiswa adalah", Mahasiswa.jumlahMahasiswa)
10
11 print(117407%9)
```

### 3.13.2 Praktek

#### 1. Jawaban soal no 1

```

1 #soal No.1
2 def AnswerNo1():
3
4     npm = input("Masukan NPM Anda:")
5     npm = list(str(npm))
6
7     number1 = {"0": "++++++ ", "1": " ++", "2": "++++++ ", "3":
8         "++++++ ", "4": "+++++", "5": "++++++", "6": "++++++",
9         "7": "++++++", "8": "+++++", "9": "+++++"}
10    number2 = {"0": "+++ ++", "1": "+++ ", "2": "++ ++", "3": "
11        +++ ++", "4": " ++ ++", "5": "++ ", "6": "+++ ",
12        "7": "++++++", "8": "++ ++", "9": "++ ++"}
13    number3 = {"0": "+++ ++", "1": "+++", "2": " ++", "3": "
14        ++++", "4": " ++ ++", "5": "++ ", "6": "+++ ",
15        "7": " ++", "8": "+++++", "9": "++ ++"}
16    number4 = {"0": "+++ ++", "1": "+++", "2": " ++", "3": "
17        ++++", "4": "++++++", "5": "++++++", "6": "++++++",
18        "7": " ++", "8": "++++++", "9": "++++++"}
19    number5 = {"0": "+++ ++", "1": "+++", "2": " +++", "3": "
20        +++ ++", "4": " ++", "5": " ++", "6": "+++ ++",
21        "7": " ++", "8": "++ ++", "9": " ++"}
22    number6 = {"0": "++++++ ", "1": "+++", "2": "++++++ ", "3": "
23        ++++", "4": " ++", "5": "++++++", "6": "++++++ ",
24        "7": "+++ ", "8": "+++++", "9": "++++++"}
25
26    Result1 = []
27    Result2 = []
28    Result3 = []
29    Result4 = []
30    Result5 = []
31    Result6 = []
32
33    for x in npm:
34        Result1.append(number1[x])
35        Result2.append(number2[x])
36        Result3.append(number3[x])
37        Result4.append(number4[x])
38        Result5.append(number5[x])
39        Result6.append(number6[x])
40
41    print(*Result1, sep=' ')
42    print(*Result2, sep=' ')
43    print(*Result3, sep=' ')
44    print(*Result4, sep=' ')
45    print(*Result5, sep=' ')
46    print(*Result6, sep=' ')
47
48    AnswerNo1()

```

#### 2. Jawaban soal no 2

```

1 #Soal no.2
2 def perulangan(npm):

```

```

3     hitung = 0
4     while (hitung < 79):
5         print("Halo, " + str(npm) + " Apa Kabar?")
6         hitung = hitung + 1
7     perulangan(int(input("Masukkan NPM: ")))

```

### 3. Jawaban soal no 3

```

1 #Soal no.3
2 def perulangan3digitakhir(npm):
3     hitung = 0
4     npm = str(npm)
5     x = npm[4:7]
6     while (hitung < 16):
7         print("Halo, " + x + " Apa Kabar?")
8         hitung = hitung + 1
9     perulangan3digitakhir(int(input("Masukkan NPM: ")))

```

### 4. Jawaban soal no 4

```

1 #Soal no.4
2 def digit3daribelakang(npm):
3     npm = str(npm)
4     x = npm[-3]
5     print("Halo, " + x + " Apa Kabar?")
6     digit3daribelakang(int(input("Masukkan NPM: ")))

```

### 5. Jawaban soal no 5

```

1 #Soal no5
2 def npmkebawah(npm):
3     for i in npm:
4         print (i)
5
6     npmkebawah(input("Masukkan NPM: "))

```

### 6. Jawaban soal no 6

```

1 #Soal no.6
2 def penjumlahannpm(npm):
3     jumlahkan = 0
4     for i in npm:
5         jumlahkan += int(i)
6     print ("Hasil Penjumlahan NPM adalah : " +str(jumlahkan))
7     penjumlahannpm(input("Masukkan NPM: "))

```

### 7. Jawaban soal no 7

```

1 #Soal No.7
2 def perkaliannpm(npm):
3     kalikan = 0
4     for i in npm:
5         kalikan *= int(i)
6     print ("Hasil Perkalian NPM adalah : " +str(kalikan))
7     perkaliannpm(input("Masukkan NPM: "))

```

## 8. Jawaban soal no 8

```

1 #Soal No.8
2 def printdigitgenap(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if(n % 2 == 0):
6             if(n != 0):
7                 print(n, end = "")
8     printdigitgenap(input("Masukkan NPM Anda bray: "))

```

## 9. Jawaban soal no 9

```

1 #Soal No.9
2 def printdigitganjil(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if(n % 2 != 0):
6             print(n, end = "")
7     printdigitganjil(input("Masukkan NPM Anda bray: "))

```

## 10. Jawaban soal no 10

```

1 #Soal No.10
2 def printdigitprima(npm):
3     npm = list(map(int, npm))
4     prima = []
5     for n in npm:
6         bilPrima = True
7         if n == 0 or n == 1:
8             bilPrima = False
9         for x in range(2, n):
10             if n % x == 0:
11                 bilPrima = False
12         if bilPrima:
13             prima.append(n)
14
15     for p in prima:
16         print(p, end = "")
17     printdigitprima(input("Masukkan NPM: "))

```

## 11. Jawaban soal no 11

```

1
2 contohimport = __import__('3lib')
3 npm = "1174079"
4
5 result1 = contohimport.AnswerNo1(npm)
6 result2 = contohimport.AnswerNo2(npm)
7 result3 = contohimport.AnswerNo3(npm)
8 result4 = contohimport.AnswerNo4(npm)
9 result5 = contohimport.AnswerNo5(npm)
10 result6 = contohimport.AnswerNo6(npm)
11 result7 = contohimport.AnswerNo7(npm)
12 result8 = contohimport.AnswerNo8(npm)
13 result9 = contohimport.AnswerNo9(npm)

```



```

14 result10 = contohimport.AnswerNo10(npm)
15 print()

```

## 12. Jawaban soal no 12

```

1 from kelas3lib import kelas3lib
2
3 npm = "1174079"
4
5 kel3lib = kelas3lib(npm)
6
7 kel3lib.AnswerNo1()
8 kel3lib.AnswerNo2()
9 kel3lib.AnswerNo3()
10 kel3lib.AnswerNo4()
11 kel3lib.AnswerNo5()
12 kel3lib.AnswerNo6()
13 kel3lib.AnswerNo7()
14 kel3lib.AnswerNo8()
15 kel3lib.AnswerNo9()
16 kel3lib.AnswerNo10()

```

### 3.13.3 Praktek

1. peringantan error dan juga penjelasan
2. syntax error terjadi ketika kode yang dijalankan terdapat kesalahan penulisan, penempatan dan lain lain dengan solusi mencari di line yang terdapat masalah
3. zero division error eror ini terjadi ketika eksekusi program yang memiliki rumus matematika menghasilkan 0 dari eksekusi bagi yang memiliki hasil 0, jangan pernah membagi dengan 0

4. name error name error adalah error yang terjadi ketika kesalahan penamaan terdeteksi, solusinya jangan salah ketik

contoh fungsi yang menggunakan try dan except

```

1 def cobainaye (masukinnama):
2     try:
3         print("UY " + str(masukinnama))
4     except:
5         print("ada yang salah bray")
6 cobainaye(input("Nama Mu nak"))

```

# DAFTAR PUSTAKA

---

1. R. Awangga, "Sampeu: Servicing web map tile service over web map service to increase computation performance," in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.



# Index

---

disruptif, xxxi  
modern, xxxi