

# Tugas Pemrograman Chapter 2

Annisa Khairani(1184071)

29 October 2019

## 1 Teori

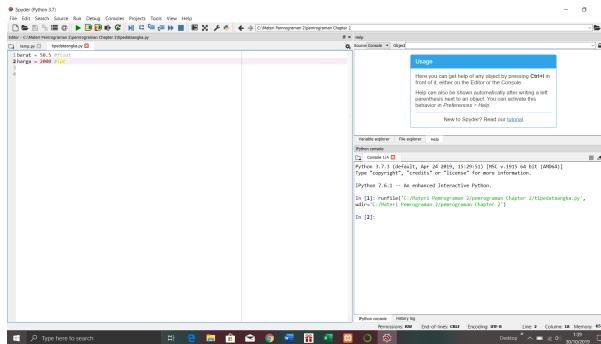
### 1.1 Jenis-Jenis Variabel dan Cara Pemakaian Variabel

Variabel merupakan suatu wadah untuk menyimpan suatu data yang bersifat mutable yang artinya nilainya bisa diubah-ubah atau sementara, sedangkan data yang disimpan didalam variabel disebut dengan tipe data. Tipe Data tersebut terbagi menjadi beberapa macam, yaitu:

#### 1. Tipe Data Angka

Tipe ini terbagi menjadi beberapa jenis lagi, yaitu:

- (a) Integer (Bilangan Bulat), contohnya adalah 1,2,3,dst.
- (b) Float (Bilangan Pecahan), contohnya adalah 1.5,2.1,3.1,dst.



#### 2. Tipe Data Teks

Tipe ini terbagi menjadi dua jenis, yaitu:

- (a) String (Kumpulan Karakter), contohnya adalah "Saya sedang Tidur"
- (b) Varchar (Karakter), contohnya adalah a,b,c,d,dst.

Penulisan tipe ini harus diapit oleh tanda petik.

The screenshot shows the Spyder Python 3.7 IDE interface. A help dialog box titled 'Usage' is open, providing information about the 'print' function. The code editor window contains the following code:

```

name = "Selina Ghazali Febrianti"
age = 19
print("Name : ", name)
print("Age : ", age)

```

The console output window shows the results of the print statements:

```

In [1]: Name : Selina Ghazali Febrianti
Out[1]: None
In [2]: Age : 19
Out[2]: None

```

### 3. Tipe Data Boolean

Tipe data ini adalah tipe yang hanya memiliki dua nilai saja yaitu True dan False atau 0 dan 1.

The screenshot shows the Spyder Python 3.7 IDE interface. A help dialog box titled 'Usage' is open, providing information about the 'bool' function. The code editor window contains the following code:

```

bool()

```

The console output window shows the result of the bool function:

```

In [1]: bool()
Out[1]: False

```

Adapun contoh penggunaan variabel menggunakan kode python adalah:

The screenshot shows the Spyder Python 3.7 IDE interface. A help dialog box titled 'Usage' is open, providing information about the 'variable' function. The code editor window contains the following code:

```

name = "Selina Ghazali Febrianti"
age = 19
isMale = True
isFemale = False

print("Name : ", name)
print("Age : ", age)
print("Is Male : ", isMale)
print("Is Female : ", isFemale)
print("Is Single : ", single)

```

The console output window shows the results of the print statements:

```

In [1]: Name : Selina Ghazali Febrianti
Out[1]: None
In [2]: Age : 19
Out[2]: None
In [3]: Is Male : True
Out[3]: None
In [4]: Is Female : False
Out[4]: None
In [5]: Is Single : False
Out[5]: None

```

Figure 1: Caption

```

# biodata mahasiswa
nama = "Annisa_Khairani_Febrianti"
alamat = 'Jambi'
umur = 19
tinggi = 160

print("Nama : ", nama)
print("Alamat : ", alamat)
print("Umur : ", umur)
print("Tinggi : ", tinggi)

```

## 1.2 Kode Untuk Meminta Input Dari user dan Melakukan Output ke Layar

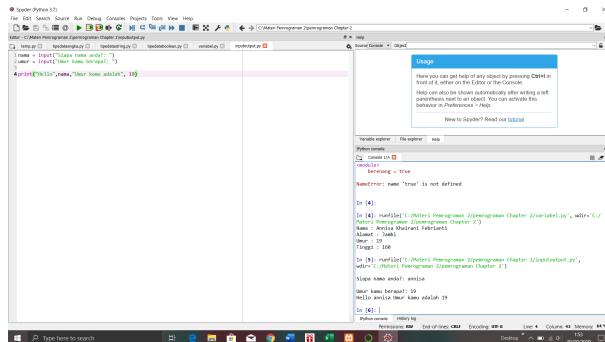
Input adalah suatu masukan yang akan kita berikan ke suatu program sedangkan hasil yang ditampilkan disebut dengan Output.

### 1. Cara mengambil input

Python menyediakan suatu fungsi input, dengan menggunakan kode input()

### 2. Cara menampilkan output

Untuk menampilkan sebuah output teks kita menggunakan kode print()



```

nama = input(" Siapa nama anda ?: ")
umur = input(" Umur kamu berapa ?: ")

```

```

print(" Hello ", nama, " Umur kamu adalah ", 19)

```

### 1.3 Operator Dasar Aritmatika, Tambah, Kali, Kurang, Bagi, dan bagaimana mengubah String ke Integer dan Integer ke String

Operator adalah suatu simbol yang digunakan untuk operasi tertentu. Dan operator aritmatika tersebut termasuk pada operator yang paling sering kita gunakan.

| Operator    | Simbol |
|-------------|--------|
| Penjumlahan | +      |
| Pengurangan | -      |
| Perkalian   | *      |
| Pembagian   | /      |
| Sisa Bagi   | %      |
| Pemangkatan | **     |

Adapun contoh penggunaanya adalah:

```
# Operator Python 3.7
# Author: Mulyadi
# Date: 2023-01-10
# Description: Demonstrasi operator dasar

a = int(input("Masukkan nilai a: "))
b = int(input("Masukkan nilai b: "))

# Penjumlahan
print("a + b = ", a+b)
print("a + b = ", a + b) # a + b = 30

# Pengurangan
print("a - b = ", a-b)
print("a - b = ", a - b) # a - b = 10

# Perkalian
print("a * b = ", a*b)
print("a * b = ", a * b) # a * b = 300

# Pembagian
print("a / b = ", a/b)
print("a / b = ", a / b) # a / b = 3.0

# Sisa Bagi
print("a % b = ", a%b)
print("a % b = ", a % b) # a % b = 0

# Pemangkatan
print("a ** b = ", a**b)
print("a ** b = ", a ** b) # a ** b = 27000000000000000.0
```

### 1.4 Penjelasan Syntax Untuk Perulangan dan Jenis-Jenisnya serta Contoh Kode dan Cara Menggunakannya di Python

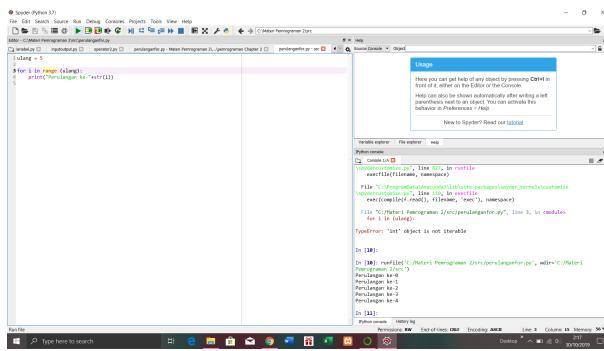
Perulangan berfungsi untuk melakukan sesuatu secara berulang atau terus menerus. Pada bahasa pemrograman terdapat dua jenis perulangan, yaitu For dan While. Perulangan For disebut juga dengan perulangan yang terhitung (counted loop) sedangkan perulangan while disebut juga dengan perulangan yang tak terhitung (uncounted loop).

Secara umum python mengeksekusi program secara berbaris, tetapi untuk perulangan satu baris dieksekusi beberapa kali. Perulangan memerlukan tes kondisi, jika hasil tes true maka blok tersebut akan terus dieksekusi sedangkan

jika false maka akan keluar dari blok perulangan dan mengeksekusi blok selanjutnya.

## 1. Perulangan For

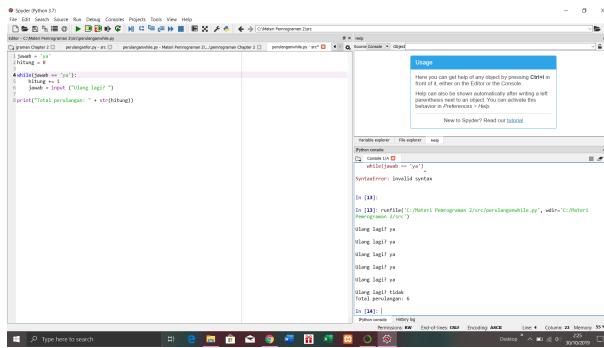
Perulangan ini biasanya digunakan untuk mengetahui sebuah kode yang sudah banyak perulangannya. Adapun contoh kodennya adalah:



```
# Spyder (Python 3.7)
File Edit Search Source Run Debug Console Projects Tools View Help
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganfor.py
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganfor.py, line 1 in <module>
for i in range(ulang):
      ^
SyntaxError: invalid syntax
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganfor.py, line 2 in <module>
    print("Ulang lagi ke-", i+1)
      ^
SyntaxError: invalid syntax
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganfor.py, line 3 in <module>
    i+=1
      ^
SyntaxError: invalid syntax
In [1]:
```

## 2. Perulangan While

Bila kondisi yang diuji salah, maka loop tidak akan pernah dieksekusi



```
# Spyder (Python 3.7)
File Edit Search Source Run Debug Console Projects Tools View Help
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganwhile.py
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganwhile.py, line 1 in <module>
while(jawab == "ya"):
      ^
SyntaxError: invalid syntax
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganwhile.py, line 2 in <module>
    hitung += 1
      ^
SyntaxError: invalid syntax
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganwhile.py, line 3 in <module>
    jawab = input ("Ulang lagi? ")
      ^
SyntaxError: invalid syntax
File C:\Users\Hermawan\OneDrive\Belajar\Python\perulanganwhile.py, line 4 in <module>
    print("Total_perulangan: " + str(hitung))
      ^
SyntaxError: invalid syntax
In [1]:
```

jawab = 'ya'

hitung = 0

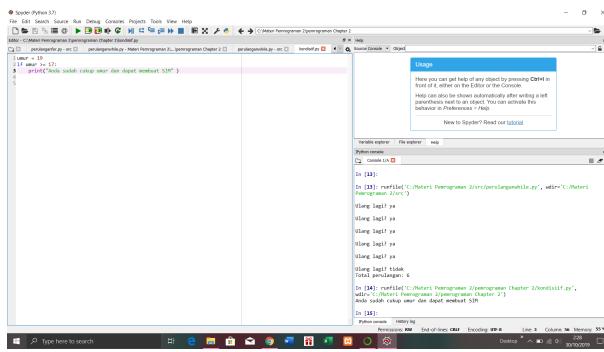
```
while(jawab == 'ya'):
    hitung += 1
    jawab = input ("Ulang lagi? ")
print("Total_perulangan: " + str(hitung))
```

## 1.5 Cara Menggunakan Syntax Untuk Memilih Kondisi dan Contoh Syntax Kondisi Di Dalam Kondisi

Python memiliki tiga jenis kondisional yang dapat digunakan untuk membangun suatu alur logika.adalah if, ifelse, dan ifelifelse.

### 1. Kondisi If

Jika kondisi utama true, maka perintah akan dijalankan

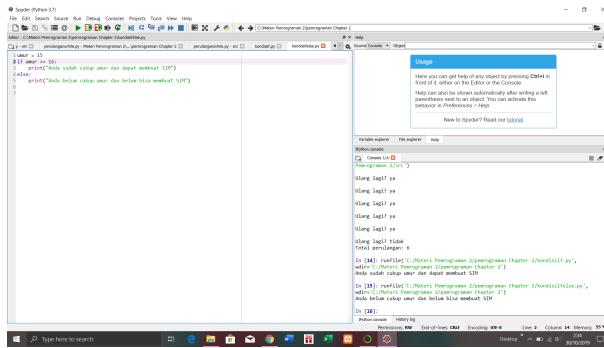


```
Lulus = 10
Lulus <= 12
if Lulus <= 12:
    print("Maaf salah coba coba user dan dapat markah S2N")
5

```

### 2. Kondisi If Else

Untuk memeriksa kondisi utama, else digunakan untuk menangani kondisi selain kondisi yang telah ditentukan.



```
21# user = 56
21# user <= 56
21# print("Maaf salah coba user dan dapat markah S2N")
21# else
21#     print("Maaf salah coba user dan belum bisa mendapat S2N")
5

```

### 3. Kondisi if elif else

Bila anda akan mendefinisikan cukup banyak kondisi, maka gunakan elif di bawah statement if dan diatas statement else.

### 4. If di dalam if (If bersarang)

Suatu kondisional dapat dsimpan di dalam if lain, berikut adalah contohnya:

```

Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:/users/acer/spyder-py3/kondisif.py
Source Console Object
kondisif.py operator.py konversi.py perulanganfor.py kondisif.py
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:/users/acer/spyder-py3/kondisif.py
Source Console Object
kondisif.py operator.py konversi.py perulanganfor.py kondisif.py
Usage
Here you can get help of any object by pressing Ctrl+H in front of it either on the Editor or the Consoles.
Variable explorer File explorer Help
Python console
Console via <ipython>
kondisif.py, wdir='C:/Users/acer/spyder-py3'
Anda sudah cukup umur dan dapat membuktikan KTP
In [21]: runfile('C:/Users/acer/spyder-py3/kondisif.py', wdir='C:/Users/acer/spyder-py3')
Anda belum cukup umur dan belum dapat membuktikan KTP
In [22]: runfile('C:/Users/acer/spyder-py3/kondisif.py', wdir='C:/Users/acer/spyder-py3')
Anda belum cukup umur dan belum dapat membuktikan KTP
In [23]: runfile('C:/Users/acer/spyder-py3/kondisif.py', wdir='C:/Users/acer/spyder-py3')
Masuklah masa paruh bayu
In [24]: Python console History log Line: 1 Column: 10 Memory: 81 %
New file Permissions: RW End-of-lines: CR LF Encoding: UTF-8 Line: 1 Column: 10 Memory: 81 %

Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py
Source Console Object
kondisif.py operator.py konversi.py perulanganfor.py kondisif.py
Usage
Here you can get help of any object by pressing Ctrl+H in front of it either on the Editor or the Consoles.
Variable explorer File explorer Help
Python console
Console via <ipython>
kondisif.py, wdir='C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py'
Anda tidak aktif
In [14]: runfile('C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py', wdir='C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py')
Anda tidak aktif
In [15]: runfile('C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py', wdir='C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py')
Anda tidak aktif
In [16]: runfile('C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py', wdir='C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py')
Anda tidak aktif
In [17]: runfile('C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py', wdir='C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py')
Anda tidak aktif
In [18]: runfile('C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py', wdir='C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py')
Anda tidak aktif
In [19]: runfile('C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py', wdir='C:/Users/acer/Desktop/PythonProgram Chapter 2/kondisif.py')
Anda tidak aktif
In [20]: Python console History log Line: 1 Column: 10 Memory: 81 %

```

## 1.6 Jenis error yang sering ditemui di Python dalam mengerjakan Syntax di atas

Biasanya error terjadi dikarenakan ada kesalahan dalam pengetikan syntax, contohnya sebagai berikut:

Gambar di atas merupakan contoh dari kesalahan syntax titik dua pada line 5.

Gambar di atas merupakan contoh dari kesalahan syntax range, jika nilai bertipe data integer maka harus menggunakan range setelah in.

## 1.7 Cara Memakai Try Except

Try except biasa digunakan untuk menangani error saat penggunaan IO, database, atau pengaksesan indeks suatu list atau dictionary, dll.

```

# coding: utf-8
print("Anda di kota?")
tidak = "tidak"
KMT = "KMT"
if tidak == input():
    print("Maaf anda masuk dalam KMT")
elif KMT == input():
    print("Maaf anda tidak diizinkan")
else:
    print("Maaf anda tidak aktif diluar")
if java:
    print("Maaf anda tidak diluar")
else:
    print("Maaf anda tidak aktif diluar")

```

```

for i in range(5):
    print("Halo")

```

```

try:
    print("Maaf")
except:
    print("Maaf ada yang error")
else:
    print("Maaf ada yang error")

```

## 2 Ketrampilan Pemrograman

1. Luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2. Output:

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

```

*Created on Thu Oct 17 14:28:55 2019*

```

In [1]: print("*****")
print("*****")
print("*****")
print("*****")
print("*****")

```

*@author: USER*  
"""

```

print("*****")
print("*****")
print("*****")
print("*****")
print("*****")

```

2. Program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama **NPM** dan output sebanyak dua digit belakang NPM, contoh NPM : 1184071 maka akan ada output sebanyak 87 dengan tulisan ‘Hallo, 1181071 apa kabar?’

```

In [1]: NPM=int(input("Masukkan NPM:")) # Membuat variabel dan inputan user
Tld=NPM%100 # 
for i in range(Tld): # pengulangan i dalam batas (range) untuk int
    print("Hello ",NPM," How are you?") #output1

```

```

NPM=int(input("Masukkan NPM:")) # Membuat variabel dan inputan user
Tld=NPM%100 # 
for i in range(Tld): # pengulangan i dalam batas (range) untuk int
    print("Hello ",NPM," How are you?") #output1

```

3. Program hello word dengan input nama yang disimpan dalam sebuah variabel string bernama **NPM** dan beri luaran output berupa tiga karakter

belakang dari NPM sebanyak penjumlahan tiga dijat tersebut.

The screenshot shows the Spyder Python IDE interface. In the top-left, there's a toolbar with icons for file operations like Open, Save, Run, Delete, Create, Print, Tools, View, Help, and a search bar. The main window has tabs for 'Editor', 'Materi Pengantar Pemrograman', 'Materi Pengantar Pemrograman', and 'Help'. A code editor window contains the following Python code:

```
X = int(input("1"))
X = int(X)
if X <= 0:
    print("Angka yang dimasukkan kurang dari atau sama dengan nol")
else:
    print("Angka yang dimasukkan lebih dari nol")
```

A tooltip titled 'Usage' is displayed over the code editor, containing the message: "How can you get help of any object by pressing Ctrl+I in IPython? See also on the Editor or the Console. Help can also be shown automatically after writing a left parenthesis. To turn it off, go to Help > Preferences > Help. Behavior in Preferences > Help". Below the tooltip, a link says "New to Spyder? Read our tutorial".

In the bottom-right corner, there's a terminal window with the title 'Terminal' and the command 'ls'. The output shows several files and folders in the current directory:

```
hallo1 11504073.py hallo2 11504073.py hallo3 11504073.py hallo4 11504073.py hallo5 11504073.py hallo6 11504073.py hallo7 11504073.py hallo8 11504073.py hallo9 11504073.py hallo10 11504073.py hallo11 11504073.py hallo12 11504073.py hallo13 11504073.py hallo14 11504073.py hallo15 11504073.py
```

The terminal also shows the command 'cd ..' and the path 'Materi Pengantar 2/11504073'. The status bar at the bottom indicates the number of rows (14), encoding (ASCII), line 1, and memory usage (56).

```
NPM=input ("Masukan Npm anda : ")
```

```
X =int(NPM[4])  
Y =int(NPM[5])  
Z =int(NPM[6])
```

hitung1 = X + Y + Z  
hitung2 = X + Y + Z

```
while hitung1 > 0:
```

hitung1 =

```
... , str(hitung2), " kali( " , str(X) , "+" , str(
```

... , **ser** ( *mwangz* ), **ka**

Bisnow halle wond dat een groot nummer mensen gezien had en zei dat ze de zaak goed hadden gedaan.

- variabel string bernama **NPM** dan beri luaran output berupa digit ketiga dari belakang dari variabel NPM.

```
NPM = input("Npm anda : ")
```

```
print("Halo , " ,NPM[4] , " ,How_Are_You?" )
```

5. (untuk soal no 5 dan selanjutnya wajib menggunakan perulangan dan kondisi) buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

The screenshot shows the Spyder Python 3.7 IDE interface. The code editor contains the following Python script:

```
i=0
NPM = input("Npm_anda: ")
while i<1:
    if len(NPM)<7:
        print("Npm_anda_kurang_dari_7!")
        NPM = input("Npm_anda: ")
    elif len(NPM)>7:
        print("Npm_lebih_dari_7!")
        NPM = input("Npm_anda: ")
    else:
        i=1

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]

for this in A,B,C,D,E,F,G:
    print(this ,end = " ") ,
```

The terminal window shows the execution of the script. It first asks for input, then checks the length of the input. If it's less than 7, it prints a message and asks again. If it's more than 7, it prints a message and asks again. Once the length is exactly 7, it enters the loop where it prints each character of the NPM string followed by a space. The output shows the input '071 Apa kabeh ?' being processed and then printed back as '0 7 1 Apa kabeh ?'.

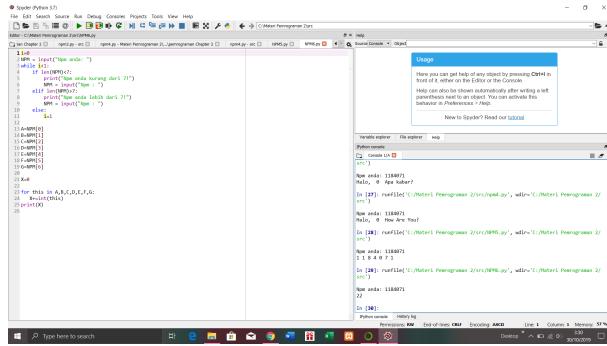
```
i=0
NPM = input("Npm_anda: ")
while i<1:
    if len(NPM)<7:
        print("Npm_anda_kurang_dari_7!")
        NPM = input("Npm_anda: ")
    elif len(NPM)>7:
        print("Npm_lebih_dari_7!")
        NPM = input("Npm_anda: ")
    else:
        i=1

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]

for this in A,B,C,D,E,F,G:
    print(this ,end = " ") ,
```

6. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut

```
i=0
NPM = input("Npm_anda: ")
```



```

while i<1:
    if len(NPM)<7:
        print ("Npm anda kurang dari 7!")
        NPM = input("Npm: ")
    elif len(NPM)>7:
        print ("Npm anda lebih dari 7!")
        NPM = input("Npm: ")
    else:
        i=1

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]

X=0

for this in A,B,C,D,E,F,G:
    X+=int(this)
    print(X)

```

7. Dari soal no 5, Lakukan perkalian dari seluruh variabel tersebut

```

i=0
NPM = input("Masukan npm anda: ")
while i<1:
    if len(NPM)<7:
        print ("Npm anda kurang dari 7!")
        NPM = input("Npm: ")
    elif len(NPM)>7:
        print ("Npm anda lebih dari 7!")

```

```

1 NPM = input("Masukan NPM anda : ")
2 if i>NPM:
3     print("Masukan NPM anda kurang dari 21")
4     NPM = int(NPM)
5 else:
6     print("Masukan NPM anda lebih dari 21")
7     NPM = int(NPM)
8 else:
9     i=1
10
11 A=NPM[0]
12 B=NPM[1]
13 C=NPM[2]
14 D=NPM[3]
15 E=NPM[4]
16 F=NPM[5]
17 G=NPM[6]
18
19 for this in A,B,C,D,E,F,G:
20     X*=int(this)
21 print(X)

```

```

NPM = input("Npm.: ")
else:
    i=1

```

```

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]
X=1

```

```

for this in A,B,C,D,E,F,G:
    X*=int(this)
print(X)

```

8. Dari soal no 5, Lakukan print secara vertikal dari NPM anda menggunakan variabel diatas.

```

1 NPM = input("Masukan NPM anda : ")
2 if i>NPM:
3     print("Masukan NPM anda kurang dari 21")
4     NPM = int(NPM)
5 else:
6     print("Masukan NPM anda lebih dari 21")
7     NPM = int(NPM)
8 else:
9     i=1
10
11 A=NPM[0]
12 B=NPM[1]
13 C=NPM[2]
14 D=NPM[3]
15 E=NPM[4]
16 F=NPM[5]
17 G=NPM[6]
18
19 for this in A,B,C,D,E,F,G:
20     print(this)
21

```

```

i=0
NPM = input("Masukan Npm anda : ")

```

```

while i < 1:
    if len(NPM) < 7:
        print("Npm anda kurang dari 7!")
        NPM = input("Npm : ")
    elif len(NPM) > 7:
        print("Npm anda lebih dari 7!")
        NPM = input("Npm : ")
    else:
        i=1

A=NPM[0]
B=NPM[1]
C=NPM[2]
D=NPM[3]
E=NPM[4]
F=NPM[5]
G=NPM[6]

for this in A,B,C,D,E,F,G:
    print(this)

```

9. Dari soal no 5, Lakukan print NPM anda tapi hanya dijut genap saja.

```
# In [1]: import numpy as np
# In [2]: import matplotlib.pyplot as plt
# In [3]: %matplotlib inline
# In [4]: def f(x,y):
# In [5]:     return np.sin(x)*np.cos(y)
# In [6]: x = np.linspace(0, 2 * np.pi, 100)
# In [7]: y = np.linspace(0, 2 * np.pi, 100)
# In [8]: X,Y = np.meshgrid(x,y)
# In [9]: Z = f(X,Y)
# In [10]: fig = plt.figure()
# In [11]: ax = fig.add_subplot(111, projection='3d')
# In [12]: surf = ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.coolwarm,
# In [13]:                         linewidth=0, antialiased=False)
# In [14]: ax.set_zlim(-1.0, 1.0)
# In [15]: plt.show()
# In [16]: print("Done!")

# In [17]: for this in A,B,C,D,E,F,G:
# In [18]:     if int(this)<20:
# In [19]:         this+=10;
# In [20]:         print(int(this),"-")
```

```
i=0
NPM = input("Masukan npm anda : ")
while i <1:
    if len(NPM) <7:
        print("Npm anda kurang dari 7!")
        NPM = input("Npm : ")
    elif len(NPM) >7:
        print("Npm anda lebih dari 7!")
        NPM = input("Npm : ")
    else:
        i=1
```

```

A=NPM[ 0 ]
B=NPM[ 1 ]
C=NPM[ 2 ]
D=NPM[ 3 ]
E=NPM[ 4 ]
F=NPM[ 5 ]
G=NPM[ 6 ]

X=1

for this in A,B,C,D,E,F,G:

    if int(this)%2==0:
        if int(this)==0:
            this=""
        print(this,end = " ")

```

10. Dari soal no 5, Lakukan print NPM anda tapi hanya dijut ganjil saja.

The screenshot shows the PyCharm IDE interface. On the left, the code editor displays the following Python script:

```

1 i=0
2 print("Masukan Npm anda : ")
3 NPM = input()
4 print("Npm anda kurang dari 7!")
5 elif len(NPM)>7:
6     print("Npm anda lebih dari 7!")
7 else:
8     print(NPM)
9
10 for this in A,B,C,D,E,F,G:
11     if int(this)%2!=0:
12         print(this,end = " ")
13
14 print(" ")
15
16 print(" ")
17
18 print(" ")
19
20 print(" ")
21
22 print(" ")
23
24 print(" ")
25
26 print(" ")
27

```

On the right, the terminal window shows the output of the script. It first asks for input, then prints a message about the length of the input, and finally prints the odd digits of the input. The output is as follows:

```

Masukan Npm anda : 135672
Npm anda kurang dari 7!
Npm anda lebih dari 7!
135672

```

```

i=0
NPM = input("Masukan Npm anda : ")
while i<1:
    if len(NPM)<7:
        print("Npm anda kurang dari 7!")
        NPM = input("Npm : ")
    elif len(NPM)>7:
        print("Npm anda lebih dari 7!")
        NPM = input("Npm : ")
    else:
        i=1

```

```

A=NPM[ 0 ]
B=NPM[ 1 ]

```

C=NPM [ 2 ]  
D=NPM [ 3 ]  
E=NPM [ 4 ]  
F=NPM [ 5 ]  
G=NPM [ 6 ]

X=1

```
for this in A,B,C,D,E,F,G:  
    if int(this)%2==1:  
        print(this, end=" ")
```

11. Dari soal no 5, Lakukan print NPM anda tapi hanya dijut yang termasuk bilangan prima saja.

```

i=0
NPM = input("Masukan Npm anda : ")
while i <1:
    if len(NPM)<7:
        print("Npm anda kurang dari 7!")
        NPM = input("Npm : ")
    elif len(NPM)>7:
        print("Npm anda lebih dari 7!")
        NPM = input("Npm : ")
    else:
        i=1

```

A=NPM[0]  
B=NPM[1]  
C=NPM[2]  
D=NPM[3]  
E=NPM[4]  
F=NPM[5]

G=NPM[ 6 ]

X=1

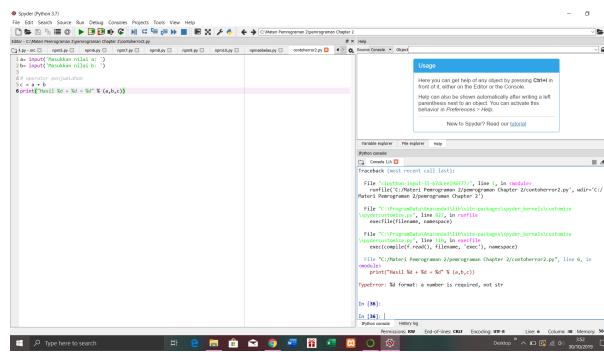
```
for X in A,B,C,D,E,F,G:  
    if int(X) > 1:  
        for i in range (2,int(X)):  
            if (int(X) % i) == 0:  
                break  
            else:  
                print(int(X),end ="" ),
```

### 3 Keterampilan Penanganan Error

Bagian Penanganan error dari script python.

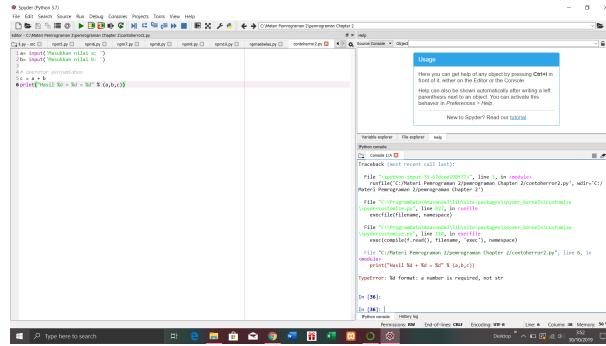
1. Tuliskan peringatan error yang didapat dari mengerjakan praktek kedua ini, dan jelaskan cara penanganan error tersebut.

Peringatan error yang didapat adalah seperti berikut, dijelaskan bahwa penggunaan persen (%) untuk integer dan bukan string. Tetapi tipe data yang dimasukkan atau diinputkan adalah string, maka terjadi error.



```
a= input( 'Masukkan nilai a: ' )  
b= input( 'Masukkan nilai b: ' )  
  
# operator penjumlahan  
c = a + b  
print(" Hasil %d + %d = %d" % ( a , b , c ))
```

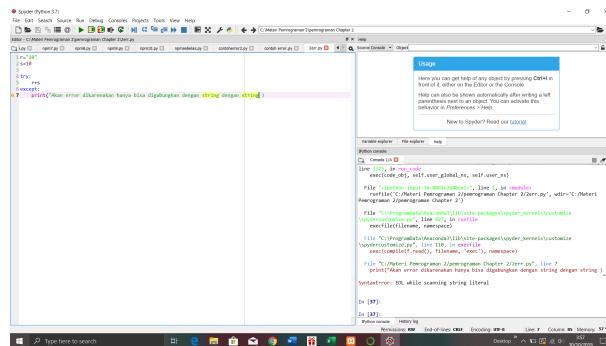
Error akan diatasi oleh try except, dengan cara menangkap error lalu memprint peringatan error tersebut.



```
a= input( 'Masukkan nilai a: ')
b= input( 'Masukkan nilai b: ')

# operator penjumlahan
c = a + b
try:
    print("Hasil %d + %d = %d" % (a,b,c))
except:
    print("Maaf tidak sesuai")
```

2. Membuat file 2err.py dan mengisinya dengan script pengisian variabel sebagai string dan pengisian variabel sebagai interger. Kemudian jumlahkan antara variabel integer dan string dan tangkap jenip errornya, gunakan try except untuk menunjukkan error tersebut dengan bahasa indonesia.



```
r="10"
s=10

try:
    r+s
except:
    print("Akan error dikarenakan hanya bisa digabungkan dengan string dengan")
```