

# Chapter2

Murnia Lestari

1184006

## 1 Teori

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

Variabel merupakan wadah untuk menyimpan beberapa data, sedangkan data yang disimpan di dalam variabel disebut dengan tipe data. Tipe data terbagi menjadi beberapa macam, yaitu:

#### 1. Tipe Data Angka

Tipe ini terbagi menjadi beberapa jenis lagi, yaitu:

- (a) Integer (bilangan bulat), Contoh : 110,112,113,dst.
- (b) Float (bilangan pecahan), contoh : 2.5, 2.6, dst.

#### 2. Tipe Data Teks

Tipe ini terbagi menjadi dua jenis, yaitu:

- (a) String (kumpulan karakter), Contoh : "Saya pergi ke kampus"
- (b) Varchar (karakter), Contoh : a,b,c,dst.

#### 3. Tipe Data Boolean

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

Adapun contoh penggunaan variabel menggunakan kode python adalah:

```
C:\Users\acer> Documents > Chapter2 > JenisVariable.py > ...
1  kuantitasmañana
2  Nama_Penerima: str  'ia Lestari'
3  Nama_Penerima = "Hani Dian"
4  Alamat_Pengirim = "Kelapa Tujuh"
5  No_HP="0895610009160"
6
7  print ("Nama Pengirim :",Nama_Pengirim)
8  print ("Nama Penerima :",Nama_Penerima)
9  print ("Alamat Pengirim :",Alamat_Pengirim)
10 print ("No HP :",No_HP)
```

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

Input merupakan suatu masukan yang akan kita berikan ke program. Sedangkan hasil yang ditampilkan disebut output.

1. Cara mengambil input

Python menyediakan fungsi input, menggunakan kode input()

2. Cara menampilkan output

Untuk menampilkan sebuah output teks kita menggunakan kode print()



```
C:\Users> acer > Documents > Chapter2 > inputoutput.py > ...  
1 input  
2 nama = input("Nama : ")  
3 npm = input("NPM : ")  
4 kelas = input("Kelas: ")  
5 #output  
6 print("Nama",nama,"NPM",npm,"Kelas",kelas)  
7
```

## 1.3 Operator Dasar Aritmatika, Tambah, Kali, Kurang, Bagi, dan Bagaimana Mengubah String ke Integer dan Integer Ke String

Operator merupakan suatu simbol-simbol yang digunakan untuk operasi tertentu. Dan operator aritmatika termasuk pada operator yang paling sering digunakan.

Adapun contoh penggunaannya adalah:

```

C:\Users\acer> Documents\Chapter2> Operatordasar.py > [a]
1  a = int(input("masukkan nilai a :"))
2  b = int(input("masukkan nilai b :"))
3  c = a + b
4  print ("%d + %d = %d" % (a,b,c))
5  c = a * b
6  print ("%d * %d = %d" % (a,b,c))
7  c = a - b
8  print ("%d - %d = %d" % (a,b,c))
9  c = a / b
10 print ("%d / %d = %d" % (a,b,c))

```

Cara mengubah string ke integer dan integer ke string

```

C:\Users\acer> Documents\Chapter2> strint-intstr.py > [a]
1  i=25
2  ii="41"
3  string = str(i)
4  print(string)
5  integer = int(ii)
6  print(integer)

```

## 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 perulangan yang terhitung (counted loop) sedangkan perulangan while disebut perulangan yang tak terhitung (un-counted 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 kode yang sudah banyak perulangannya. Adapun contoh kodenya adalah:

```

C:\Users> acer > Documents > Chapter2 > perulangan.py > ...
1  for
2  ulang = 10
3  for i in range(ulang):
4      print("perulangan ke - "+str(i))
5

```

## 2. Perulangan While

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

```

C:\Users> acer > Documents > Chapter2 > while.py > [ej] jawab
1  jawab="ya"
2  hitung=0
3  while(True):
4      hitung +=1
5      jawab = input("ulangi lagi tidak?")
6      if jawab == "tidak":
7          break
8  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. Yaitu if, ifelse, dan ifelifelse.

### 1. Kondisi If

Jika kondisi utama true, maka perintah akan dijalankan

```

C:\Users> acer > Documents > Chapter2 > logikaif.py > ...
1  #kondisi if adalah kondisi yang akan dieksekusi apabila bernilai
2  nilai=9
3  #jika kondisi benar/true maka program akan dieksekusi
4  if(nilai >7):
5      print("selamat anda lulus")
6  #jika kondisi salah/false maka program tidak akan dieksekusi
7  if(nilai <7):
8      print("selamat anda lulus")

```

### 2. Kondisi If Else

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

```

C:\Users> acer > Documents > Chapter2 > logikaifelse.py > ...
Run Cell | Run Below | Debug cell
1 In[1]:
2
3 nilai = 8
4 if(nilai > 7):
5     print("Selamat anda lulus")
6 else:
7     print("Maaf anda tidak lulus")
8
9 # In[2]:
10
11 def ifelse ():
12     nilai = 3
13     if(nilai > 7):
14         print("Selamat anda lulus")
15     if (nilai >7):
16         print("Maaf anda tidak lulus")
17     ifelse()

```

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

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

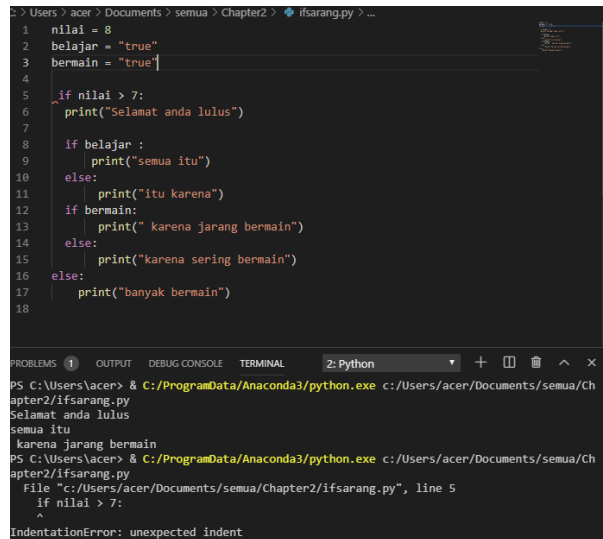
```

C:\Users> acer > Documents > Chapter2 > ifsarang.py > [0] nilai
1 nilai = 8
2 belajar = "true"
3 bermain = "true"
4
5 if nilai: Selamat: str
6     print("Selamat anda lulus")
7
8     if belajar :
9         print("semua itu")
10    else:
11        print("itu karena")
12    if bermain:
13        print(" karena jarang bermain")
14    else:
15        print("karena sering bermain")
16 else:
17     print("banyak bermain")
18

```

## 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:



```
1 nilai = 8
2 belajar = "true"
3 bermain = "true"
4
5 if nilai > 7:
6     print("Selamat anda lulus")
7
8 if belajar :
9     print("semua itu")
10 else:
11     print("itu karena")
12 if bermain:
13     print(" karena jarang bermain")
14 else:
15     print("karena sering bermain")
16 else:
17     print("banyak bermain")
18
```

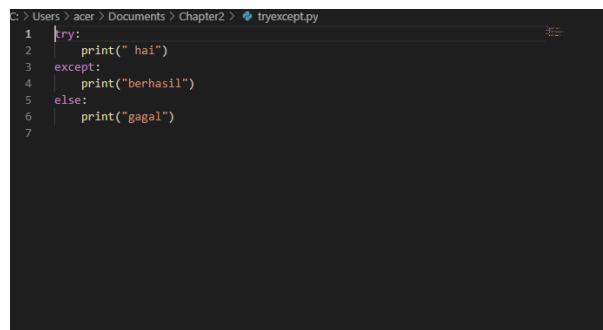
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 2: Python

PS C:\Users\acer> & C:/ProgramData/Anaconda3/python.exe c:/Users/acer/Documents/semua/Chapter2/ifsarang.py  
Selamat anda lulus  
semua itu  
karena jarang bermain  
PS C:\Users\acer> & C:/ProgramData/Anaconda3/python.exe c:/Users/acer/Documents/semua/Chapter2/ifsarang.py  
File "c:/Users/acer/Documents/semua/Chapter2/ifsarang.py", line 5  
if nilai > 7:  
 ^  
IndentationError: unexpected indent

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

## 1.7 Cara Memakai Try Except

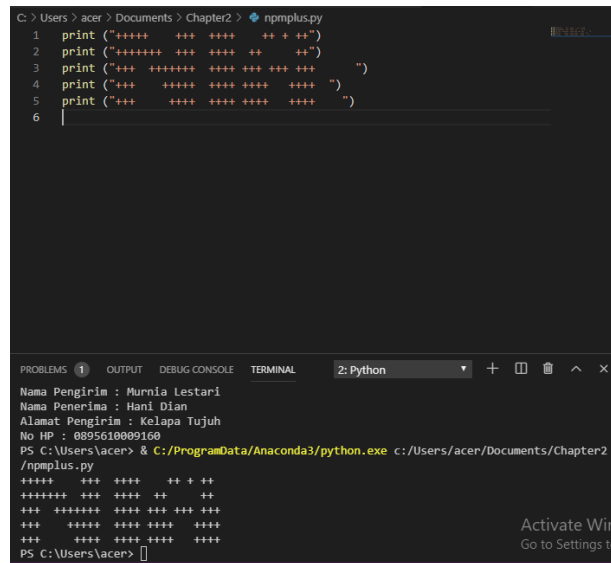
Try except biasa digunakan untuk menangani error saat penggunaan IO, database, atau pengaksesan indeks suatu list atau dictionary, dll.



```
1 try:
2     print(" hai")
3 except:
4     print("berhasil")
5 else:
6     print("gagal")
7
```

## 2 Keterampilan Pemrograman

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



The image shows a code editor window with a Python script named `npmplus.py` and a terminal window below it. The script contains five `print` statements that use combinations of asterisks, plus signs, and hyphens to create a decorative border around a text block. The terminal window shows the output of the script, which is a text block with personal information enclosed in a decorative border.

```
C:\Users\acer> cd Documents\Chapter2 & python npmplus.py
1 print ("+++++  +++  +++  ++ + ++")
2 print ("+++++++  +++  +++  ++  ++")
3 print ("+++  ++++++  ++++  +++  +++  ")
4 print ("+++  ++++++  ++++  ++++  +++  ")
5 print ("+++  +++  ++++  ++++  +++  ")
6
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 2: Python

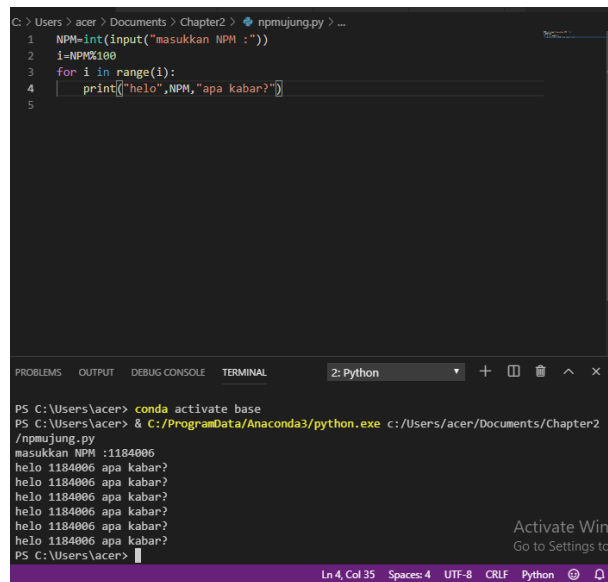
Nama Pengirim : Murnia Lestari  
Nama Penerima : Hani Dian  
Alamat Pengirim : Kelapa Tujuh  
No HP : 0895610009160  
PS C:\Users\acer> & C:/ProgramData/Anaconda3/python.exe c:/Users/acer/Documents/Chapter2/npmplus.py

+++++ +++ +++ ++ + ++  
+++++++ +++ +++ ++ ++  
+++ ++++++ ++++ +++ +++   
+++ ++++++ ++++ ++++ +++   
+++ +++ ++++ ++++ +++

Activate Windows  
Go to Settings to activate Windows.

PS C:\Users\acer>

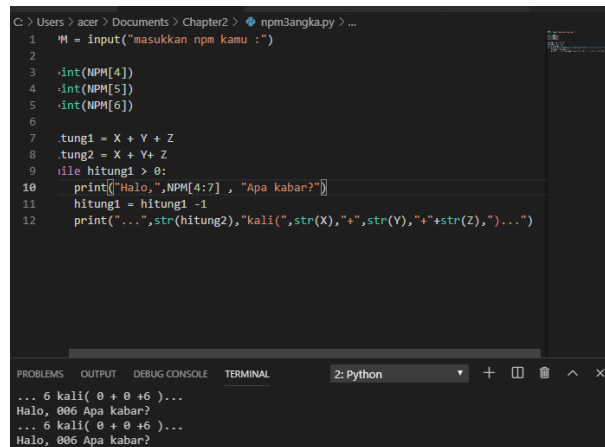
2. Program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama **NPM** dan output sebanyak dua dijit belakang NPM, contoh NPM : 113040087 maka akan ada output sebanyak 87 dengan tulisan 'Halo, 113040087 apa kabar?'



```
C:\Users\acer> Documents > Chapter2 > npmujung.py > ...
1  NPM=int(input("masukkan NPM :"))
2  i=NPM%100
3  for i in range(i):
4      print("halo",NPM,"apa kabar?")
5

PS C:\Users\acer> conda activate base
PS C:\Users\acer> & C:/ProgramData/Anaconda3/python.exe c:/Users/acer/Documents/Chapter2
/npmujung.py
masukkan NPM : 1184006
halo 1184006 apa kabar?
halo 1184006 apa kabar?
halo 1184006 apa kabar?
halo 1184006 apa kabar?
halo 1184006 apa kabar?
halo 1184006 apa kabar?
halo 1184006 apa kabar?
PS C:\Users\acer>
```

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 dijit tersebut.



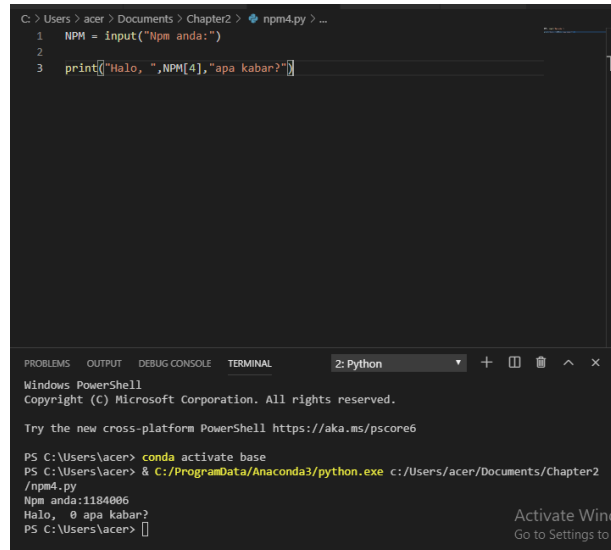
```
C:\Users\acer> Documents > Chapter2 > npm3angka.py > ...
1  NPM = input("masukkan npm kamu :")
2
3  .int(NPM[4])
4  .int(NPM[5])
5  .int(NPM[6])
6
7  .tung1 = X + Y + Z
8  .tung2 = X + Y+ Z
9  .ille hitung1 > 0:
10     print("Halo,",NPM[4:7] , "Apa kabar?")
11     hitung1 = hitung1 -1
12     print("...",str(hitung2),"kali(",str(X),"+",str(Y),"+",str(Z),"...")

... 6 kali( 0 + 0 +6 )...
Halo, 006 Apa kabar?
... 6 kali( 0 + 0 +6 )...
Halo, 006 Apa kabar?
```

4. Program hello word dengan input nama yang disimpan dalam sebuah variabel string bernama **NPM** dan beri luaran output berupa digit ketiga



dari belakang dari variabel NPM.



The screenshot shows a code editor with a Python script named `npm4.py` and a terminal window below it. The script contains three lines of code: `1 NPM = input("Npm anda:");`, `2` (blank line), and `3 print("Halo, ", NPM[4], "apa kabar?");`. The terminal window shows the execution of the script, where the user has entered `1184006` for the NPM number. The output of the script is `Halo, 0 apa kabar?`.

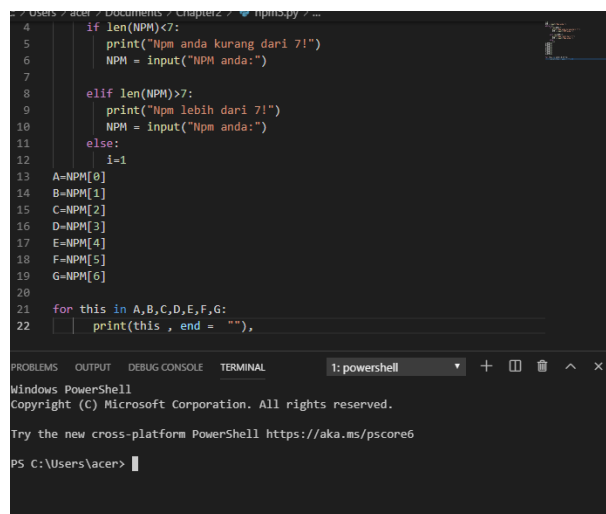
```
C:\Users\acer> Documents\Chapter2> npm4.py ...
1 NPM = input("Npm anda:")
2
3 print("Halo, ", NPM[4], "apa kabar?");

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\acer> conda activate base
PS C:\Users\acer> & C:/ProgramData/Anaconda3/python.exe c:/Users/acer/Documents/Chapter2/npm4.py
Npm anda:1184006
Halo, 0 apa kabar?
PS C:\Users\acer>
```

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 a code editor with a Python script named `npm4.py` and a terminal window below it. The script contains a conditional statement that checks the length of the NPM number. If the length is less than 7, it prints "Npm anda kurang dari 7!". If the length is greater than 7, it prints "Npm lebih dari 7!". If the length is exactly 7, it prints "Npm anda:" and then assigns the NPM number to a list of variables A through G. The terminal window shows the execution of the script, where the user has entered `1184006` for the NPM number. The output of the script is `Halo, 0 apa kabar?`.

```
C:\Users\acer> Documents\Chapter2> npm4.py ...
4 if len(NPM)<7:
5     print("Npm anda kurang dari 7!")
6     NPM = input("Npm anda:")
7
8 elif len(NPM)>7:
9     print("Npm lebih dari 7!")
10    NPM = input("Npm anda:")
11 else:
12     i=1
13     A=NPM[0]
14     B=NPM[1]
15     C=NPM[2]
16     D=NPM[3]
17     E=NPM[4]
18     F=NPM[5]
19     G=NPM[6]
20
21 for this in A,B,C,D,E,F,G:
22     print(this , end = " "),

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\acer>
```

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

```
C:\Users\acer> Documents > Chapter2 > npm7.py > ...
4      if len(NPM)<7:
5          print("Npm anda kurang dari 7!")
6          NPM = input("NPM anda:")
7
8      elif len(NPM)>7:
9          print("Npm lebih dari 7!")
10         NPM = input("Npm anda:")
11     else:
12         i=1
13     A=NPM[0]
14     B=NPM[1]
15     C=NPM[2]
16     D=NPM[3]
17     E=NPM[4]
18     F=NPM[5]
19     G=NPM[6]
20     X=1
21     for this in A,B,C,D,E,F,G:
22         X*=int(this)
23         print(X)
```

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

```
C:\Users\acer> Documents > Chapter2 > npm8.py > ...
4      if len(NPM)<7:
5          print("Npm anda kurang dari 7!")
6          NPM = input("NPM anda:")
7
8      elif len(NPM)>7:
9          print("Npm lebih dari 7!")
10         NPM = input("Npm anda:")
11     else:
12         i=1
13     A=NPM[0]
14     B=NPM[1]
15     C=NPM[2]
16     D=NPM[3]
17     E=NPM[4]
18     F=NPM[5]
19     G=NPM[6]
20
21     for this in A,B,C,D,E,F,G:
22         print(this)
```

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

```
C:\Users\acer> Documents > Chapter2 > npm9.py > ...
1      i=0
2      NPM = input("Npm anda:")
3      while i<1:
4          if len(NPM)<7:
5              print("Npm anda kurang dari 7!")
6              NPM = input("NPM anda:")
7
8          elif len(NPM)>7:
9              print("Npm lebih dari 7!")
10             NPM = input("Npm anda:")
11         else:
12             i=1
13     A=NPM[0]
14     B=NPM[1]
15     C=NPM[2]
16     D=NPM[3]
17     E=NPM[4]
18     F=NPM[5]
19     G=NPM[6]
20     X=1
```

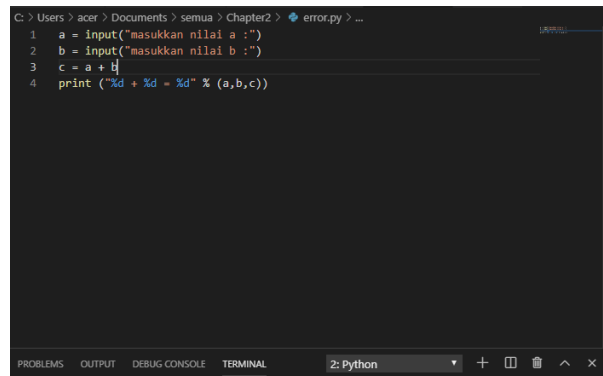
9. Dari soal no 5, Lakukan print NPM anda tapi hanya dijit genap saja.

```
C:\Users\acer> Documents > semua > Chapter2 > error.py > ...
1 a = input("masukkan nilai a :")
2 b = input("masukkan nilai b :")
3 c = a + b
4 print ("%d + %d = %d" % (a,b,c))
```

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

```
C:\Users\acer> Documents > Chapter2 > npm11.py > ...
14 B=NPM[1]
15 C=NPM[2]
16 D=NPM[3]
17 E=NPM[4]
18 F=NPM[5]
19 G=NPM[6]
20 X=1
21 for X in A,B,C,D,E,F,G:
22     if int(X) > 1:
23         for i in range(2,int(X)):
24             if (int(X) % i) == 0:
25                 break
26     else :
27         print(int(X),end = " "),
```

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



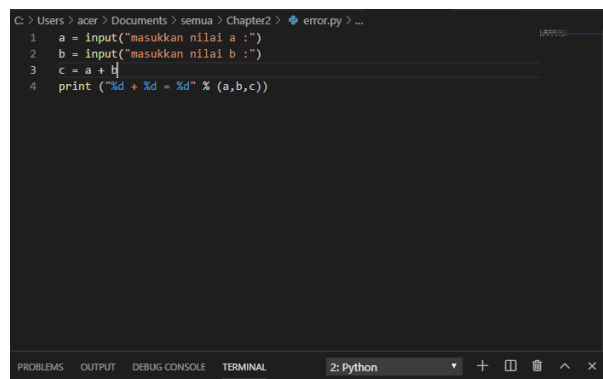
```
C:\Users> acer > Documents > semua > Chapter2 > error.py > ...
1  a = input("masukkan nilai a :")
2  b = input("masukkan nilai b :")
3  c = a + b
4  print ("%d + %d = %d" % (a,b,c))
```

### 3 Keterampilan Penanganan Error

Bagian Penanganan error dari script python.

- (a) 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 d untuk integer dan bukan string. Tetapi tipe data yang dimasukkan atau diinputkan adalah string, maka terjadi error.



```
C:\Users> acer > Documents > semua > Chapter2 > error.py > ...
1  a = input("masukkan nilai a :")
2  b = input("masukkan nilai b :")
3  c = a + b
4  print ("%d + %d = %d" % (a,b,c))
```

Error tersebut diatasi dengan try dan except

```
C:\Users\acer> Documents > semua > Chapter2 > error.py > ...
1 a = input("masukkan nilai a :")
2 b = input("masukkan nilai b :")
3 c = a + b
4 try:
5     print ("%d + %d = %d" % (a,b,c))
6 except :
7     print ("Tidak sesuai")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python
PS C:\Users\acer> & C:/ProgramData/Anaconda3/python.exe c:/Users/acer/Documents/semua/Chapter2/error.py
File "c:/Users/acer/Documents/semua/Chapter2/error.py", line 5
    print ("%d + %d = %d" % (a,b,c))
    ^
IndentationError: expected an indented block
PS C:\Users\acer> & C:/ProgramData/Anaconda3/python.exe c:/Users/acer/Documents/semua/Chapter2/error.py
masukkan nilai a :1
masukkan nilai b :2
Tidak sesuai
PS C:\Users\acer>
```

- (b) 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 jenis errornya, gunakan try except untuk menunjukkan error tersebut dengan bahasa indonesia.

```
C:\Users\acer> Documents > semua > Chapter2 > error2.py > ...
1 A = 8
2 B = "2"
3 try:
4     A+B
5 except :
6     print ("integer harus dengan integer dan string harus dengan string")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\acer> conda activate base
PS C:\Users\acer> & C:/ProgramData/Anaconda3/python.exe c:/Users/acer/Documents/semua/Chapter2/error2.py
integer harus dengan integer dan string harus dengan string
PS C:\Users\acer>
```