

Tugas Chapter 2

Pemrograman II



Alifia Zahra

1184051

D4 Teknik Informatika 2B

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

Politeknik Pos Indonesia

Bandung 2019

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

Acknowledgements

Pertama-tama kami panjatkan puji dan syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga Modul Praktikum ini dapat diselesaikan.

Abstract

Modul Praktikum ini dibuat dengan tujuan memberikan acuan, bagi mahasiswa dan dosen Pengajar Mata Kuliah. Pada intinya buku ini menjelaskan secara lengkap tentang Standar penilaian mata kuliah pemrograman II di Program Studi D4 Teknik Informatika, dan juga mengatur mekanisme, teknik penulisan, serta penilaiannya. Dengan demikian diharapkan semua pihak yang terlibat dalam aktivitas belajar dan mengajar berjalan lancar dan sesuai dengan standar.

Contents

1	Pemrograman Dasar	1
1.1	Teori	1
1.1.1	Jenis-Jenis Variabel dan Cara Pemakaian Variabel	1
1.1.2	Kode Untuk Meminta Input Dari User dan Melakukan Output ke Layar	3
1.1.3	Operator Dasar Aritmatika, Tambah, Kali, Kurang, Bagi, dan Bagaimana Mengubah String ke Integer dan Integer Ke String	4
1.1.4	Penjelasan Syntax Untuk Perulangan dan Jenis-Jenisnya serta Contoh Kode dan Cara Menggunakannya di Python	5
1.1.5	Cara Menggunakan Syntax Untuk Memilih Kondisi dan Contoh Syntax Kondisi Di Dalam Kondisi	6
1.1.6	Jenis error yang sering ditemui di Python dalam mengerjakan Syntax di atas	8
1.1.7	Cara Memakai Try Except	9
1.2	Ketrampilan Pemrograman	9
1.3	Ketrampilan Penanganan Error	17

List of Figures

Chapter 1

Pemrograman Dasar

Tujuan pembelajaran pada pertemuan kedua antara lain:

1. Menenal Jenis Variabel Python
2. Input dan output user
3. Operator Dasar
4. Perulangan
5. Kondisi
6. Mengatasi Error
7. Try Except

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten IRC. Kode program dipisah dalam folder src NPM.py yang berisi praktek dari masing-masing tugas file terpisah sesuai nomor yang kemudian dipanggil menggunakan input listing ke dalam file latex penjelasan atau nomor pengerjaan. Masing masing soal bernilai 5 dengan total nilai 100.

1.1 Teori

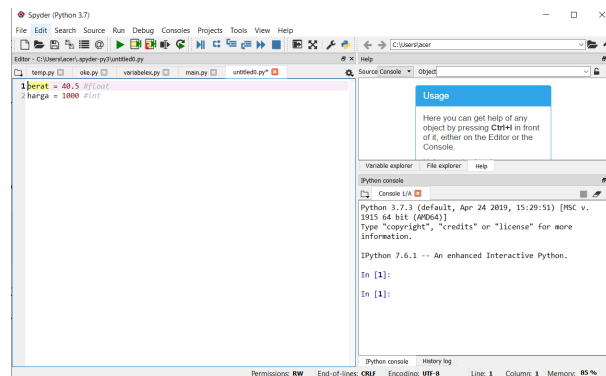
1.1.1 Jenis-Jenis Variabel dan Cara Pemakaian Variabel

Variabel merupakan suatu wadah untuk menyimpan data, sedangkan data yang disimpan di dalam variabel disebut 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 : 1,2,3,dst.
- (b) Float (bilangan pecahan), contoh : 1.5, 2.1, dst.

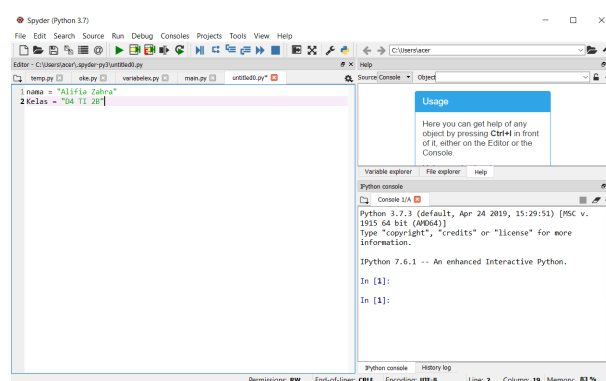


```
1 berat = 40.5 #float
2 harga = 1000
```

2. Tipe Data Teks

Tipe ini terbagi menjadi dua jenis, yaitu:

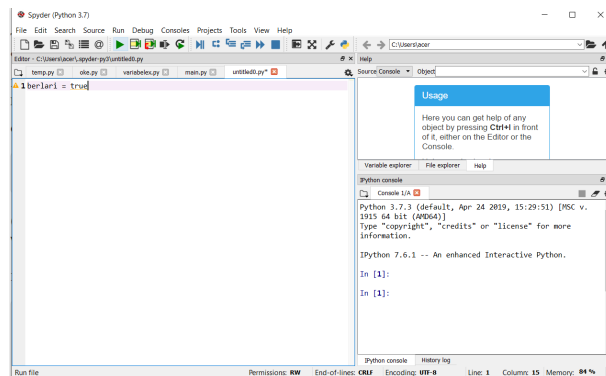
- (a) String (kumpulan karakter), Contoh : "Saya sedang makan"
- (b) Varchar (karakter), Contoh : a,b,c,dst.



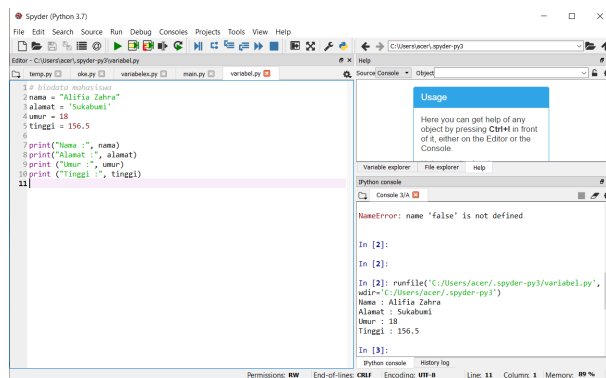
Penulisan tipe ini harus diapit oleh tanda petik.

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:



```

1 # biodata mahasiswa
2 nama = "Alifia Zahra"
3 alamat = 'Sukabumi'
4 umur = 18
5 tinggi = 156.5
6
7 print("Nama :", nama)
8 print("Alamat :", alamat)
9 print ("Umur :", umur)
10 print ("Tinggi :", tinggi)

```

1.1.2 Kode Untuk Meminta Input Dari User dan Melakukan Output ke Layar

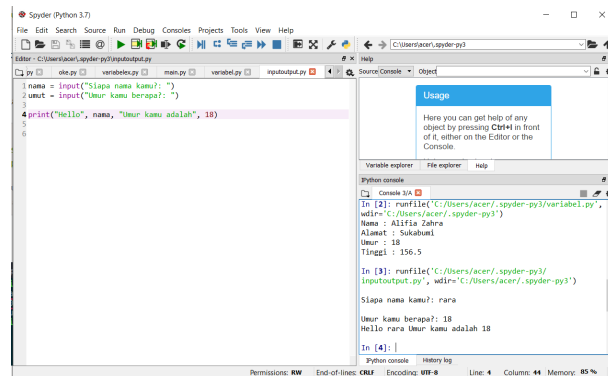
Input merupakan suatu masukan yang akan kita berikan ke program. Sedangkan hasil yang ditampilkan disebut output.

1. Cara mengambil input

Python menyediakan fungsi input, menggunakan kode input()

2. Cara menampilkan output

Untuk menampilkan sebuah output teks kita menggunakan kode `print()`



```
1 nama = input("Siapa nama kamu?: ")
2 umur = input("Umur kamu berapa?: ")
3
4 print("Hello", nama, "Umur kamu adalah", 18) #output
```

1.1.3 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.

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

Adapun contoh penggunaannya adalah:

Cara mengubah string ke integer dan integer ke string

```

1 a = int(input("Masukkan nilai a: "))
2 b = int(input("Masukkan nilai b: "))
3
4 # operator penjumlahan
5 c = a + b
6 print("Hasil %d + %d = %d" % (a,b,c))
7
8 # operator pengurangan
9 c = a - b
10 print("Hasil %d - %d = %d" % (a,b,c))
11
12 # operator perkalian
13 c = a * b
14 print("Hasil %d * %d = %d" % (a,b,c))
15
16 # operator pembagian
17 c = a / b
18 print("Hasil %d / %d = %d" % (a,b,c))

```

Python console output:

```

In [63]:
Masukkan nilai a: 5
Masukkan nilai b: 6
Hasil 5 + 6 = 11
Hasil 5 - 6 = -1
Hasil 5 * 6 = 30
Hasil 5 / 6 = 0

```

```

1 a=150
2 b="151"
3 string = str(a)
4 print(string)
5 integer = int(b)
6 print(integer)
7

```

Python console output:

```

In [14]:
Hasil 150

In [15]:
Hasil 151

In [16]:
Hasil 151

In [17]:

```

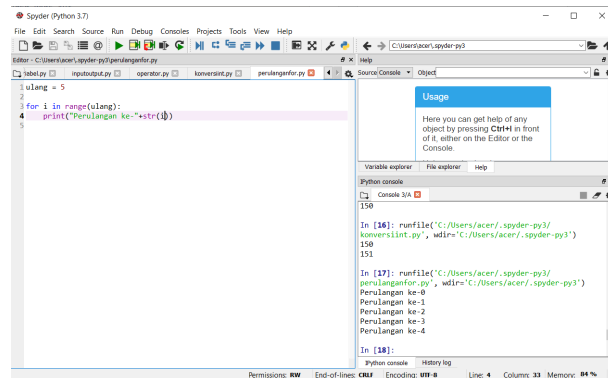
1.1.4 Penjelasan Syntax Untuk Perulangan dan Jenis-Jenisnya serta Contoh Kode dan Cara Menggunakannya di Python

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 (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

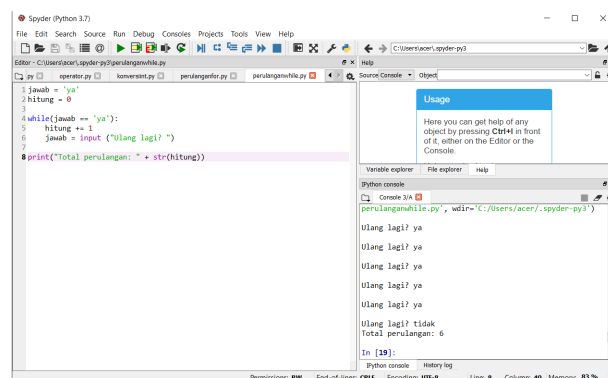
Perulangan ini biasanya digunakan untuk mengetahui kode yang sudah banyak perulangannya. Adapun contoh kodenya adalah:



```
1 ulang = 5
2
3 for i in (ulang):
4     print("Perulangan ke-"+str(i))
```

2. Perulangan While

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



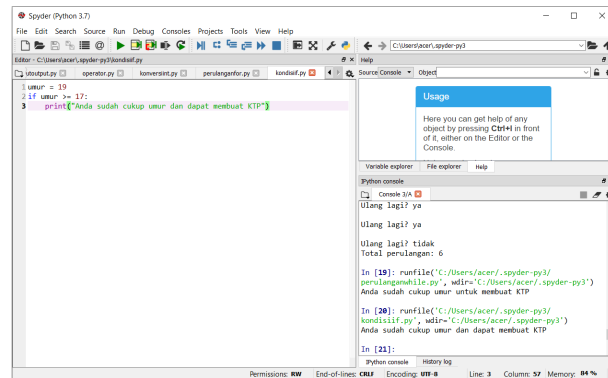
```
1 umur = 19
2 if umur > 17:
3     print("Anda sudah cukup umur untuk membuat KTP")
```

1.1.5 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.

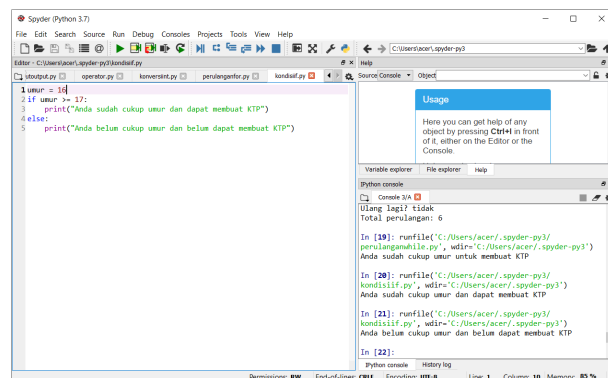
1. Kondisi If

Jika kondisi utama true, maka perintah akan dijalankan



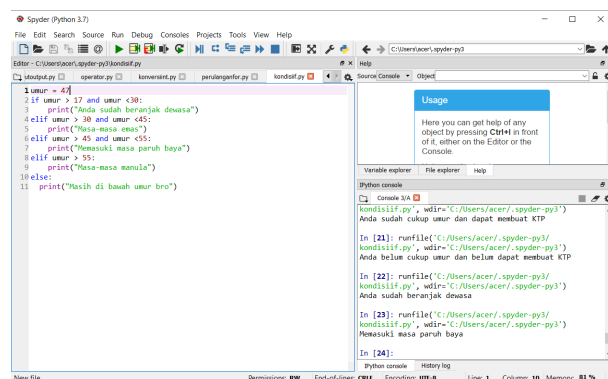
2. Kondisi If Else

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



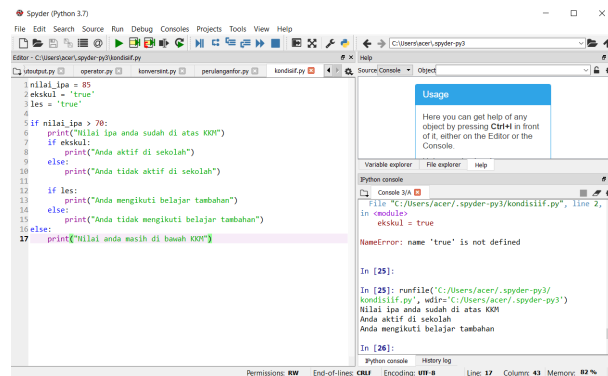
3. Kondisi if elif else

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



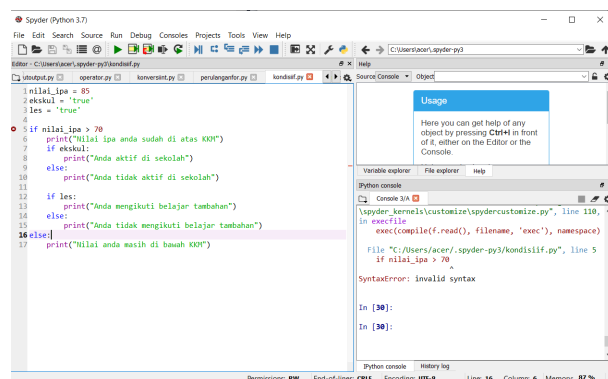
4. If di dalam if (If bersarang)

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

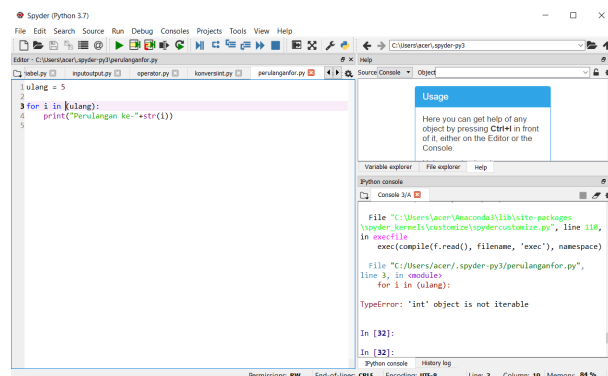


1.1.6 Jenis error yang sering ditemui di Python dalam mengerjakan Syntax di atas

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



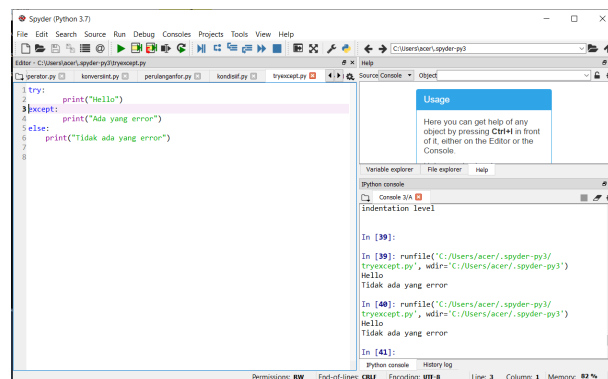
Gambar di atas merupakan contoh dari kesalahan syntax titik dua pada line 5.



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

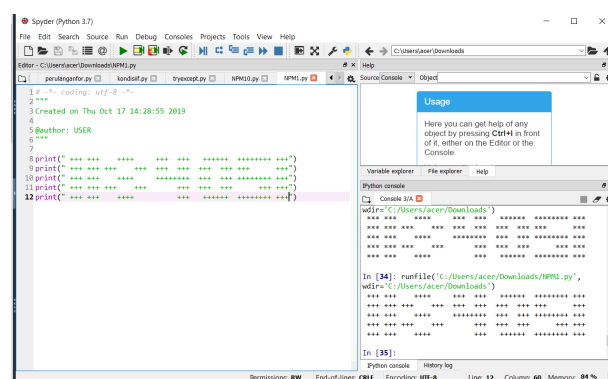
1.1.7 Cara Memakai Try Except

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



1.2 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:



```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Oct 17 14:28:55 2019
4
5 @author: USER
6 """
7

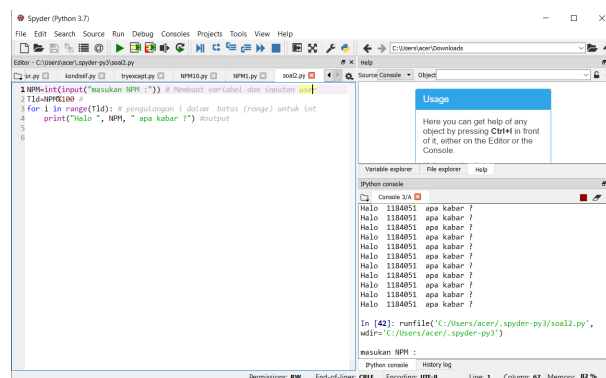
```

```

8 print(" +++ ++++ +++++ +++++ +++++ +++++ +++++ +++++")
9 print(" +++ ++++ +++++ +++++ +++++ +++++ +++++ +++++")
10 print(" +++ ++++ +++++ +++++ +++++ +++++ +++++ +++++")
11 print(" +++ ++++ +++++ +++++ +++++ +++++ +++++ +++++")
12 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 : 113040087 maka akan ada output sebanyak 87 dengan tulisan ‘Halo, 113040087 apa kabar?’

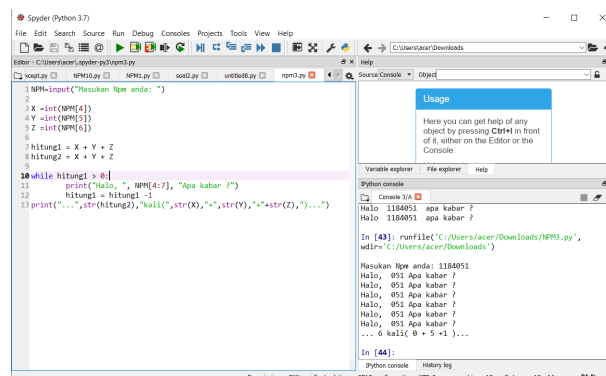


```

1 NPM=int(input("masukan NPM :")) # Membuat variabel dan inputan user
2 Tld=NPM%100 #
3 for i in range(Tld): # pengulangan i dalam batas (range) untuk int
4     print("Halo ", NPM, " apa kabar ?") #output

```

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 digit tersebut.



```

1 NPM=input("Masukan Npm anda: ")
2
3 X =int(NPM[4])
4 Y =int(NPM[5])

```

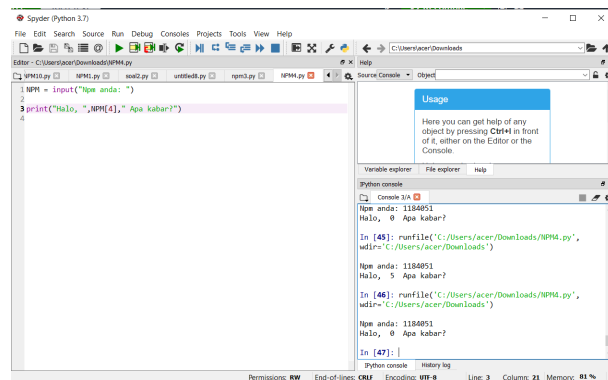


```

5 Z =int(NPM[6])
6
7 hitung1 = X + Y + Z
8 hitung2 = X + Y + Z
9
10 while hitung1 > 0:
11     print("Halo, ", NPM[4:7], "Apa kabar ?")
12     hitung1 = hitung1 -1
13 print(" ... ",str(hitung2)," kali(",str(X),"+",str(Y),"+"+str(Z),") ...
    ")

```

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.

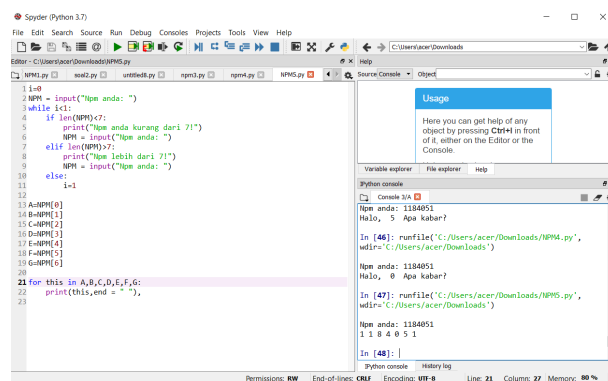


```

1 NPM = input("Npm anda: ")
2
3 print("Halo, ",NPM[4], " Apa kabar?")

```

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

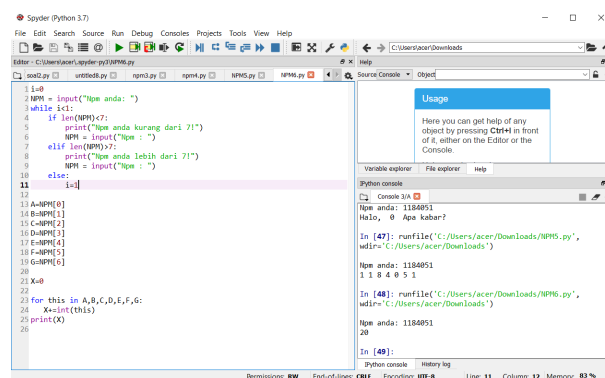


```

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     elif len(NPM) > 7:
8         print("Npm lebih dari 7!")
9         NPM = input("Npm anda: ")
10    else:
11        i=1
12
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 = " "),

```

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



```

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 : ")
7     elif len(NPM)>7:
8         print("Npm anda lebih dari 7!")
9         NPM = input("Npm : ")
10    else:
11        i=1
12
13 A=NPM[0]
14 B=NPM[1]
15 C=NPM[2]

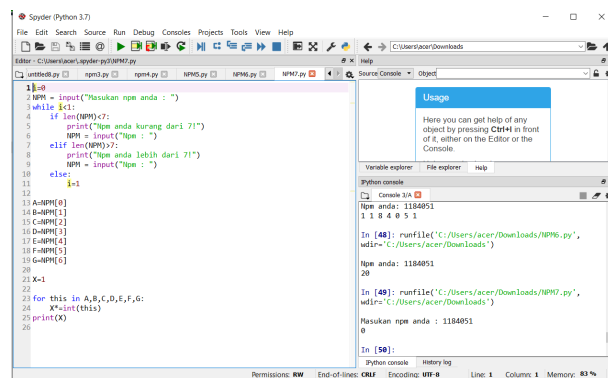
```

```

16 D=NPM[3]
17 E=NPM[4]
18 F=NPM[5]
19 G=NPM[6]
20
21 X=0
22
23 for this in A,B,C,D,E,F,G:
24     X+=int( this )
25 print(X)

```

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



```

1 i=0
2 NPM = input("Masukan npm anda : ")
3 while i < 1:
4     if len(NPM) < 7:
5         print("Npm anda kurang dari 7!")
6         NPM = input("Npm : ")
7     elif len(NPM) > 7:
8         print("Npm anda lebih dari 7!")
9         NPM = input("Npm : ")
10    else:
11        i=1
12
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 X=1
22
23 for this in A,B,C,D,E,F,G:
24     X*=int( this )
25 print(X)

```

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

The screenshot shows the Spyder Python IDE with a file named `NPM8.py` open. The script contains the following code:

```

1 i=0
2 NPM = input("Masukan Npm anda : ")
3 while i<1:
4     if len(NPM)<7:
5         print("Npm anda kurang dari 7!")
6         NPM = input("Npm : ")
7     elif len(NPM)>7:
8         print("Npm anda lebih dari 7!")
9         NPM = input("Npm : ")
10    else:
11        i=1
12
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)

```

The Python console on the right shows the execution output:

```

Masukan npm anda : 1184051
0
In [50]: runfile('C:/Users/acer/Downloads/NPM8.py',
      wdir='C:/Users/acer/Downloads')
Masukan Npm anda : 1184051
1
1
8
4
0
5
1
In [51]:

```

```

1 i=0
2 NPM = input("Masukan Npm anda : ")
3 while i<1:
4     if len(NPM)<7:
5         print("Npm anda kurang dari 7!")
6         NPM = input("Npm : ")
7     elif len(NPM)>7:
8         print("Npm anda lebih dari 7!")
9         NPM = input("Npm : ")
10    else:
11        i=1
12
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)

```

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

The screenshot shows the Spyder Python IDE with a file named `NPM9.py` open. The script contains the following code:

```

1 i=0
2 NPM = input("Masukan npm anda : ")
3 while i<1:
4     if len(NPM)<7:
5         print("Npm anda kurang dari 7!")
6         NPM = input("Npm : ")
7     elif len(NPM)>7:
8         print("Npm anda lebih dari 7!")
9         NPM = input("Npm : ")
10    else:
11        i=1
12
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 X=1
22
23 for this in A,B,C,D,E,F,G:
24     if int(this)%2==0:
25         if int(this)==0:
26             this=""
27         print(this,end=" ")
28
29

```

The Python console on the right shows the execution output:

```

Masukan npm anda : 1184051
1
1
8
4
0
5
1
In [51]: runfile('C:/Users/acer/Downloads/NPM9.py',
      wdir='C:/Users/acer/Downloads')
Masukan npm anda : 1184051
8 4
In [52]:

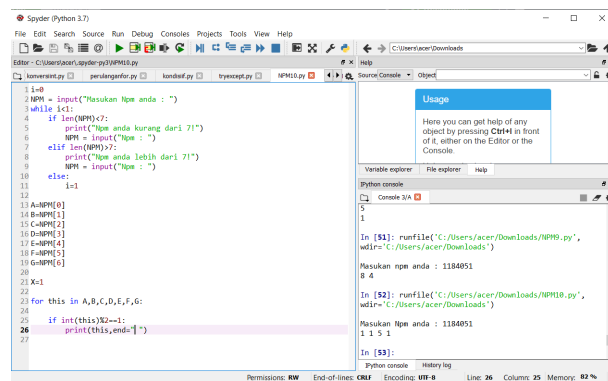
```

```

1 i=0
2 NPM = input("Masukan npm anda : ")
3 while i<1:
4     if len(NPM)<7:
5         print("Npm anda kurang dari 7!")
6         NPM = input("Npm : ")
7     elif len(NPM)>7:
8         print("Npm anda lebih dari 7!")
9         NPM = input("Npm : ")
10    else :
11        i=1
12
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 X=1
22
23 for this in A,B,C,D,E,F,G:
24
25     if int(this)%2==0:
26         if int(this)==0:
27             this=""
28     print(this ,end =" ")

```

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



```

1 i=0
2 NPM = input("Masukan Npm anda : ")
3 while i<1:
4     if len(NPM)<7:
5         print("Npm anda kurang dari 7!")
6         NPM = input("Npm : ")
7     elif len(NPM)>7:
8         print("Npm anda lebih dari 7!")
9         NPM = input("Npm : ")

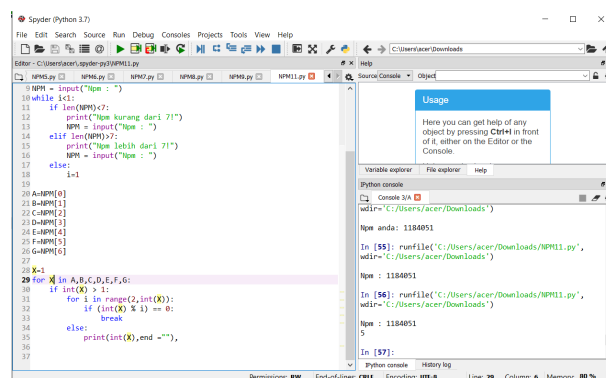
```

```

10     else :
11         i=1
12
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 X=1
22
23 for this in A,B,C,D,E,F,G:
24
25     if int(this)%2==1:
26         print(this ,end=" ")

```

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



```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Tue Oct 22 14:34:55 2019
4
5  @author: USER
6  """
7
8  i=0
9  NPM = input("Npm : ")
10 while i<1:
11     if len(NPM)<7:
12         print("Npm kurang dari 7!")
13         NPM = input("Npm : ")
14     elif len(NPM)>7:
15         print("Npm lebih dari 7!")
16         NPM = input("Npm : ")
17     else :
18         i=1

```

```

19
20 A=NPM[0]
21 B=NPM[1]
22 C=NPM[2]
23 D=NPM[3]
24 E=NPM[4]
25 F=NPM[5]
26 G=NPM[6]
27
28 X=1
29 for X in A,B,C,D,E,F,G:
30     if int(X) > 1:
31         for i in range(2,int(X)):
32             if (int(X) % i) == 0:
33                 break
34     else:
35         print(int(X),end=""),

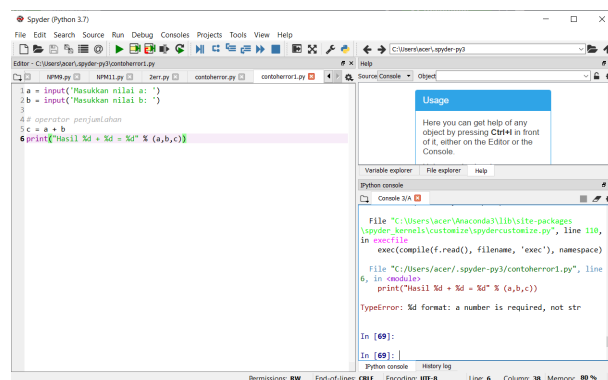
```

1.3 Ketrampilan 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 d untuk integer dan bukan string. Tetapi tipe data yang dimasukkan atau diinputkan adalah string, maka terjadi error.

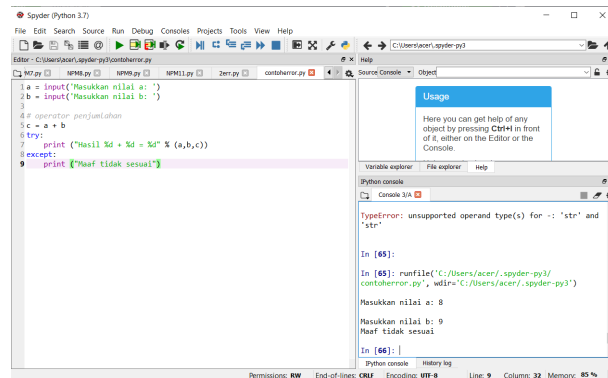


```

1 a = input('Masukkan nilai a: ')
2 b = input('Masukkan nilai b: ')
3
4 # operator penjumlahan
5 c = a + b
6 print("Hasil %d + %d = %d" % (a,b,c))

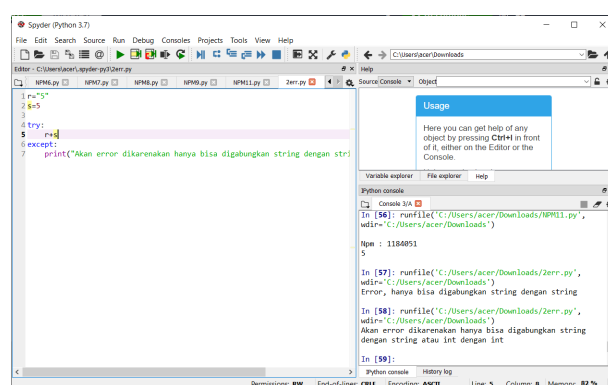
```

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



```
1 a = input('Masukkan nilai a: ')
2 b = input('Masukkan nilai b: ')
3
4 # operator penjumlahan
5 c = a + b
6 try:
7     print ("Hasil %d + %d = %d" % (a,b,c))
8 except:
9     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 jenis errornya, gunakan try except untuk menunjukkan error tersebut dengan bahasa indonesia.



```
1 r="5"
2 s=5
3
4 try:
5     r+s
6 except:
```



```
7 print("Akan error dikarenakan hanya bisa digabungkan string  
dengan string atau int dengan int")
```