

Tugas Chapter 2

Pemrograman II



Dyning Aida Batrishya

1184030

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

Assalamualaikum warahmatullahi wabarakaatuh. Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga dapat menyelesaikan laporan Tugas Chapter 1 ini, tanpa bantuan-Nya maka penulis tidak dapat menyelesaikannya dengan baik dan tepat pada waktunya. Shalawat serta salam semoga terlimpahkan kepada junjungan Nabi Muhammad SAW yang akan kita nantikan syafaatnya di yaumul qimayah nanti.

Laporan ini disusun guna memenuhi kelulusan matakuliah Pemrograman II Program Studi DIV Teknik Informatika. Proses penyelesaian laporan ini tidak luput dari bantuan berbagai pihak. Oleh karenanya, penulis mengucapkan terima kasih kepada :

1. Allah SWT yang telah memberikan rahmat dan hidayah-Nya
2. Orang tua yang selalu memberikan dukungan dan motivasi dalam penyelesaian laporan
3. Bapak Rolly Awangga yang telah memberikan dukungan dan bantuan dalam penyelesaian laporan
4. Teman-teman yang saya sayangi yang selalu memberikan dukungan dan motivasinya kepada penulis

Penulis mengharapkan kritik dan saran dari pembaca jika terdapat kesalahan dalam penyusunan laporan ini sehingga penulis dapat memperbaiki penyelesaian tugas yang selanjutnya

Bandung, 16 Oktober 2019

Penulis

Chapter 1

LAPORAN

1.1 TUGAS TEORI

- Jenis Variabel dan Cara Pemakaiannya di Python

Variabel adalah tempat yang digunakan untuk menampung data atau nilai dari suatu data. Untuk menyimpan suatu nilai, variabel dapat menggunakan berbagai macam tipe data dan bersifat dinamis. Sehingga, dalam pendeklarasiannya, variabel tidak diperlukan untuk meenyertakan tipe data apa yang digunakan dan nilai variabel tersebut juga dapat diubah. Berikut ini merupakan aturan dalam penulisan variabel di dalam script python :

1. Karakter awal yang digunkaan harus berupa huruf ataupun gatis bawah
2. Selain karakter pertama, karakter yang digunakan dapat berupa angka, huruf, ataupun underscore (garis bawah)
3. Karakter ynag digunakan pada penamaan variabel bersifat case sensitive, sehingga penggunaan huruf besar dan huruf kecil sangat berpengaruh dalam menjalankan script.

Berikut ini contoh pengaplikasian variabel di dalam script



```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 16 18:58:08 2019
@author: AS5
"""
nama = "Syling Alida Batrisyha"
print(nama)
```

Figure 1.1: implementasi variabel

- Kode Memasukkan Input dari User dan Menghasilkan Output ke Layar.

1. Input merupakan masukan yang diberikan kepada program. Dalam python, terdapat 2 sintaks fungsi yang digunakan untuk mengambil inputan. Di antaranya yaitu :

- (a) fungsi `input()`

fungsi ini digunakan untuk mengambil inputan berupa data angka. Namun pada python 3, fungsi ini juga dapat digunakan untuk mengambil inputan berupa data huruf

- (b) fungsi `input` berupa raw input

fungsi ini digunakan untuk mengambil inputan berupa data huruf. Fungsi inputan jenis ini hanya dapat digunakan pada python versi 2. Sedangkan pada python versi 3, semua fungsi input dapat digunakan pada fungsi `input()`

2. Output merupakan hasil keluaran yang dihasilkan dari suatu program. Untuk menghasilkan keluaran yang dibuat oleh program, maka digunakan fungsi `print()` di dalam script program

- Operator Dasar Matematika dan Mengubah Tipe Data

- dasar yang ada pada bahasa pemrograman python ada 6. Di antaranya yaitu :

1. Aritmatika

Berikut ini merupakan operasi dasar aritmatika pada python :

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

Figure 1.2: operator aritmatika

2. Pembandingan

Berikut ini merupakan operasi pembandingan pada python :

Operator	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	==
Tidak Sama dengan	!=
Lebih Besar Sama dengan	>=
Lebih Kecil Sama dengan	<=

Figure 1.3: operator pembandingan

3. Penugasan

Operator penugasan memiliki fungsi untuk memberi tugas kepada variabel.

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

Figure 1.4: operator pembandingan

4. Logika

Operator logika berfungsi untuk menjalankan operasi logika. Operasi logika terdiri atas or, and dan not. Berikut ini merupakan notasi yang merupakan operator logika

Nama	Simbol di Python
Logika AND	and
Logika OR	or
Negasi/kebalikan	not

Figure 1.5: operator logika

5. Bitwise

Operasi ini merupakan operasi yang digunakan pada bilangan bit ataupun binary. Operasi bitwise meliputi :

Nama	Simbol
AND	&
OR	
XOR	^
Negasi/kebalikan	~
Left Shift	<<
Right Shift	>>

Figure 1.6: operator bitwise

6. Ternary

Operator ini disebut juga sebagai operator kondisi. Hal itu dikarenakan operator ini biasanya digunakan pada percabangan if else.

– Mengkonversi Tipe Data

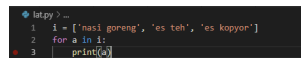
1. Tipe data Integer
2. Tipe data String
3. Tipe data Float
4. Tipe data Complex

Untuk mengkonversi tipe data, kita hanya perlu untuk menambahkan tipe data yang akan kita konversikan menjadi. Contoh penamaannya : `string = str(var)` `float = float(var)` `kompleks = complex(var)` `long = long(var)`
keterangan : `var` merupakan variabel yang akan kita ubah tipe datanya.

• Sintaks Perulangan

1. FOR

For merupakan perulangan yang akan mengulang kondisi true sampai batas yang telah ditentukan. Berikut ini merupakan contoh penggunaan sintaks perulangan for.

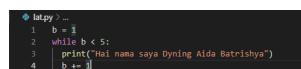


```
1 i = ['nasi goreng', 'es teh', 'es kopyor']
2 for a in i:
3     | print(a)
```

Figure 1.7: contoh penggunaan kondisi perulangan for

2. While

While merupakan perulangan yang akan terjadi dengan dua kondisi jika benar dan jika salah. Berikut ini merupakan contoh penggunaan sintaks perulangan while.



```
1 b = 1
2 while b < 5:
3     | print('Hai nama saya Dyming Aida Batrishya')
4     | b += 1
```

Figure 1.8: contoh penggunaan perulangan while

• Sintaks Kondisi

1. Struktur if dengan kondisi tunggal.

Struktur if dengan kondisi tunggal adalah suatu struktur yang memiliki suatu perlakuan jika terjadi suatu kondisi. Akan tetapi, tidak terjadi sesuatu yang lain atau terjadi apa-apa ketika berada di dalam luar kondisi tersebut. struktur if kondisi tunggal yaitu :

if (kondisi):

aksi

contoh penggunaan struktur if itu sendiri di antaranya :

```
lat.py > ...
1 x=7
2 if x > 5:
3     print("bilangan lebih besar dari 5")
```

Figure 1.9: contoh penggunaan if kondisi tunggal

2. Struktur if else

Struktur if else hampir sama dengan struktur if yang sebelumnya. Akan tetapi, pada struktur if else ini sendiri memiliki perbedaan dengan memiliki perlakuan terhadap 2 kondisi.

struktur if else yaitu :

if (kondisi):

aksi

else

aksi2

Berikut ini merupakan contoh pengaplikasiannya :

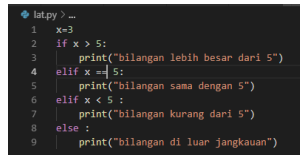
```
lat.py > ...
1 x=3
2 if x > 5:
3     print("bilangan lebih besar dari 5")
4 else:
5     print("bilangan kurang dari 5")
```

Figure 1.10: contoh aplikasi struktur if else

3. Kondisi elif dan logika majemuk

Kondisi if elif merupakan suatu struktur logika majemuk yang memiliki banyak pilihan aksi terhadap berbagai kemungkinan kejadian yang terjadi.

Contoh pengaplikasian if elif yaitu :



```

1 x=3
2 if x > 5:
3     print("bilangan lebih besar dari 5")
4 elif x == 5:
5     print("bilangan sama dengan 5")
6 elif x < 5:
7     print("bilangan kurang dari 5")
8 else:
9     print("bilangan di luar jangkauan")

```

Figure 1.11: penggunaan struktur if elif

4. Nested If atau if bersarang

Nested if atau if bersarang, seperti namanya, ialah suatu struktur yang memiliki suatu keadian if di dalam suatu kejadian if yang lainnya.

struktur penulisannya yaitu :

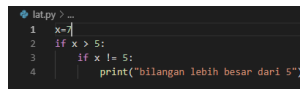
if(kondisi):

aksi

if(kondisi):

aksi

berikut ini contoh pengaplikasian nested if :



```

1 x=7
2 if x > 5:
3     if x != 5:
4         print("bilangan lebih besar dari 5")

```

Figure 1.12: penggunaan struktur if sarang atau nested if

• Jenis Error yang sering terjadi di Python

1. Logic Error.

Logic error merupakan kesalahan yang terjadi karena kesalahan pembacaan data pada command perintah seperti data tidak terbaca atau tidak ada, dan tidak sesuai dengan aturannya.

Contoh kesalahan tipe data yaitu :

a='4'

b=6

print(a+b)

2. Syntax Error

Syntax error merupakan kesalahan yang disebabkan karena terjadinya kesalahan penulisan baik format, command, karakter, ataupun simbol. Error yang disebabkan oleh kesalahan penulisan ini menyebabkan program yang telah dibuat tidak dapat berjalan dengan baik. Contohnya pemberian titik

dua atau colon (:) pada command if(kondisi):, for i in (range):

3. Error Indentasi

Error indentasi merupakan kesalahan yang terjadi karena spasi atau menjorok. Indentasi itu sendiri berasal dari bahasa inggris Indentation yang artinya menjorokkan. Indentasi biasanya digunakan di bahasa pemrograman lainnya untuk memperindah dan mempermudah pembacaan command perintah saja. Sedangkan pada bahasa pemrograman python, indentasi digunakan sebagai penanda blok program. Sehingga indentasi pada python sangatlah penting karena jika dalam blok program tidak memiliki indentasi, maka program akan error dan tidak dapat dijalankan. Contoh error indentasi yaitu pada penulisan perintah if else sebagai berikut :

```
1 x=3
2 if x > 5:
3     print("bilangan lebih besar dari 5")
4 else:
5     print("bilangan kurang dari 5")
```

Figure 1.13: penggunaan indentasi pada script

- Try Except

Try except merupakan penanganan error yang biasa terjadi ketika dalam penggunaan IO, database dan pengaksesan indeks suatu list atau dictionary.

```
1 try:
2     print("Hai!")
3 except:
4     print("Ditemukan kesalahan")
5 else:
6     print("Tidak ditemukan error")
```

Figure 1.14: contoh penggunaan try except

1.2 KETERAMPILAN

```
1 # -*- coding: utf-8 -*-
2 """
3
4 @author: ASS
5 """
6
7 print(" ++++ ++++      ++++++      ++++ ++++      ++++++      ++++++      ++++++ ")
8 print(" ++++ ++++ ++++      ++++      ++++ ++++      ++++ ++++      ++++ ++++ ")
9 print(" ++++ ++++      ++++++      ++++++      ++++ ++++      ++++ ++++ ")
```

```

10 print(" +++ +++ +++    +++          +++  +++  +++    +++  +++  +++ ")
11 print(" +++ +++    +++          +++  ++++++  ++++++  ++++++ ")

```

```

1 # -*- coding: utf-8 -*-
2 """
3
4 @author: ASS
5 """
6
7 NPM=int(input("berapa npm kamu? :"))
8 tigadigit=NPM%100
9 for i in range(tigadigit):
10     print("Halo ", NPM, " apa kabar ?")

```

```

1 # -*- coding: utf-8 -*-
2 """
3 @author: ASS
4 """
5
6 NPM=input("berapa npm kamu? ")
7
8 A =int(NPM[4])
9 B =int(NPM[5])
10 C =int(NPM[6])
11
12 hitung1 = A + B + C
13 hitung2 = A + B + C
14
15 while hitung1 > 0:
16     print("Halo, ", NPM[4:7], "Apa kabar ?")
17     hitung1 = hitung1 -1
18 print(" ...",str(hitung2),"kali(",str(A),"+",str(B),"+",str(C),") ...")

```

```

1 # -*- coding: utf-8 -*-
2 """
3
4 @author: ASS
5 """
6 NPM = input("Npm kamu: ")
7
8 print("Halo, ",NPM[4], " Apa kabar?")

```

```

1 # -*- coding: utf-8 -*-
2 """
3
4 @author: ASS
5 """
6
7 i=0
8 NPM = input("Npm : ")
9 while i<1:
10     if len(NPM)<7:
11         print("Npmnya kurang dari 7 digit")
12         NPM = input("Npm : ")

```

```

13     elif len(NPM) > 7:
14         print("Npm lebih dari digit")
15         NPM = input("Npm : ")
16     else :
17         i=1
18
19 A=NPM[0]
20 B=NPM[1]
21 C=NPM[2]
22 D=NPM[3]
23 E=NPM[4]
24 F=NPM[5]
25 G=NPM[6]
26
27 for this in A,B,C,D,E,F,G:
28     print(this ,end = " ")

```

```

1 # -*- coding: utf-8 -*-
2 """
3
4 @author: ASS
5 """
6
7 i=0
8 NPM = input("Npm kamu adalah : ")
9 while i < 1:
10     if len(NPM) < 7:
11         print("Npm kurang dari digit")
12         NPM = input("Npm kamu: ")
13     elif len(NPM) > 7:
14         print("Npm lebih dari 7 digit")
15         NPM = input("Npm kamu: ")
16     else :
17         i=1
18
19 A=NPM[0]
20 B=NPM[1]
21 C=NPM[2]
22 D=NPM[3]
23 E=NPM[4]
24 F=NPM[5]
25 G=NPM[6]
26
27 X=0
28
29 for this in A,B,C,D,E,F,G:
30     X+=int(this)
31 print(X)

```

```

1 # -*- coding: utf-8 -*-
2 """
3
4 @author: ASS
5 """

```

```

6
7 i=0
8 NPM = input("Npm kamu berapa: ")
9 while i<1:
10     if len(NPM)<7:
11         print("Npm kurang dari 7 digit")
12         NPM = input("Npm : ")
13     elif len(NPM)>7:
14         print("Npm lebih dari 7 digit")
15         NPM = input("Npm : ")
16     else:
17         i=1
18
19 A=NPM[0]
20 B=NPM[1]
21 C=NPM[2]
22 D=NPM[3]
23 E=NPM[4]
24 F=NPM[5]
25 G=NPM[6]
26
27 X=1
28
29 for this in A,B,C,D,E,F,G:
30     X*=int(this)
31 print(X)

```

```

1 # -*- coding: utf-8 -*-
2 """
3
4 @author: ASS
5 """
6
7 i=0
8 NPM = input("berapa npm kamu : ")
9 while i<1:
10     if len(NPM)<7:
11         print("Npmnya kurang dari 7 digit")
12         NPM = input("Npm kamu : ")
13     elif len(NPM)>7:
14         print("Npm lebih dari 7 digit")
15         NPM = input("Npm kamu: ")
16     else:
17         i=1
18
19 A=NPM[0]
20 B=NPM[1]
21 C=NPM[2]
22 D=NPM[3]
23 E=NPM[4]
24 F=NPM[5]
25 G=NPM[6]
26
27 for this in A,B,C,D,E,F,G:

```

```

28     print(this)

1  # -*- coding: utf-8 -*-
2  """
3
4  @author: ASS
5  """
6
7  i=0
8  NPM = input("berapa npm kamu : ")
9  while i<1:
10     if len(NPM)<7:
11         print("Npmnya kurang dari 7 digit")
12         NPM = input("Npm kamu : ")
13     elif len(NPM)>7:
14         print("Npm lebih dari 7 digit")
15         NPM = input("Npm kamu: ")
16     else:
17         i=1
18
19  A=NPM[0]
20  B=NPM[1]
21  C=NPM[2]
22  D=NPM[3]
23  E=NPM[4]
24  F=NPM[5]
25  G=NPM[6]
26
27  X=1
28
29  for this in A,B,C,D,E,F,G:
30
31     if int(this)%2==0:
32         if int(this)==0:
33             this=""
34     print(this,end =" ")

```

```

1  # -*- coding: utf-8 -*-
2  """
3
4  @author: ASS
5  """
6
7  i=0
8  NPM = input("berapa npm kamu : ")
9  while i<1:
10     if len(NPM)<7:
11         print("Npmnya kurang dari 7 digit")
12         NPM = input("Npm kamu : ")
13     elif len(NPM)>7:
14         print("Npm lebih dari 7 digit")
15         NPM = input("Npm kamu: ")
16     else:
17         i=1

```

```

18
19 A=NPM[0]
20 B=NPM[1]
21 C=NPM[2]
22 D=NPM[3]
23 E=NPM[4]
24 F=NPM[5]
25 G=NPM[6]
26
27 X=1
28
29 for this in A,B,C,D,E,F,G:
30
31     if int(this)%2==1:
32         print(this,end=" ")

```

```

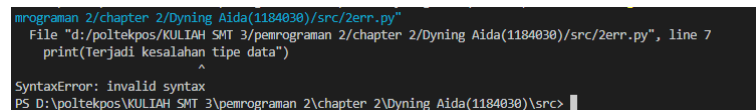
1 #-*- coding: utf-8 -*-
2 """
3
4 @author: ASS
5 """
6
7 i=0
8 NPM = input("berapa npm kamu : ")
9 while i<1:
10     if len(NPM)<7:
11         print("Npmnya kurang dari 7 digit")
12         NPM = input("Npm kamu : ")
13     elif len(NPM)>7:
14         print("Npm lebih dari 7 digit")
15         NPM = input("Npm kamu: ")
16     else :
17         i=1
18
19 A=NPM[0]
20 B=NPM[1]
21 C=NPM[2]
22 D=NPM[3]
23 E=NPM[4]
24 F=NPM[5]
25 G=NPM[6]
26
27 X=1
28 for X in A,B,C,D,E,F,G:
29     if int(X) > 1:
30         for i in range(2,int(X)):
31             if (int(X) % i) == 0:
32                 break
33         else :
34             print(int(X),end =""),

```

1.3 KETERAMPILAN PENANGANAN ERROR

1. peringatan error pada praktek kedua dan penjelasan penanganan error tersebut.

Peringatan error yang muncul salah satunya yaitu seperti gambar di bawah ini :



```
mrograman 2/chapter 2/Dyning Aida(1184030)/src/2err.py
File "d:/poltekpos/KULIAH SMT 3/pemrograman 2/chapter 2/Dyning Aida(1184030)/src/2err.py", line 7
    print(Terjadi kesalahan tipe data")
      ^
SyntaxError: invalid syntax
PS D:\poltekpos\KULIAH SMT 3\pemrograman 2\chapter 2\Dyning Aida(1184030)\src>
```

Figure 1.15: peringatan error

Cara penanganan yang digunakan untuk mengatasi error tersebut yaitu dengan merujuk ke baris/line ke-7 dan kemudian mengecek sintaks yang salah ataupun tidak sesuai dengan aturan penulisan pada python. Pada sintaks tersebut ditemukan kesalahan pada penulisan `print(Terjadi kesalahan tipe data")`. Seharusnya, penulisan yang benar untuk mencetak string adalah dengan menyertakan dua tanda kutip di antara kata atau kalimat yang akan dicetak. Sehingga menjadi `print("Terjadi kesalahan tipe data")`.

2. file 2err.py

```
1 a=" 35"
2 b=3
3
4 try :
5     a+b
6 except :
7     print(" Terjadi kesalahan penjumlahan tipe data")
```