

Tugas Chapter 2

Pemrograman II



Yusuf Jordan El Anwar

1184026

Applied Bachelor of Informatics Engineering

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

Chapter 1

Teori

1.1 Jenis-Jenis Variabel pada *Python*

Python memiliki 2 jenis variabel yaitu variabel global dan variabel lokal

1.1.1 Global

Variabel global merupakan variabel yang dapat digunakan atau dipanggil oleh semua fungsi. Meskipun variabel tersebut berada di file yang berbeda maupun fungsi yang berbeda. Pendeklarasian variabel global di tuliskan pada inisiasi awal file python atau fungsi main.

1.1.2 Lokal

Variabel lokal adalah sebuah variabel yang hanya dapat digunakan dan dipanggil dalam satu file saja .

1.2 Input dan Output

Input dimaksud disini adalah bagaimana caranya kita menuliskan kode yang membuat user harus menginputkan sebuah nilai yang mana nilai tersebut akan kita buat outputnya jika tidak di input maka outputnya juga kosong, seperti berikut listing 1.1

```
1 nama = input("Nama Anda: ")  
2  
3 print(nama)
```

Listing 1.1: Input dan Output

dari kode diatas variabel npm akan menyimpan sebuah input yang diisi oleh user, di variabel npm terdapat sebuah fungsi input() yang digunakan untuk menerima

input dari user, dan ada perintah `print(npm)` berfungsi agar hasil input dari user yang disimpan didalam variabel `npm` dapat di tampilkan ke layar user.

1.3 Contoh Pemakaian Operator Dasar Aritmatika dan Perubahan Tipe Data

Untuk pemakaian operator dasar aritmatika dapat dilihat pada listing 1.2

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7 string = "1"
8
9 tambah = int(string) + 1
10
11 kurang = 2 - 1
12
13 kali = 1 * 2
14
15 bagi = 6 / 3
16
17 strbagi = str(bagi)
18
19 print(tambah)
20 print(kurang)
21 print(kali)
22 print(bagi)
23
24 print(strbagi)
```

Listing 1.2: Operator Aritmatika

untuk mengubah dari tipe data string ke integer dapat menggunakan fungsi `int()` dan harus dipastikan didalam string tersebut harus berisi integer (angka) tidak boleh string yang lain. Seperti di listing 1.2, saya mencoba memasukkan karakter 1 kedalam variabel bernama `string` dan tipe datanya string, lalu saya ubah menjadi integer dengan cara `int(string)` sehingga program mendeteksi bahwa itu adalah integer bukan string dan dapat menjalankan perintah aritmatika tanpa error. Sebaliknya pun sama, dari integer ke dalam string seperti bagian variabel `bagi` yang isinya integer saya ubah ke string dengan menggunakan fungsi `str()`.

1.4 Perulangan

Untuk perintah perulangan pada python ada 3 yaitu **for**, **while**, dan **nested** penggunaan ketiganya memang berbeda tetapi untuk fungsinya tetap sama yaitu mengulang perintah yang berada di dalam sintaks looping dengan parameter tertentu untuk membuat looping tersebut keluar atau berhenti atau selesai.

1.4.1 *for*

For adalah perulangan yang mana parameter pengulangannya menggunakan list atau range, contoh pada listing 1.3

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

Listing 1.3: For Loop

program diatas akan mengeprint sebuah tulisan "Nilai ke-" dari 0 sampai dengan 5.

1.4.2 *while*

While akan mengulang terus menerus jika parameter yang dikembalikan bernilai true dan berakhir jika bernilai false, contoh pada listing 1.4

```
1 jawab = 'ya'
2 hitung = 0
3
4 while(jawab == 'ya'):
5     hitung += 1
6     jawab = input("Ulang lagi tidak? ")
7
8 print ("Total perulangan: " + str(hitung))
```

Listing 1.4: While Loop

pada kode program diatas akan mengulang terus menerus jika di jawab "ya" dan akan berhenti jika kita jawab "tidak" atau selain "ya".

1.4.3 *nested*

Nested merupakan sebuah pengulangan yang memungkinkan untuk memasukkan parameter pada sebuah pengulangan, contoh pada listing 1.5

```

1 a = 0
2
3 while a < 4:
4     print("WikWikWikWikWik")
5     a += 1

```

Listing 1.5: Nested Loop

pada kode program diatas kita melihat bahwa looping while yang sebelumnya menggunakan tipe data boolean sekarang bisa menggunakan sebuah parameter khusus yaitu nested sehingga kita bisa menentukan parameter yang diperlukan agar looping while dapat berhenti.

1.5 Percabangan

Percabangan merupakan algoritma yang menentukan sebuah paramater bernilai True atau False.

```

1 npm = input ("Masukan NPM Anda: ")
2
3 if npm != "":
4     print("NPM Terisi")
5     if npm == "1184026":
6         print("Kamu Jojo")
7     else:
8         print("Apakah Kamu Hacker?")
9 else:
10    print("Masak Lupa Sama NPM sendiri")

```

Listing 1.6: Percabangan If dan If Bersarang

program diatas merupakan program yang bisa di bilang bercabang, sehingga hanya bertemu dua jalan True atau False? jika kita lihat pada listing 1.6 maka pada percabangan pertama kita mengecek sebuah value dari npm apakah kosong atau berisi? jika kosong maka akan mengeprint **masak lupa sama npm sendiri** jika tidak kosong maka akan print **npm terisi** dan melanjutkan percabangan yang selanjutnya yaitu mengecek lagi, apakah isi dari npm itu sama dengan **"11840w6"**? jika sama maka akan mengeprint **kamu adalah jojo**, jika berbeda maka akan menampilkan pesan **Apakah kamu hacker?**

1.6 Kesalahan yang Sering Terjadi

Kesalahan yang sering terjadi dalam melakukan semua perintah diatas yaitu biasanya terjadi yaitu:

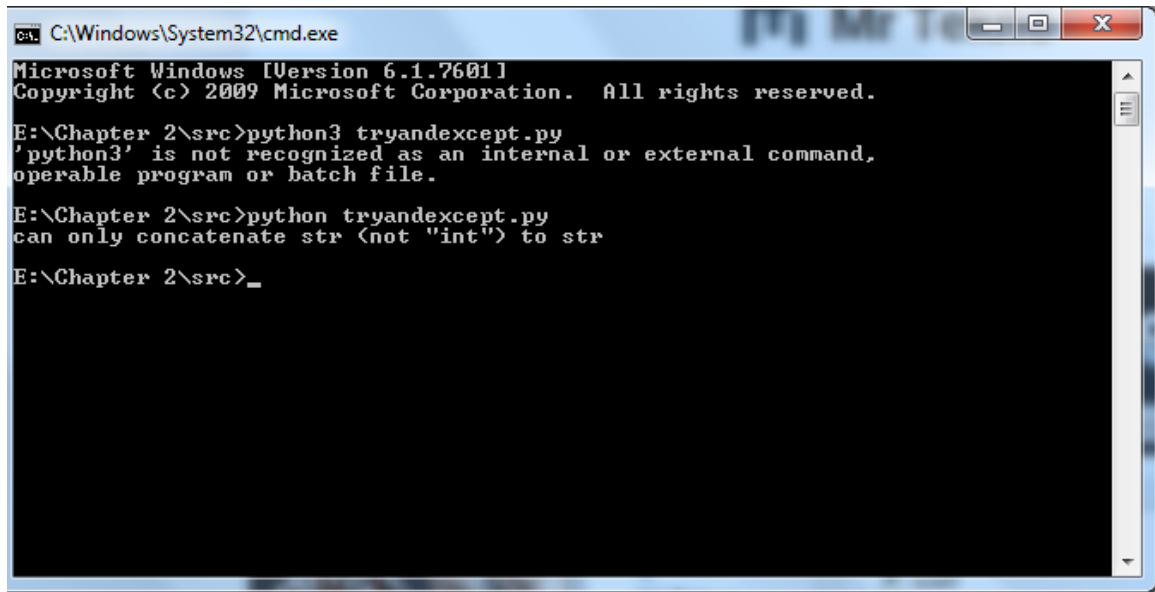
1. pertama, biasanya programmer sering salah atau lupa dalam menggunakan operator aritmatika. yaitu ketika memasukan sebuah variabel numerik tidak bisa di ditambah atau dikurang dengan variabel karakter begitu juga sebaliknya.
2. kedua, dalam perulangan biasanya programmer sering lupa dengan menggunakan perintah ***for i in ...*** pada ... biasanya programmer sering menggunakan angka integer, sehingga program akan error dan tidak bisa melooping sesuai harapan si developer, seharusnya untuk menggunakan looping dengan integer terkhusus untuk for yaitu menggunakan ***range***, seperti contoh pada listing 1.3
3. ketiga, dalam percabangan biasanya programmer suka salah dalam membandingkan 2 parameter nilai, yang satu variabel string atau karakter, yang satu lagi variabel numerik, sehingga program akan menuju perintah ***Else***.

1.7 *Try and Except*

Perintah ini digunakan untuk menangkap sebuah error, dan meneruskan program kita, sehingga program kita ketika terjadi error akan tetap berjalan dan tidak berhenti ditengah jalan. contoh kode bisa dilihat pada listing 1.7 dan hasilnya dapat dilihat pada gambar 1.1

```
1 try :  
2     hasil = "99" + 1  
3 except Exception as a:  
4     print (a)
```

Listing 1.7: Try and Except



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

E:\Chapter 2\src>python3 tryandexcept.py
'python3' is not recognized as an internal or external command,
operable program or batch file.

E:\Chapter 2\src>python tryandexcept.py
can only concatenate str (not "int") to str

E:\Chapter 2\src>_
```

Figure 1.1: Gambar Try and Except

pada program tersebut error menjelaskan bahwa **"99"** yang merupakan karakter hanya bisa di tambah oleh tipe data yang bertipe karakter juga, tidak bisa dengan integer, begitu juga sebaliknya.

Chapter 2

Praktek

2.1 Nomor 1

```
1 npm = input("Masukkan NPM anda: ")
2
3 nPm = int(npm)
4 print(nPm % 3)
5
6 print("###   ###   #####   ###   ###   #####   #####   #####")
7 print(" ")
8 print("###   ###   #####   ###   ###   #####   #####   #####")
9 print(" ")
10 print("###   ###   #####   #####   ###   ###   #####   #####")
11 print(" ")
12 print("###   ###   #####   #####   ###   ###   #####   #####")
13 print(" ")
14 print("###   ###   #####   #####   ###   ###   #####   #####")
15 print(" ")
```

Listing 2.1: Nomor 1

2.2 Nomor 2

```
1 npm = input("Masukan NPM Anda: ")
2
3 nPm = int(npm[-2:])
4
5 for a in range(nPm):
6     print("Halo, " + npm + " apa kabar?")
```

Listing 2.2: Nomor 2

2.3 Nomor 3

```
1 npm = input ("Masukan NPM Anda: ")
2
3 nPm = npm[-3:]
4
5 b = 0
6
7 for i in nPm:
8     c = int(i) + b
9     b = c
10
11 for i in range(b):
12     print ("Hallo, " + nPm + "Apa Kabar?")
```

Listing 2.3: Nomor 3

2.4 Nomor 4

```
1 npm = input ("Masukan NPM Anda: ")
2
3
4 nPm = npm[-3]
5
6 for i in nPm:
7     print("Halo, " + nPm + " apa kabar?")
```

Listing 2.4: Nomor 4

2.5 Nomor 5

```
1 npm = input ("Masukan NPM Anda: ")
2
3 string = "abcdefg"
4
5 index = 0
6
7 for i in string:
8     print(i + " = " + npm[index])
9     index += 1
```

Listing 2.5: Nomor 5

2.6 Nomor 6

```
1 npm = input ("Masukan NPM Anda: ")
2
3 a = 0
4 b = 0
5 for i in npm:
6     c = int(i) + b
7     b = c
```

```

8     a += 1
9
10 print(b)

```

Listing 2.6: Nomor 6

2.7 Nomor 7

```

1 npm = input ("Masukan NPM Anda: ")
2
3 a = 0
4 b = 0
5 for i in npm:
6     c = int(i) * b
7     b = c
8     a += 1
9
10 print(b)

```

Listing 2.7: Nomor 7

2.8 Nomor 8

```

1 npm = input ("Masukan NPM Anda: ")
2
3 for i in npm:
4     print(i)

```

Listing 2.8: Nomor 8

2.9 Nomor 9

```

1 npm = input("Masukan NPM Anda: ")
2
3 for i in npm:
4     a = int(i)
5
6     if a != 0:
7         if a % 2 == 0:
8             print(i, end=" ")

```

Listing 2.9: Nomor 9

2.10 Nomor 10

```

1 npm = input("Masukan NPM anda: ")
2
3 for i in npm:
4     a = int(i)
5
6     if a != 0:
7         if a % 2 == 1:

```

```
8 print(i, end="")
```

Listing 2.10: Nomor 10

2.11 Nomor 11

```
1 npm = input("Masukan NPM Anda: ")
2
3 for i in npm:
4
5     number = int(i)
6
7     if number > 1:
8
9         for i in range(2, number):
10             if (number % i) == 0:
11                 break
12     else:
13         print(number)
```

Listing 2.11: Nomor 11

Chapter 3

Penanganan Error

3.1 Error

Karena Saya mengerjakan di bantu oleh temen kos. Alhamdulillah gak ada eror.

3.2 2err.py

```
1 try:
2     string = "Bambang Tamvan"
3     integer = 1
4
5     hasil = string + integer
6 except:
7     print("Ngulang Algoritma lagi sana!!!")
```

Listing 3.1: 2err