Python (Chapter 2)

Tri Angga Dio Simamora

NIM : 1.18.4.047

Politeknik Pos Indonesia

Applied Bachelor Program of Informatics Engineering

Bandung 2019

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

Contents

List of Figures

Listings

Chapter 1

Teori

1.1 Jenis-Jenis Variabel pada *Python*

Jenis variabel pada python ada 2 yaitu variabel global dan variabel local

1.1.1 Global

Variabel global merupakan variabel yang dapat digunakan atau dipanggil di berbeda file ataupun fungsi biasanya variabel tersebut berapa dalam sebuah def fungsi() dan memiliki awalan ***self***

1.1.2 Lokal

Variabel lokal adalah sebuah variabel yang tercipta pada sebuah 1 file python dan hanya perintah dalam file yang sama yang dapat mengaksesnya.

1.2 Input dan Output

Input dimaksud disini adalah bagaimana caranya kita menuliskan kode yang membuat user menginputkan sebuah nilai yang mana nilai tersebut akan kita buat outputnya, seperti berikut listing 1.1

Listing 1.1: Input dan Output

print (npm)

terlihat dari kode diatas variabel npm akan menampung sebuah input yang diisi oleh user, di variabel npm ada fungsi input() yang digunakan untuk membuat kode python dalam menerima input dari user, dan ada perintah print(npm) yang berfungsi agar hasil input user yang ditampung oleh npm dapat di tampilkan ke layar user.

1.3 Contoh Pemakaian Operator Dasar Aritmatika dan Pengubahan Tipe Data

Untuk pemakaian operator dasar aritmatika dapat dilihat pada listing 1.2

Listing 1.2: Operator Aritmatika

```
string = "1"

tambah = int(string) + 1

kurang = 2 - 1

kali = 1 * 2

bagi = 6 / 3

strbagi = str(bagi)
```

untuk pengubahan dari tipe data string ke integer dapat menggunakan fungsi int() dan harus dipastikan didalam string tersebut harus berisi angka tidak boleh simbol, dan abjad. Seperti di listing 1.2, saya 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 integer bukan string dan dapat menjalankan perintah aritmatika tanpa error. Sebalikanya 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 berbeda tetapi fungsinya tetap sama yaitu mengulang perintah yang berada di dalam sintaks looping dengan parameter tertentu untuk membuat looping tersebut keluar atau out atau selesai.

1.4.1 *for*

For biasanya digunakan untuk perulangan yang mana parameter pengulangannya menggunakan list atau range, contoh pada listing 1.3

Listing 1.3: For Loop

```
for i in range (0,10):
        print(i)
```

program diatas akan mengeprint sebuah angka dari 0 sampai dengan 9.

1.4.2 *while*

While akan mengulang terus menerus jikalau parameter yang dikembalikan bernilai true, contoh pada listing 1.4

Listing 1.4: While Loop

```
crot = True
a = 0
while crot:
        print("ngocrot")
        if a == 3:
                crot = False
        a += 1
```

pada kode program diatas akan mengulang terus menerus hingga variabel crot berisi False jika False maka looping dari while ada berhenti.

1.4.3 *nested*

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

Listing 1.5: Nested Loop

```
a = 0

while a < 4:
        print("crot crot crot josss")
        a += 1
```

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 seperti apa yang diperlukan agar looping while berhenti.

1.5 Percabangan

Percabangan merupakan algoritma yang menentukan sebuah paramater apakah True apakah False.

Listing 1.6: Percabangan If dan If Bersarang

```
npm = "1184047"
```

```
if npm != "":
        print("tidak kosong")
        if npm == "1184047":
                print("kamu adalah crot")
        else:
                print("kamu bukanlah si crot")
else:
        print("kok kosong?")
```

program diatas merukapan 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 **kok kosong?** jika tidak kosong maka akan print **tidak kosong** dan melanjutkan percabangan yang selanjutkan yaitu mengecek lagi, apakah isi dari npm itu sama dengan **"1184047"**? jika sama maka akan mengeprint **kamu adalah crot**, jika berbeda maka akan menampilkan pesan **kamu bukanlah si crot**

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 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 berjalan terus dan tidak berhenti ditengah jalan contoh kode bisa dilihat pada listing 1.7 dan hasilnya dapat dilihat pada gambar 1.1

Listing 1.7: Try and Except

```
try:
        hasil = "1" + 1
except Exception as e:
        print(e)
```

Figure 1.1: Gambar Try and Except

pada program tersebut error menjelaskan bahwa **"1"** 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

Listing 2.1: Nomor 1

```
# nomor 1
print("Soal nomor 1")
nPm = int(npm)
print(nPm % 3)

print("### ### ######### ### ### ######### ### ### ########
print("### ### ######### ### ### ######### ### ### ########
print("### ### ### ### ### ### ### ### ### ### ### ### ### ##
print("### ### ### ### ### ### ### ### ### ### ### ### ### ###
print("### ### ######### ######### ### ### ######### ### ###"
print("### ### ### ### ### ### ### ### ### ### ### ### ### ###")
print("### ### ### ### ### ### ### ### ### ### ### ### ### ###")
print("### ### ######### ### ######### ### ### ### ###")
print("### ### ######### ### ######### ### ### ### ###")
```

2.2 Nomor 2

Listing 2.2: Nomor 2

```
# nomor 2
print("soal nomor 2")

nPm = int(npm[-2:])

for i in range(nPm):
    print("Halo, " + npm + " apa kabar?")
```

2.3 Nomor 3

Listing 2.3: Nomor 3

```
# nomor 3
print("soal nomor 3")

nPm = npm[-3:]

b = 0

for i in nPm:
    c = int(i) + b
    b = c

for i in range(b):
    print("Halo, " + nPm + " apa kabar?")
```

2.4 Nomor 4

Listing 2.4: Nomor 4

```
# nomor 4
print("soal nomor 4")

nPm = npm[-3]

for i in nPm:
    print("Halo, " + nPm + " apa kabar?")
```

2.5 Nomor 5

Listing 2.5: Nomor 5

```
# nomor 5
print("soal nomor 5")

string = "abcdefg"

index = 0

for i in string:
    print(i + " = " + npm[index])
    index += 1
```

2.6 Nomor 6

Listing 2.6: Nomor 6

```
# nomor 6
```

```
print("soal nomor 6")

a = 0
b = 0
for i in npm:
    c = int(i) + b
    b = c
    a += 1

print(b)
```

2.7 Nomor 7

Listing 2.7: Nomor 7

```
# nomor 7
print("soal nomor 7")

a = 0
b = 0
for i in npm:
    c = int(i) * b
    b = c
    a += 1

print(b)
```

2.8 Nomor 8

Listing 2.8: Nomor 8

```
# nomor 8
print("soal nomor 8")

for i in npm:
    print(i)
```

2.9 Nomor 9

Listing 2.9: Nomor 9

```
# nomor 9
print("soal nomor 9")

for i in npm:
    a = int(i)
```

```
    if a != 0:
        if a % 2 == 0:
            print(i, end="")
```

2.10 Nomor 10

Listing 2.10: Nomor 10

```
# nomor 10
print("soal nomor 10")

for i in npm:
    a = int(i)

    if a != 0:
        if a % 2 == 1:
            print(i, end="")
```

2.11 Nomor 11

Listing 2.11: Nomor 11

```
# nomor 11
print("soal nomor 11")
for i in npm:

    number = int(i)

    if number > 1:

        for i in range(2, number):
            if (number % i) == 0:
                break
        else:
            print(number)
```

Chapter 3

Penanganan Error

3.1 Error

Alhamdulillah saya mengerjakan tugas ini ga ada yang error :D

3.2 2err.py

Listing 3.1: 2err

```
try:
        string  = "1"
        integer = 1

        hasil = string + integer
except:
        print("String dan Integer gak bisa ditambah croooottt ....")
```