

Tugas Pemrograman Chapter 2

Dian Markuci(1184095)

25 October 2019

1 TEORI

1.1 Variable

Variabel merupakan suatu tempat untuk menampung value dimemori, ibarat sebuah ruangan, variabel dibagi menjadi dua berdasar ruang lingkup yaitu variabel global dan lokal, untuk menentukan variabel global atau lokal, tergantung dari tempat dideklarasikannya variabel pada program yang sedang dibangun. Variabel global adalah variabel yang bisa diakses di semua lingkup dalam program yang sedang dibuat, atau disebut variabel global ini bisa dikenali oleh semua fungsi dan prosedur, sementara variabel lokal adalah variabel yang dapat diakses hanya di lingkup khusus, atau variabel lokal ini hanya bisa diakses pada fungsi/prosedur dimana variabel itu dideklarasikan.

1.2 Input dan Output User

- Input menggunakan fungsi `input ()`
- Python memiliki fungsi `input` yang memungkinkan Anda meminta input teks kepada pengguna. Anda memanggil fungsi ini untuk memberi tahu program untuk berhenti dan menunggu pengguna memasukkan data
- Pada Python 2, Anda memiliki fungsi built-in `raw_input ()`
- Pada Python 3, Anda memiliki `input ()`.

1.3 Operator 2

1.3.1 Operasi Aritmatika

Operator	Simbol
Pembagian	/
Perkalian	*
Penjumlahan	+
Pengurangan	-
Modulus	%
Pangkat	**

Operasi matematika sebagai bahasa pemrograman, Python memiliki operasi aritmatika seperti tambah, kurang, kali, bagi. Berikut adalah contoh penggunaan operasi aritmatika pada python Contoh misalnya kita mempunyai variable : a=6 dan b=4

1.3.2 Casting

Lalu apa yang akan terjadi bila ternyata variabel a adalah string dan variabel b adalah integer, contoh a="6" dan b=4, tentunya program akan error bukan? Disinilah casting digunakan. Casting adalah cara untuk mengubah tipe data dari suatu data primitive, Jadi misal kita akan menjumlahkan variabel a dan b seperti contoh diatas tetapi logikanya sebuah kata (string) tidak akan bisa dijumlahkan dengan angka ("6" + 4) karena variabel a diapit oleh tanda kutip, ini berarti variabel a bertipe data string untuk itu kita perlu merubah dulu variabel a yang tadinya string menjadi integer. Berikut adalah syntax untuk melakukan casting :

- int(var/value) : mengubah tipe data ke integer, contoh int(angka)
- float(var/value) : mengubah tipe data ke float, contoh float(hasil)
- string(var/value) : mengubah tipe data ke str, contoh string(12)

Mengubah string ke integer : type data string harus dilakukan casting dengan "int(variable)". Mengubah integer ke string : type data integer harus dilakukan casting dengan "str(variable)".

1.4 Syntax Perulangan

Di dalam bahasa pemrograman Python pengulangan dibagi menjadi 3 bagian

- While Loop
- For Loop

- Nested Loop

While Loop

While : untuk melakukan looping yang tidak pasti

Contoh :

```
i = 0
while True :
    if i < 100:
        print ("i bernilai : "), i
        i = i + 1
    elif i >= 100:
        break
```

For Loop

For : Melakukan looping yang sudah pasti jumlahnya

Contoh :

```
for i in range(0, 100):
    print (i)
```

Nested Loop

Bahasa pemrograman Python memungkinkan penggunaan satu lingkaran di dalam loop lain. Dibawah ini menunjukkan beberapa contoh untuk menggambarkan konsep tersebut

Contoh penggunaan Nested Loop

```
i = 2
while(i < 100):
    j = 2
    while(j <= (i/j)):
        if not(ij = j + 1)
        if (j > i/j) : print(i, " is prime")
        i = i + 1

    print "Good bye!"
```

1.5 Syntax Kondisi

Struktur if

Sederhananya struktur if dalam Python dijalankan untuk mengecek kondisi ini bernilai benar atau salah. Apabila kondisi bernilai benar, maka python akan menjalankan statement dalam blok kondisi tersebut dan sebaliknya jika kondisi bernilai salah maka statement dalam blok tersebut tidak akan dijalankan.

Struktur if – else

Struktur if sebelumnya hanya menjalankan statement dalam blok kondisi jika bernilai benar, maka struktur if-else adalah membuat statement kondisi yang bernilai benar dan salah.

If : digunakan untuk percabangan

Contoh :

```
umur = 20
if umur >17:
    print("Remaja")
```

Else

Else : jika kondisi if tidak terpenuhi maka dijalankan kondisi else

Contoh :

```
umur = 6
if umur >17:
    print ("beranjak dewasa")
else:
    print ("anak-anak")
```

1.6 Try Except

Menangani Eksepsi Menggunakan Try, Except, dan Finally

Terjadinya eksepsi pada suatu program bisa membuat program berhenti. Untuk mencegahnya, Maka kita harus mengantisipasi hal tersebut.

Python menyediakan metode penanganan eksepsi dengan menggunakan pernyataan *try dan except*.

Dalam blok try kita akan meletakkan baris program yang mungkin akan terjadi error. Apabila terjadi error, maka cara penanganannya diserahkan kepada blok except.

contoh try...finally untuk mengoperasikan file. try:

```
f = open("C:test.txt")
melakukan operasi terhadap file
finally:
f.close()
```