Tugas Pemrograman Chapter 2

Dyah Ayu Anandra(1184098)

8 Desember 2019

1 TEORI

1.1 Variable

Variabel merupakan tempat di memori yang digunakan untuk menyimpan nilai. Saat kita akan membuat sebuah variabel, kita dapat 'memesan' tempat di dalam memori. Variabel nilainya dapat berubah-ubah atau mutable.Tempat di dalam memori dapat diisi dengan data atau objek, baik itu bilangan bulat (integer), pecahan (float), karakter (string), dan lain – lain.

1.2 Input dan Output User

Input adalah masukan perintah yang kita berikan ke program setelah itu perintah akan diproses dan menampilkan hasil outputnya.

Cara Mengambil Input dari Keyboard, Python menyediakan fungsi input() dan rawinput() agar dapat mengambil inputan dari keyboard. Fungsi input() digunakan untuk mengambil data angka. Sedangkan rawinput() untuk mengambil teks. Sedangkan pada Python3 cukup menggunakan fungsi input() saja, karena fungsi rawinput() sudah digabungkan di sana.

1.3 Operator 2

1.3.1 Operasi Aritmatika

Operator	penjelasan
Pembagian (/)	Untuk membagi nilai di sebelah kiri menggunakan nilai yang ada di sebelah kanan
Perkalian (*)	Mengalikan bilangan
Penjumlahan (+)	Menjumlahkan nilai dari masing-masing bilangan
Pengurangan (-)	Mengurangi nilai di sebelah kiri menggunakan nilai yang ada di sebelah kanan

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=8 dan b=4

1.4 Syntax Perulangan

Di dalam bahasa pemrograman Python pengulangan dibagi menjadi 3 bagian

- While
- For
- Nested

While
While : untuk melakukan pengulangan yang tidak pasti, Dengan while melakukan perulangan, lalu menambahkan satu variabel hitung setiap kali pengulangannya. kemudian menanyakan ke pengguna, apakah ingin memberhentikan Pengulangan atau tidak?
Contoh :
i = 0
while True :
if i <200:
print ("i bernilai : "), i
i = i + 1
elif i >= 200:
break

For Loop
For : Melakukan looping yang sudah pasti jumlahnya
Contoh :
for i in range(0, 200):
print (i)

For

Pertama kita menentukan banyak perulangannya sebanyak 10x. Variabel x berfungsi untuk menampung suatu indeks, dan fungsi untuk membuat list dengan range dari 0-10 menggunakan range(). Fungsi untuk merubah tipe data ineger ke string menggunakan str(). contoh penggunaan For i = ['kopi','nasi','teh','jeruk']
for isi in i: print isi

Nested

Bahasa pemrograman Python memungkinkan penggunaan satu lingkaran di dalam loop lain. Dibawah ini menunjukkan beberapa contoh untuk menggambarkan konsep tersebut
Contoh penggunaan Nested
i = 2
while(i <200):
j = 2

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

print "Selamat Datang!"

1.5 Syntax Kondisi

Struktur if

Percabangan ini hanya memiliki satu pilihan. pilihan di dalam IF hanya akan dikerjakan atau dilakukan apabila kondisinya benar. Namun jika salah tidak akan melakukan apa-apa. Alias lanjut eksekusi ke perintah berikutnya.

Struktur if – else

SUntuk percabangan IF/ELSE mempunyai pilihan alternatif apabila kondisinya salah. IF: "Apabila benar maka akan dikerjakan, apabila salah tidak dapat dikerjakan dan tidak dapat lanjut ke tahap berikutnya", IF/ESLE: "Jika kondisi dalam keadaan benar maka akan dikerjakan, jika dalam keadaan salah maka mengerjakan yang lainnya, setelah itu baru lanjut".

1.6 Try Except

Menangani Eksepsi Menggunakan Try, Except, dan Finally

Dengan menggunakan statement try.except dapat menangani bentuk error di Python. mendeteksi error dengan memanfaatkan kondisional biasa menggunakan if.else, namun itu akan lebih praktis apabila ditangani dengan menggunakan try.except. mengurung suatu blok kode dengan try.except untuk dapat menangani error yang mungkin saja tidak diketahui. try.except biasanya digunakan untuk menangani error saat penggunaan IO, operasi database, atau pengaksesan indeks suatu list atau dictionary, dan berbagai kasus lainnya.Sekarang kita dapat mengenal beberapa kasus sederhana yang menggunakan try.except.

Menangani error pembagian nol Misalkan dalam kode berikut terjadi pembagian yang membagi suatu angka dengan nol. telah menjadi ketentuan jika sebuah angka dibagi nol maka program akan error. Maka dari itu kita dapat mengurung dengan try..except, kemudian kita keluarkan error-nya begitu error tertangkap oleh except. Baris "print x+1" akan dieksekusi dan hanya akan muncul error yang ditimbulkan oleh pembagian nol.

Di dalam kode ini dapat mencoba menangkap dua error pada kode yang telah dikurung oleh try.except. ada sebuah dictionary berisi key nama, kota, dan umur. Kemudian kita membuka sebuah file yang bernama contact.txt.

Meskipun terdapat kode yang error setelahnya yang dapat mengakibatkan error dalam pengaksesan indeks, yang akan ditangkap terlebih dahulu adalah error yang diakibatkan gagalnya membaca file.

Mengenal "finally" Mungkin kita juga membutuhkan sebuah penanganan khusus apabila terjadi suatu error ataupun tidak pasti harus ada kode yang dieksekusi. Misal Anda ingin memutus koneksi kepada server jika terdapat error karena request yang overload atau ada struktur folder yang berubah. Sebagai contoh dapat melihat bagaimana statement finally bekerja pada try.except.