

1 Teori

1.1 Jenis-Jenis Variable

1. Bilangan (Number) Tipe data bilangan yang umum ada 2 yaitu, integer dan float. Integer adalah bilangan bulat, sedangkan float adalah bilangan pecahan. Selain itu ada tipe bilangan lain, yaitu bilangan kompleks yaitu bilangan yang memiliki bagian real dan imajiner. Integer, float, dan kompleks masing-masing di Python diwakili oleh kelas int, float, dan complex.

2. String String adalah satu atau serangkaian karakter yang diletakkan di antara tanda kutip, baik tanda kutip tunggal (') maupun ganda ("). Huruf, angka, maupun karakter lainnya yang digabung menjadi teks adalah contoh string.

String adalah tipe data yang anggotanya berurutan dan memiliki indeks. Indeks dimulai dari angka 0 bila dimulai dari depan dan -1 bila diindeks dari belakang. Tiap karakter bisa diakses menggunakan indeksnya dengan format namastring[indeks] . Pada string juga bisa dilakukan slicing atau mengakses sekelompok substring dengan format namastring[awal:akhir]

3. List List adalah tipe data yang berisi item yang berurutan. Seperti halnya tipe data string, tiap item (anggota) list memiliki indeks sesuai dengan urutannya. Indeks dimulai dari 0 dan bukan dari 1.

List bisa berisi anggota dengan tipe yang sama maupun berbeda. Untuk mendeklarasikan list, digunakan tanda kurung [] dan masing-masing anggotanya dipisahkan oleh tanda koma.

4. Tuple Tuple adalah jenis data lain yang mirip dengan list. Perbedaannya dengan list adalah anggotanya tidak bisa diubah (immutable). List bersifat mutable, sedangkan tuple bersifat immutable. Sekali tuple dibuat, maka isinya tidak bisa dimodifikasi lagi.

Tuple dideklarasikan dengan menggunakan tanda kurung . dan anggotanya dipisahkan oleh tanda koma. Tuple berguna untuk data yang dimaksudkan tidak diubah isinya. Misalnya tuple komposisi warna untuk putih adalah (255,255,255).

5. Set Set adalah salah satu tipe data di Python yang tidak berurutan (unordered). Set memiliki anggota yang unik (tidak ada duplikasi). Jadi misalnya kalau kita meletakkan dua anggota yang sama di dalam set, maka otomatis set akan menghilangkan yang salah satunya.

Set dibuat dengan meletakkan anggota – anggotanya di dalam tanda kurung kurawal , dipisahkan menggunakan tanda koma. Kita juga bisa membuat set dari list dengan memasukkan list ke dalam fungsi set()

6. Dictionary Dictionary adalah tipe data yang tiap anggotanya terdiri dari pasangan kunci-nilai (key-value). Mirip dengan kamus dimana ada kata ada arti. Dictionary umumnya dipakai untuk data yang besar dan untuk mengakses anggota data secara acak. Anggota dictionary tidak memiliki indeks.

Dictionary dideklarasikan dengan menggunakan tanda kurung kurawal , dimana anggotanya memiliki bentuk kunci:nilai atau key:value dan tiap anggota dipisah tanda koma. Kunci dan nilainya bisa memiliki tipe sembarang.

1.2 Cara Menampilkan Kode untuk meminta Input dan output nya

Untuk menampilkan Python sudah menyediakan fungsi input() dan rawinput() untuk mengambil inputan dari keyboard. Cara pakainya: Namavariabel = input (“Sebuah teks”) Artinya, teks yang kita inputkan dari keyboard akan disimpan ke dalam namavariabel. Untuk menampilkan output teks, kita menggunakan fungsi print().

1.3 Operator Dasar Aritmatika

1. Penjumlahan +
2. Pengurangan -
3. Perkalian *
4. Pembagian /
5. Sisa Bagi percent
6. Pemangkatan **

Integer = int(a) untuk konversi string ke integer

String = str(a) untuk konversi integer ke string

1.4 Sintax Perulangan

1. The while Loop

Dengan while loop, kita dapat menjalankan serangkaian pernyataan selama suatu kondisi benar.

Contoh : Print i sepanjang i adalah kurang dari 6:

```
i = 1
```

```
while i < 6:
    print(i)
    i += 1
```

2. Python For Loops

For loop digunakan untuk mengulangi urutan (baik daftar, tuple, dictionary, set, atau string).

Ini kurang seperti kata kunci untuk dalam bahasa pemrograman lain, dan berfungsi lebih seperti metode iterator seperti yang ditemukan dalam bahasa pemrograman berorientasi objek lainnya. Dengan for loop kita dapat mengeksekusi seperangkat pernyataan, satu kali untuk setiap item dalam list, tuple, set dll.

Contoh :

Print each fruit in a fruit list:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

1.5 Syntax Kondisi

Kondisi If Else

Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai. Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif. Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar.

Kondisi if else adalah kondisi dimana jika pernyataan benar true maka kode dalam if akan dieksekusi, tetapi jika bernilai salah false maka akan mengeksekusi kode di dalam else.

Dibawah ini adalah contoh penggunaan kondisi if else pada Python

```
nilai = 3
if (nilai < 7):
    print ("Selamat anda lulus")
else :
    print ("Maaf anda tidak lulus")
```

1.6 Jenis error di Python

1. Syntax Errors

Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Python interpreter dapat mendeteksi kesalahan ini saat kode dieksekusi.

```
print "Hello World"
```

```
SyntaxError: invalid syntax
```

Output dari program yang dieksekusi akan menampilkan pesan “invalid syntax”. Penanganan Syntax Errors dilakukan dengan memperbaiki penulisan kode yang salah tersebut.

untuk menanganinya cukup tambahkan tanda kurung () pada :

```
print ("Hello World")
```

2. Exceptions

Exceptions adalah suatu keadaan saat penulisan syntax sudah benar, namun kesalahan terjadi karena syntax tidak bisa dieksekusi. Banyak hal yang menyebabkan terjadinya Exceptions, mulai dari kesalahan matematika, kesalahan nama function, kesalahan library, dan lain-lain.

1.7 Cara menggunakan Try Except

Di kode ini kita akan mencoba menangkap dua error pada kode yang dikurung oleh try..except. Terdapat sebuah dictionary yang berisi key nama, kota, dan umur. Kemudian kita membuka sebuah file yang bernama contact.txt. Walaupun ada kode error setelahnya yang akan mengakibatkan error pengaksesan indeks, yang akan ditangkap terlebih dahulu adalah error yang diakibatkan gagalnya membaca file.

contoh :

```
orang = {"nama": "syuaib", "kota": "jepara", "umur": "20"}
```

```
try:
```

```
contact = open("contact.txt", 'r')
```

```
print orang["pekerjaan"]
```

```
except IOError, e:
```

```
print "Terjadi error IO: ", e
```

```
except KeyError, e:
```

```
print "Terjadi kesalahan pada akses list/dict/
```

```
tuple:", e
```

```
print orang
```