MODUL PEMROGRAMAN PYTHON

Pengenalan Pemrograman Python Bagian 1

oleh:

Vicky Vernando Dasta

vicky.vernando@student.unri.ac.id

1. Pengenalan

Pemrograman tingkat tinggi menekankan pada produktifitas. Python awalnya digunakan sebagai bahasa scripting layaknya Bash pada `*nix` platform, seiring berkembangnya Python, penggunaan Python tidak hanya pada bahasa scripting pengganti bash, namun sebagai bahasa pemrograman tingkat tinggi yang mampu melakukan hampir semua kebutuhan komputasi seperti simulasi saintifik, pemrosesan data dan kebutuhan komputasi lain. Pada modul ini akan dijelaskan mengenai penggunaan Python dari dasar sampai pembuatan aplikasi yang bisa langsung dipakai. Python menggunakan dua konsep dalam bahasa pemrograman, interpreted language dan compiled language. Untuk menjalankan script `*.py` diperlukan sebuah interpreter yang bisa diunduh di http://www.python.org.

2. Penggunaan

Untuk pengajaran Python, akan digunakan Google Colab, platform ini dipilih karena tidak diperlukan instalasi lokal sehingga lebih efisien dalam segi waktu. Sedangkan untuk penggunaan python secara lokal *(offline)*, bisa mengikuti tutorial instalasi Python di http://www.python.org. Pada kelas ini, kita akan menggunakan Python versi 3.x (>3.4) lebih baik, karena terdapat beberapa perbaikan dan pembaruan fitur yang lebih baik untuk performa program.

3. Google Colab

Google Colab adalah platform yang ditujukan pada researcher/peminat machine learning, platform ini berupa Jupyter Notebook yang sudah dibekali *instance Graphic Processing Unit* (GPU) yang diberikan oleh Google .Inc secara cuma-cuma.

Berikut cara penggunaan Google Colab:

1. Buka http://colab.research.google.com
2. Masuk dengan akun google anda

3. Pilih *create new notebook Python 3* untuk memulai notebook dengan kernel Python versi 3.x

Pada kelas ini, sudah disiapkan notebook untuk latihan yang bisa dikunjungi di link https://colab.research.google.com/drive/1bSxSkzkA6rt60llsqx2CuoG9a9NoLcsA.

4. Pengenalan bahasa pemrograman Python

a. Python keywords

> print, if, else, elif, and, or, while, for, def, lambda, import, from, as, continue, break, try, except, with, list, dict, zip, set, class, in

Keyword di atas tidak dapat dijadikan sebagai nama untuk variabel di Python, karena keyword di atas sudah *reserved* sebagai *keyword/grammar* Python. Untuk penamaan variabel yang memiliki nama yang sama dengan *reserved keyword*, disarankan menggunakan *underscore*, contoh: `__list__`.

b. Indentation/scoping

Indentasi digunakan untuk memisahkan satu statement dengan statement yang lain. Pada bahasa C/C++, satu statement dan statement lainnya dipisahkan dengan curly bracket `{ }`, di Python, pemisahan statement digunakan whitespace. Aturannya, Untuk satu statement dengan statement berikutnya, dipisahkan dengan 4 spasi atau 1 tab. Konsistensi diperlukan dalam penggunaan spasi dan tab dalam sebuah script, bila ingin menggunakan spasi, maka untuk seterusnya gunakan spasi, bila keduanya digunakan, maka intepreter Python akan mengeluarkan error `IndentationError`.

contoh pada javascript:

```
for(var i=0; i < 10; i++){
   // lakukan sesuatu di sini
}
```

di Python, curly bracket pada for diganti dengan whitespace.

```
for i in range(10):
   // lakukan sesuatu di sini
```

c. Hello, World!

Pada Python 2.x, print (`std out`), menggunakan keyword `print`, sedangkan pada Python 3.x, `print` dijadikan sebagai fungsi `print(hello, world)`. Untuk

menampilkan `Hello, world!` di layar, buka REPL/IDLE Python, kemudian ketikkan perintah:

```
print('Hello, world!')
```

d. Variable

Variable adalah suatu cara untuk menyimpan sebuah nilai. Dalam membuat sebuah variabel di Python tidak diperlukan penulisan tipe data dari variabel secara implisit.

misal:

```
phi = 3.14
```

atau:

```
c = 3e+8
```

5. Tipe dan struktur data

a. Integer

Bilangan bulat. Panjang integer pada Python tergantung dari tipe prosesor dari sistem. Untuk sistem dengan prosesor 32-bit, maka banyak digit yang bisa disimpan adalah sebanyak 2e+31 dan 2e+63 digit untuk 64-bit.

contoh: `num = 1337`

b. Boolean

Boolean merupakan representasi dari nilai kebenaran, berisi `True` atau `False`

contoh: `truth_val = True if a == 1 else False`

c. String

String merupakan sebuah urutan nilai yang merupakan representasi dari sebuah nilai unicode. String pada Python diapit oleh satu quote atau lebih.

contoh:

```
welcome_message = Hello {}!

print(welcome_message.format(Guido))
```

d. Float

Bilangan berkoma (floating point)

contoh:

```
phi = 3.14
```

e. List

List merupakan kumpulan dari data.

contoh:

```
nums = [1, 2, 3, 4, 5]
```

Untuk mengakses item pada list `num`, digunakan bracket `[indeks]` dan indeks dari item> yang ingin diakses. Misal, bila ingin mengakses item di indeks 1, maka kita gunakan:

`nums[1]`.

Berikut adalah operasi yang bisa dilakukan dengan list:

- `append`

  untuk menambahkan item baru, digunakan method `append`

  contoh:

  `nums.append(6)`

- `remove(item)`

  untuk menghapus item dari sebuah list

  contoh:

  `nums.remove(6)`

- `count(item)`

  untuk mengetahui berapa banyak item `item` di dalam list

  contoh:

  `nums.count(1)`

- `pop([n])`

  Fungsi `pop` digunakan untuk menghapus item di indeks terakhir pada list bila index tidak ditentukan secara langsung, bila indeks ditentukan, maka item di index `n` akan dihapus dari list. Pada `nums`, bila method pop dioperasikan/dipanggil maka item 5 akan dihapus dari list `nums`.

contoh:

`nums.pop()`

f. Dictionary

Dictionary atau bahasa lainnya named-list merupakan sebuah tipe data yang mirip JSON, tipe data ini memiliki *key* dan *value* dimana setiap *key* tidak boleh redundant.

contoh:

```
user_data = {name: guido, password: rossum}
```

Untuk mengakses item dengan key `name`, digunakan braket seperti di list, bedanya pada dictionary, **di dalam braket bukan indeks item, tapi key dari item tersebut**.

contoh:

`user_data[name` # guido

‘user_data[‘password’] # rossum

g. Tupple

Tupple adalah bentuk lain dari list. Tidak seperti list yang bersifat mutable, tupple bersifat immutable, artinya, isi dari sebuah tupple tidak bisa ditambah atau dihapus setelah tupple tersebut dibuat.

contoh:

```
nums = (1, 2, 3, 4, 5)
```

Untuk mengakses item di dalam nums, bisa digunakan operasi seperti di list, namun bedanya pada tupple, tidak ada operasi `append` dan `remove`.

h. Set

Set adalah struktur data di python yang mengeliminasi item yang sama, artinya, bila terdapat item yang sama di dalam sebuah list, maka item-item yang sama akan dihapus dari list tersebut, sehingga akhirnya hanya tersisa 1 item yang berbeda satu sama lain.

6. Operasi dan ekspresi

Untuk operasi di Python, terdapat beberapa seperti:

- tambah: +

  contoh:

  `1 + 1`

- kurang: `-`

  contoh:

  `2 - 1`

- kali: ‘*

  contoh:

  2*1

- bagi: `/`

  contoh:

  `3/2`

- modulus: `%`

  contoh:

  `4%2`

- pangkat: `**`

  contoh:

  `2**2`

- kecil dari: `<`

  contoh:

  `1 < 3`

- besar dari: `>`

  contoh:

  `3 > 0`

- besar atau sama dengan: `>=`

  contoh:

  `3 >= 1`

- kecil atau sama dengan: `<=`

contoh:

`1 <= 3`

- equals to: `==`

  contoh:

  `1 == 1`

7. Kontrol alir

a. Kondisi dengan satu keadaan

Kontrol kondisi berisi situasi atau kondisi yang mana hanya akan menjalankan sebuah hanya bila salah satu kondisinya bernilai benar.

misal:

```
a = 1
b = 3

if a < b: # benar bahwa a kecil daripada b

   # lakukan sesuatu disini
```

b. Kondisi dengan dua keadaaan

Pola ini digunakan bila terdapat dua keadaan dan program akan melakukan sebuah operasi bila salah satu keadaan bernilai benar.

```
a = 1
b = 2

if a == b:
   # lakukan sesuatu

else:
   # lakukan sesuatu
```

c. Kondisi dengan tiga keadaan

Pola ini digunakan bila terdapat 3 keadaaan, dan program hanya akan melakukan sebuah operasi bila salah satu keadaan bernilai benar.

```
a = 1
b = 2

if a == b:
   # lakukan sesuatu

elif a > b:
   # lakukan sesuatu

else:
   # lakukan sesuatu
```

8. Perulangan

a. Perulangan dengan for

`For` digunakan untuk melakukan perulangan pada objek yang *iterable* iterable maksudnya, objek tersebut berupa kumpulan/sequence, misalnya list, string, dictionary, tupple.

looping pada `range()`

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

looping pada string

```
nama = "guido van rossum"

for item in nama:

   print(item)
```

b. Perulangan dengan while

`while` merupakan perulangan yang hanya berjalan dan berhenti bila sebuah kondisi terpenuhi.

```
a = 10

while a < 20:

   print(a)
   a += 1
```

Untuk menghentikan perulangan secara hard bisa menggunakan statement `break`, misal:

```
for i in range(10):
   if i%2 == 0:
      break

   print(i)
```

Statement `continue` adalah kebalikan dari `break`.

9. Fungsi

Fungsi atau rutin adalah sebuah bagian dari program yang mengerjakan operasi spesifik. Pada Python digunakan keyword `def` dan `lambda` untuk membuat sebuah fungsi.

struktur fungsi:

- def:

```
def namafungsi(param1, param2, ..., paramN):
   # lakukan sesuatu terhadap params
   return sesuatu
```

- lambda:

```
x = lambda x: x*12
print(x(12)) # 144
```