

Hello, Python!

Ali Akbar Septiandri

Universitas Al-Azhar Indonesia

aliakbars@live.com

February 21, 2017

Overview

- 1 Administrasi
- 2 Pendahuluan
 - Hello, World!
 - Manipulasi Variabel
- 3 Control Flow & Looping
 - If-Else
 - For & While
- 4 Struktur Data Standar
- 5 Fungsi
 - Penulisan Fungsi
 - Fungsi Bawaan Python

Administrasi

- 1 Lihat situs <http://uai.aliakbars.com/python.html>
- 2 Kuliah pukul 07.30-10.00 setiap hari Selasa
- 3 Terdapat 4 tugas, dimulai dari pertemuan keempat
- 4 Ujian Tengah Semester
- 5 Proyek akhir
- 6 Komponen penilaian: 40% tugas, 30% UTS, 30% proyek akhir

Tentang Mata Kuliah Ini

- 1 Secara umum akan menggunakan Python 2.7

Tentang Mata Kuliah Ini

- 1 Secara umum akan menggunakan Python 2.7
- 2 Diasumsikan sudah familiar dengan pengembangan perangkat lunak secara umum

Tentang Mata Kuliah Ini

- 1 Secara umum akan menggunakan Python 2.7
- 2 Diasumsikan sudah familiar dengan pengembangan perangkat lunak secara umum
- 3 Berfokus pada pengenalan pustaka (*library*) dan kakas (*tools*)

Tentang Mata Kuliah Ini

- 1 Secara umum akan menggunakan Python 2.7
- 2 Diasumsikan sudah familiar dengan pengembangan perangkat lunak secara umum
- 3 Berfokus pada pengenalan pustaka (*library*) dan kakas (*tools*)
- 4 Sangat disarankan menggunakan Linux atau Mac (*virtual image* akan disediakan)

Tentang Mata Kuliah Ini

- 1 Secara umum akan menggunakan Python 2.7
- 2 Diasumsikan sudah familiar dengan pengembangan perangkat lunak secara umum
- 3 Berfokus pada pengenalan pustaka (*library*) dan kakas (*tools*)
- 4 Sangat disarankan menggunakan Linux atau Mac (*virtual image* akan disediakan)
- 5 Terdapat pengenalan terhadap beberapa materi yang membutuhkan keahlian khusus, e.g. AI, NLP, ML

Tentang Mata Kuliah Ini

- 1 Secara umum akan menggunakan Python 2.7
- 2 Diasumsikan sudah familiar dengan pengembangan perangkat lunak secara umum
- 3 Berfokus pada pengenalan pustaka (*library*) dan kakas (*tools*)
- 4 Sangat disarankan menggunakan Linux atau Mac (*virtual image* akan disediakan)
- 5 Terdapat pengenalan terhadap beberapa materi yang membutuhkan keahlian khusus, e.g. AI, NLP, ML
- 6 Tugas bersifat individual, jadi silakan berdiskusi, tapi **dilarang keras** menyalin kode teman

Prerequisites

- 1 Dasar Pemrograman
- 2 Struktur Data
- 3 Pemrograman Berorientasi Objek
- 4 Basisdata
- 5 E-commerce & Web Development

Beberapa materi daring yang dapat diakses dan memiliki ulasan mengenai pemrograman prosedural antara lain:

- ① <https://learnpythonthehardway.org/book/>
- ② https://en.wikibooks.org/wiki/Python_Programming
- ③ [https://www.udacity.com/course/
design-of-computer-programs--cs212](https://www.udacity.com/course/design-of-computer-programs--cs212)

Pendahuluan

Apa itu Python?



Gambar : Guido van Rossum

- Pertama kali diluncurkan tahun 1991 oleh Guido van Rossum
- Merupakan *interpreted language*
- Menekankan pada *readability*
- Multi-paradigma

Example (C)

```
#include <stdio.h>

int main() {
    printf("Hello, World!");
    return 0;
}
```

Example (Python)

```
print 'Hello, World!'
```

Tidak ada kurung kurawal, tidak perlu titik-koma

Python menggunakan prinsip *dynamically typing*, i.e. tidak perlu deklarasi tipe

Example

```
tinggi = 185  
berat = 81.6  
nama = "Barack Obama"  
jabatan = 'presiden'  
menikah = True
```

Example

```
>>> tinggi = 185
>>> tinggi
185
>>> tinggi = 'ya'
>>> tinggi
'ya'
```

Example

```
print "Mari membahas tentang %s" % nama
print "%s memiliki tinggi badan %d cm dan berat %f kg" % (nama, tinggi, berat)
```

Mencetak dengan Formatter

Example

```
print "{} tingginya {} cm dan beratnya {} kg".format(nama, tinggi, berat)
print "Tinggi badan {2} {0} cm, beratnya {1} kg".format(tinggi, berat, nama)
```

Pengisian Variabel - Raw Input

Example (latihan1.py)

```
print "Nama:",  
nama = raw_input()  
  
print "Umur:",  
umur = raw_input()  
  
print "{} berumur {} tahun".format(nama, umur)
```

```
$ python latihan1.py
```

Pengisian Variabel - Argumen

Example (latihan2.py)

```
import sys

nama = sys.argv[1]
umur = sys.argv[2]
print "{} berumur {} tahun".format(nama, umur)
```

```
$ python latihan2.py Obama 55
```

Pengisian Variabel - Argumen

Example (latihan2.py)

```
import sys

nama = sys.argv[1]
umur = sys.argv[2]
print "{} berumur {} tahun".format(nama, umur)
```

```
$ python latihan2.py Obama 55
```

Example (latihan2.py)

```
import sys

_, nama, umur = sys.argv
print "{} berumur {} tahun".format(nama, umur)
```

Manipulasi Variabel

Pengisian variabel dengan dua cara tersebut membuat tipenya menjadi “raw” (%r) secara default. Variabel tersebut kemudian dapat diubah tipenya dengan melakukan *casting*.

Example (segitiga.py)

```
import sys

_, alas, tinggi = sys.argv
print "Segitiga dengan alas {} cm dan tinggi {} cm".format(alas, tinggi)
print "Luas: {} cm2".format(int(alas) * int(tinggi) / 2)
```


Manipulasi Variabel

Pengisian variabel dengan dua cara tersebut membuat tipenya menjadi “raw” (%r) secara default. Variabel tersebut kemudian dapat diubah tipenya dengan melakukan *casting*.

Example (segitiga.py)

```
import sys

_, alas, tinggi = sys.argv
print "Segitiga dengan alas {} cm dan tinggi {} cm".format(alas, tinggi)
print "Luas: {} cm2".format(int(alas) * int(tinggi) / 2)
```

Pembulatan

Di Python, pembagian integer akan dibulatkan ke bawah
→ mungkin perlu mengubah salah satunya menjadi float!

Operator

- Aritmetika: `+`, `-`, `*`, `/`, `%`, `**`, `//`
- Perbandingan: `<`, `>`, `==`, `!=`, `>=`, `<=`
- Logika: `and`, `or`, `not`
- String: `+`
- Bitwise: `|`, `&`, `~`, `^`, `>>`, `<<`
- Spesial: `is`, `in`

Example (Division)

```
>>> 4.0 / 2
2.0
>>> 4.2 / 2
2.1
>>> 4 / 2
2
>>> 3.5 / 2
1.75
>>> 3.5 // 2
1
>>> 7 & 3
3
```

Example (Bitwise)

```
>>> 8 >> 1 # 1000 -> 100
4
>>> 13 & 3 # 1101 & 0011 -> 0011
1
```

Example (in)

```
>>> x = [1, 3, 5, 7, 9]
>>> 3 in x
True
>>> 2 in x
False
```

Control Flow & Looping

- 1 Python tidak menggunakan kurung kurawal “{ }” sebagai pemisah blok kode, tetapi hanya mengandalkan indentasi (spasi atau tab)
- 2 Spasi lebih diutamakan dibandingkan tab
- 3 Standar menurut PEP-8: 4 spasi
- 4 Tab cenderung tidak seragam, kodenya bisa tidak berfungsi!

If - Else if - Else

Example (bmi.py)

```
print "Weight:", # 185
weight = raw_input()
print "Height:", # 81.6
height = raw_input()

bmi = float(weight) / ((float(height)/100) ** 2)
if bmi < 18.5:
    print "underweight"
elif 18.5 <= bmi < 25:
    print "normal"
else:
    print "overweight"
```

While

Example (factorial.py)

```
n = int(raw_input())
factorial = 1

while n > 0:
    factorial *= n
    n -= 1

print factorial
```

For-loop

Di Python, “for” lebih menyerupai “foreach” di PHP

Example (fruits.py)

```
buah = ['apel', 'jeruk', 'pir']  
  
for b in buah:  
    print b
```

Example (fruits.php)

```
$buah = array('apel', 'jeruk', 'pir');  
  
foreach($buah as $b) {  
    echo $b;  
}
```


For

For-loop

Di Python, “for” lebih menyerupai “foreach” di PHP

Example (fruits.py)

```
buah = ['apel', 'jeruk', 'pir']  
  
for b in buah:  
    print b
```

Example (fruits.php)

```
$buah = array('apel', 'jeruk', 'pir');  
  
foreach($buah as $b) {  
    echo $b;  
}
```

Example (it.py)

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

Struktur Data Standar

List

- 1 Struktur data yang dasarnya adalah *array*
- 2 Dapat difungsikan sebagai *stack* (LIFO)
- 3 Dapat difungsikan sebagai *queue* (FIFO)
- 4 String = list of characters

List of Characters

Example (chars.py)

```
print "Masukkan nama Anda:",  
nama = raw_input()  
  
a, e, i = 0, 0, 0  
for c in nama: # memproses karakter dalam string  
    if c == 'a':  
        a += 1  
    elif c == 'e':  
        e += 1  
    elif c == 'i':  
        i += 1  
  
print "Terdapat %d huruf 'a'" % a  
print "Terdapat %d huruf 'e'" % e  
print "Terdapat %d huruf 'i'" % i
```

Mengakses Elemen dalam List

- ➊ Indeks untuk mengakses list dalam Python diawali dari 0
- ➋ Terdapat *negative index* untuk mengakses elemen dari akhir
- ➌ *Slicing* (e.g. untuk mengambil *substring*) dapat dilakukan dengan menggunakan indeks
- ➍ Besaran langkah dapat diatur sebagai elemen terakhir indeks

Mengakses Elemen dalam List

Example

```
>>> buah = ['apel', 'jeruk', 'mangga', 'jambu', 'pir']
>>> buah[0]
'apel'
>>> buah[2]
'mangga'
>>> buah[-1]
'pir'
>>> buah[1:3]
['jeruk', 'mangga']
>>> buah[::2]
['apel', 'mangga', 'pir']
```

Pertanyaan

- 1 Apa yang terjadi jika saya menjalankan: `buah[1:1]`?

Pertanyaan

- 1 Apa yang terjadi jika saya menjalankan: `buah[1:1]`?
- 2 Bagaimana saya dapat membalik (*reverse*) list tersebut?

Pertanyaan

- ❶ Apa yang terjadi jika saya menjalankan: `buah[1:1]`?
- ❷ Bagaimana saya dapat membalik (*reverse*) list tersebut?
- ❸ Apakah tipe data dalam suatu list bisa berbeda?

Mengubah List

Example

```
>>> buah = ['apel', 'jeruk', 'mangga', 'jambu', 'pir']
>>> x = ['nangka', 'durian']
>>> buah + x
['apel', 'jeruk', 'mangga', 'jambu', 'pir', 'nangka', 'durian']
>>> buah
?
?
```

Mengubah List

Example

```
>>> buah = ['apel', 'jeruk', 'mangga', 'jambu', 'pir']
>>> x = ['nangka', 'durian']
>>> buah + x
['apel', 'jeruk', 'mangga', 'jambu', 'pir', 'nangka', 'durian']
>>> buah
?

```

Example

```
>>> x * 2
['nangka', 'durian', 'nangka', 'durian']

```

Break

Fungsi

Mendefinisikan Fungsi

Example

```
def fungsi(argumen): # tidak perlu definisi tipe untuk argumen  
    pass
```

Mendefinisikan Fungsi

Example

```
def fungsi(argumen): # tidak perlu definisi tipe untuk argumen
    pass
```

Example

```
def luasSegitiga(alas, tinggi):
    return alas * tinggi / 2
```

Memanggil Fungsi

Example

```
def luasSegitiga(alas, tinggi):  
    return alas * tinggi / 2  
  
print "Alas:",  
alas = float(raw_input())  
print "Tinggi:",  
tinggi = float(raw_input())  
  
print "Luas segitiga: %.2f" % luasSegitiga(alas, tinggi)
```


Fungsi dengan Jumlah Parameter Tak Tentu

Example

```
def jumlah(*args):  
    total = 0  
    for arg in args:  
        total += arg  
    return total  
  
print jumlah(3, 5, 1, 2, -1, -10, 8, 2, 1, 9)
```

Fungsi dengan Jumlah Parameter Tak Tentu

Example

```
def jumlah(*args):  
    total = 0  
    for arg in args:  
        total += arg  
    return total  
  
print jumlah(3, 5, 1, 2, -1, -10, 8, 2, 1, 9)
```

Argumen Tak Tentu

Ingat kembali tentang `sys.argv`!

Beberapa Fungsi yang Berguna

Menjumlahkan dengan `sum()`

```
>>> sum([1, 2, 3, 4])  
10
```

Menghitung panjang list atau string dengan `len()`

```
>>> len([5, 6, 7, 8])  
4
```

Mengurutkan list dengan `sorted()`

```
>>> sorted([9, 2, 8, 7, 1])  
[1, 2, 7, 8, 9]
```

Beberapa Fungsi yang Berguna

Referensi: <https://docs.python.org/2/library/functions.html>

- Melihat isi modul dengan `dir()`
- Membuka manual dengan `help()`
- Membuka file dengan `open()`

Membaca File

Example

```
f = open('angka.in')
total = 0
for line in f:
    total += int(line)
print total
```

Example

```
f = open('angka.in')
total = 0
for line in f:
    total += int(line)
print total
```

Example

```
f = open('angka.in')
line_no = int(f.readline()) # mengetahui jumlah baris
total = 0
for line in range(line_no):
    total += int(f.readline())
print total
```

Kuis

Buatlah sebuah program untuk menghitung luas n bangun datar yang terdiri dari:

- 1 Persegi
- 2 Persegi panjang
- 3 Segitiga
- 4 Lingkaran (gunakan 3.14 untuk nilai pi)
- 5 Trapesium

Permintaan masukan harus disesuaikan dengan bangun datar yang akan dihitung luasnya, e.g. jika persegi, maka program hanya akan meminta satu masukan, i.e. sisi. Setelah didapatkan n luas bangun datar, urutkan luas bangun datar dari terkecil hingga terbesar dan hitung total luas bangun datar tersebut!

Contoh Masukan

Example (luas.in)

```
4
segitiga
3
4
lingkaran
7
persegi
9
trapesium
8
4
3
```

Example (luas.out)

```
6.0
18.0
81.0
153.86
Total: 258.86
```

Terima kasih