

Latihan Pengantar Python dan NLTK

Disusun Oleh: Ade Romadhony

I. Aturan Dasar Penulisan Sintaks Python

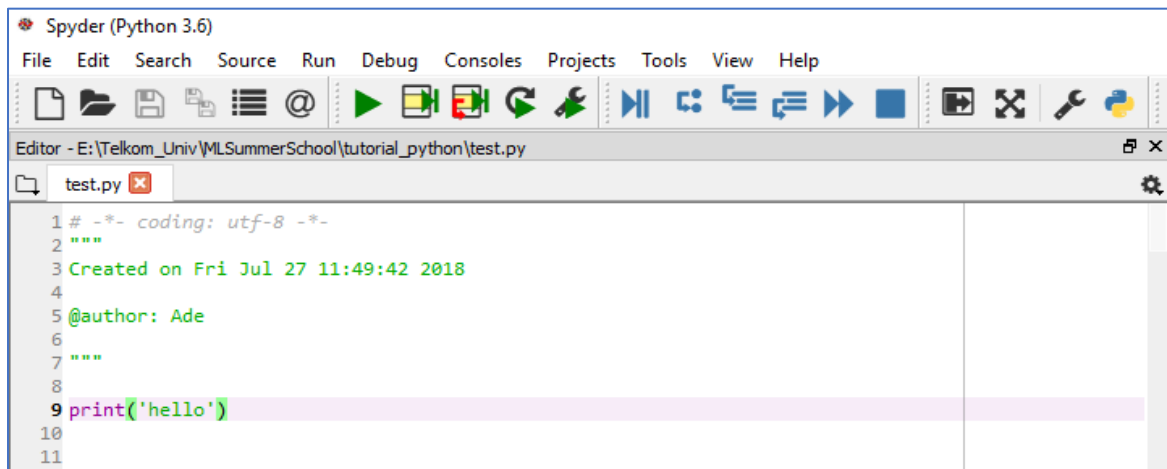
Bagi mahasiswa yang belum familiar dengan sintaks Python, berikut adalah beberapa hal yang perlu diperhatikan:

- Tidak digunakan kurung kurawal untuk mengelompokkan perintah.
- Tidak digunakan titik koma untuk mengakhiri sebuah perintah.
- Indentasi kode program dinyatakan dengan spasi (2, 4, atau 8 spasi)
- Tidak ada pendefinisian tipe

II. Penggunaan Editor Teks atau IDE

Untuk menuliskan kode Python, dapat digunakan teks editor seperti Notepad++, atau IDE seperti JetBrains PyCharm dan Spyder (jika Anda melakukan instalasi menggunakan Anaconda, biasanya Spyder termasuk dalam paket yang di-instal).

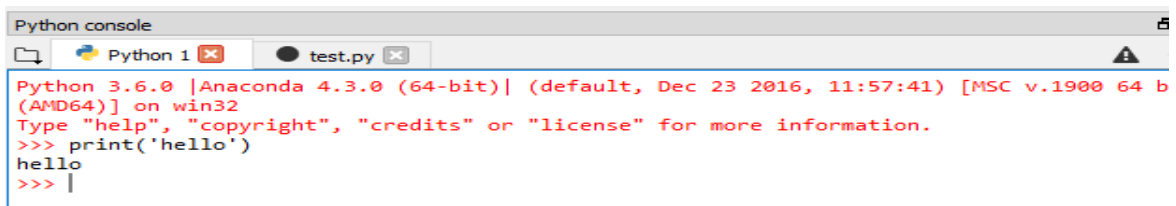
Contoh tampilan antarmuka editor Spyder:



III. Menjalankan (run) Perintah

Untuk me-run perintah, Anda dapat mengakses melalui *command prompt*, atau jika menggunakan IDE seperti Spyder bisa memanfaatkan *console*.

Contoh tampilan antarmuka *console* pada IDE Spyder:



Jika mengakses melalui *command prompt* Anaconda, ketikkan perintah “python”, maka akan tampil *prompt*, contohnya seperti berikut:

```
Anaconda Prompt - python

(D:\anaconda) E:\Telkom_Univ\MLSummerSchool>python
Python 3.6.0 [Anaconda 4.3.0 (64-bit)] (default, Dec 23 2016, 11:57:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

IV. Menjalankan (*run*) File

Untuk dapat menjalankan (*run*) file Python (ekstensi .py) melalui IDE, Anda dapat menggunakan button atau menu **run**, sementara jika menjalankan melalui *command prompt*, dapat dilakukan dengan perintah `python <nama_file.py>`, seperti yang terlihat pada contoh berikut:

```
(D:\anaconda) E:\Telkom_Univ\MLSummerSchool\tutorial_python>python test.py
hello
```

V. Tipe

1. Tipe Dasar

a. Numbers (integers and floating point)

Integer:

```
>>> a = 1
>>> b = 2
>>> c = 3
>>> a + b * c
7
```

Float:

```
>>> a = 1.5
>>> b = 2.7
>>> c = 3.3
>>> a + b * c
10.41
```

Tipe Integer dan Float dapat dioperasikan bersama, dengan hasil akhir adalah Float:

```
>>> d = 5
>>> e = 1.2
>>> d + e
6.2
```

Operasi yang dapat dilakukan pada tipe Integer dan Float adalah:

- Penjumlahan, misal: `a + b`
- Pengurangan, misal: `a - b`
- Perkalian, misal: `a * b`
- Exponen/pangkat, misal: `a ** b`
- Unary minus, misal: `-a`
- Pembagian integer, misal: `a // b`
- Pembagian float, misal: `a / b`

b. Strings

Tipe data string dituliskan di antara tanda kutip tunggal ' atau tanda kutip ganda ". Penulisan dilakukan dalam 1 baris, jika terdiri atas beberapa baris, digunakan karakter \n (*newline*). Operasi penggabungan/konkatenasi string dilakukan dengan operator +.

Contoh:

```
>>> nama_depan = 'adi'
>>> nama_belakang = 'darmawan'
>>> nama_lengkap = nama_depan + ' ' + nama_belakang
>>> nama_lengkap
'adi darmawan'
```

c. List

Python tidak mempunyai tipe data dasar Array, sehingga biasanya digunakan List. Jika ingin menggunakan tipe data Array, dapat memanfaatkan library **Numpy**. Pendefinisian anggota List pada Python dilakukan dengan tanda kurung siku [], dan antar elemen dipisahkan dengan tanda koma , .

```
>>> list_bilangan = [1, 2, 3, 4, 5]
>>> list_bilangan[2]
3
>>> list_bilangan.append(6)
>>> list_bilangan
[1, 2, 3, 4, 5, 6]
```

Pada tipe data *List*, kita dapat melakukan pengurutan (*sort*) dan penambahan (*append*) elemen.

```
>>> list_nama_hari.sort()
>>> list_nama_hari
['jumat', 'kamis', 'minggu', 'rabu', 'sabtu', 'selasa', 'senin']
>>> list_nama_hari[2]
'minggu'
```

```
>>> list_bilangan.append(6)
>>> list_bilangan
[1, 2, 3, 4, 5, 6]
```

d. Dictionary

Tipe data *Dictionary* mempunyai indeks khusus, yang disebut sebagai *key*. Selain Integer, tipe data String juga dapat digunakan sebagai *key*.

```
>>> siswa = {'123001': 'Wati', '123002': 'Budi'}
>>> siswa['123001']
'Wati'
>>> '123002' in siswa
True
```

VI. Dasar komponen program

a. *Control Structure dan Conditions*

Skema umum *control structure* (if....else...) adalah sebagai berikut:

```
if <condition>:
    <statement 1>
    <statement 2>
    <statement 3>
<statement after the if>
```

Conditions berupa tipe boolean, seperti contoh berikut berikut:

- `a == b`
- `a != b`
- `a < b`
- `a > b`
- `a >= b`
- `a in lst`
- `a not in lst`

```
a = 3
if a%2 == 0:
    print('genap')
else:
    print('ganjil')
```

Jika terdapat `if` bersarang (nested if), digunakan kata kunci `elif`.

b. Pengulangan / *Loop*

Pada Python, *loop* dapat dinyatakan dengan kata kunci *for* atau *while*.

Berikut contoh *loop* pada indeks integer dengan kata kunci *for*:

```
for i in range(5):
    print(i)
```

Hasilnya adalah sebagai berikut:

```
0
1
2
3
4
```

Pada perintah *range* juga dapat didefinisikan batas bawah dan atas indeks, seperti contoh berikut:

```
for i in range(2,5):  
    print(i)
```

Dengan hasil sebagai berikut:

```
2  
3  
4
```

Indeks juga dapat berupa elemen sebuah List atau Dictionary, misal:

```
list_nama_hari = ['senin', 'selasa', 'rabu', 'kamis', 'jumat', 'sabtu', 'minggu']  
for nama_hari in list_nama_hari:  
    print(nama_hari)
```

```
siswa = {'123001': 'Wati', '123002': 'Budi'}  
for nis in siswa:  
    print(nis, siswa[nis])
```

Sementara contoh pengulangan dengan *while* adalah sebagai berikut:

```
a = 0 # inisialisasi  
while a<5: # condition  
    print(a+1) # action  
    a += 1 # next element
```

c. Break dan Continue

Kata kunci *break* pada blok pengulangan menyatakan keluar dari blok tersebut, sementara kata kunci *continue* menyatakan lanjut ke iterasi berikutnya.

d. Pendefinisian dan Pemanggilan Fungsi

Fungsi didefinisikan dengan kata kunci *def* <nama_fungsi>:

Berikut contoh pendefinisian dan pemanggilan sebuah fungsi:

```
def pangkat_dua(x):  
    return x**2  
  
print(pangkat_dua(5))
```

VII. Numpy

Numpy adalah sebuah library untuk operasi numerik. Dengan Numpy, kita dapat mendefinisikan array multidimensional, dan memanfaatkan fungsi-fungsi yang tersedia untuk operasi numerik. Numpy banyak digunakan jika kita membutuhkan penggunaan struktur data matriks atau vektor. Untuk dapat menggunakan sebuah library, kita gunakan kata kunci *import*. Berikut adalah contoh pendefinisian array dengan menggunakan library Numpy:

```
import numpy as np # <- import library

A = np.array([
    [0,1,2],
    [3,4,5],
    [5,7,8]])
print(A[0,0])
print(A[0,1])
print(A[1])
```

Daftar fungsi-fungsi untuk operasi pada array Numpy dapat dilihat di: <https://docs.scipy.org/doc/numpy-1.14.0/genindex.html>

VIII. Pandas dan Matplotlib

Pandas merupakan salah satu library untuk analisis data. Dalam penggunaannya untuk Pembelajaran Mesin, Pandas banyak dimanfaatkan untuk pembacaan data dari file spreadsheet (misal: xls, csv). Sementara Matplotlib adalah library untuk visualisasi data, yang juga banyak dimanfaatkan dalam bidang Pembelajaran Mesin. Dengan visualisasi, kita dapat lebih mudah menangkap karakteristik data.

Berikut adalah contoh pemanfaatan pandas untuk pembacaan dataset Iris, dan kemudian *plotting* (scatterplot) ke bidang 2 dimensi dengan Matplotlib.

Import library yang diperlukan:

```
# Import Library
import pandas as pd
import matplotlib.pyplot as plt
```

Pembacaan data dari file csv:

```
#Membaca dataset IRIS dengan Pandas Dataframe
dataset = pd.read_csv('iris.csv')

#Print 10 baris teratas dari data
print(dataset.head(10))
```

Plot data ke bidang 2 dimensi:

```
# Plotting akan dilakukan terhadap ratio sepal = sepal length / sepal width
ratio = dataset["sepal.length"]/dataset["sepal.width"]
# Scatter plot, kelompokkan berdasar jenis / variety Iris
for name, group in dataset.groupby("variety"):
    plt.scatter(group.index, ratio[group.index], label=name)

# Keterangan sumbu x dan sumbu y
plt.xlabel("ID Iris")
plt.ylabel("Rasio Sepal")

# Legend
plt.legend()

# Show
plt.show()
```

Sumber: <https://stackoverflow.com/questions/45862223/use-different-colors-in-scatterplot-for-iris-dataset> , <http://www.pybloggers.com/2015/09/my-first-time-using-matplotlib/>

IX. Jupyter/IPython Notebook

Jupyter/IPython Notebook adalah aplikasi berbasis web (dijalankan menggunakan browser), dengan konsep live-coding, dan dengan menggunakannya kita bisa melakukan kombinasi beberapa hal: menuliskan dan eksekusi kode serta menuliskan keterangan yang juga dapat dilakukan dalam bentuk teks, formula matematika, grafik, dan *rich media* lainnya.

Untuk instalasi, silahkan mengunjungi link berikut: <https://jupyter.org/install> , kemudian petunjuk tutorial singkat dapat diikuti dari sini: <https://hub.mybinder.org/user/ipython-ipython-in-depth-cvz1k3w/notebooks/binder/Index.ipynb> . Beberapa bagian kode singkat yang tercantum dalam tutorial ini juga dituliskan dalam bentuk Notebook (tutorial1.ipynb dan tutorial2.ipynb).

X. NLTK (Natural Language Toolkit)

NLTK merupakan sebuah library Python yang cukup populer digunakan di bidang pemrosesan bahasa alami. NLTK mempunyai banyak fungsi yang terkait dengan pemrosesan teks dan bahasa, mulai dari tokenisasi, pemberian label POSTag, *parsing* sebuah kalimat, dll. NLTK juga menyediakan koleksi dokumen/corpora, termasuk dataset Bahasa Indonesia. Untuk dapat menggunakan NLTK, silakan melakukan instalasi dengan mengikuti petunjuk di <http://www.nltk.org/install.html> , dan untuk menambahkan data/corpus, silakan kunjungi <http://www.nltk.org/data.html> .

Berikut contoh sederhana pemanfaatan fungsi pemrosesan teks dengan NLTK:

Tokenisasi kalimat

Sebelumnya, jangan lupa melakukan import *library*.

```
import nltk
```

#Definisikan kalimat masukan

```
Sentence = 'The operation began this morning and all the crews stay in the location'
```

#Panggil fungsi tokenisasi, tampung hasilnya dalam sebuah variabel

```
tokens = nltk.word_tokenize(sentence)
```

#Latihan 1: print token hasil tokenisasi, sertakan informasi panjang token tersebut

#Latihan 2: ubah karakter pada kalimat masukan menjadi *lowercase* semua

#Latihan 3: lakukan tokenisasi pada kalimat yang sudah diubah penulisannya menjadi *lowercase* semua dan print hasilnya

#Latihan 4: lakukan stemming pada kalimat masukan

#Latihan 5: lakukan lemmatization pada kalimat masukan

Bandingkan hasil yang diperoleh pada latihan 4 dan 5

#Latihan 6: diberikan teks yang lebih panjang (paragraf), lakukan penghitungan frekuensi kemunculan tiap kata yang ada pada teks tersebut.

par = 'Pesilat Indonesia kembali meraih medali emas ke-12, setelah Pipiet Kamelia berhasil menumbangkan pesilat Vietnam Thi Cam Nhi Nguyen di babak final pertandingan cabang olahraga Pencak Silat di ajang Asian Games 2018. Pipiet menang telak 5-0 atas Thi Cam di kelas D putri 60kg-65 kg, yang berlangsung di Padepokan Pencak Silat Taman Mini Indonesia Indah (TMII), Jakarta Timur, Rabu petang. Dengan kemenangan ini, Pipiet berhasil membawa medali emas, sementara Thi Cam harus puas juara kedua dengan medali perak.'

Tutorial ini disusun dari berbagai sumber, antara lain:

1. <https://mybinder.org/v2/gh/luispedro/talk-python-intro/master>
2. <https://docs.python.org/3/tutorial/>