

# Praktikum 1

## Tutorial Python untuk Pengantar Pembelajaran Mesin

### Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Setelah mengikuti modul bagian ini diharapkan mahasiswa dapat mengetahui secara sekilas bagaimana menggunakan Python untuk membangun suatu proyek pembelajaran mesin.
2. Detil masing-masing bagian akan disampaikan pada sesi praktikum berikutnya.

### Dasar Teori

Pada bagian ini, kita akan mempelajari tahapan pengerjaan proyek pembelajaran mesin menggunakan python. Secara umum tahapan pembuatan proyek pembelajaran mesin dengan python terdiri dari tahapan berikut :

1. Menginstal platform Python dan SciPy.
2. Memuat dataset.
3. Meringkas kumpulan data.
4. Memvisualisasikan kumpulan data.

### Praktikum

#### 1. Instalasi Library

Terdapat lima pustaka utama yang sering digunakan dalam pembelajaran mesin yang perlu diinstall terlebih dahulu sebelum Anda melakukan praktikum. Library tersebut adalah:

1. Numpy
2. Pandas
3. Matplotlib
4. Scipy
5. Scikit-learn

Namun demikian hal ini bisa dilewati jika anda tidak perlu menginstal di komputer anda, karena hal ini bisa dilakukan secara online menggunakan Google Colab. Colab singkatan dari *Colaboratory* yang memungkinkan Anda untuk menulis dan mengeksekusi Python melalui

browser secara remote. Dengan demikian Anda tidak perlu menginstall secara local dan mengkonfigurasi tools tersebut di atas.

Jalankan kode berikut pada cell di Google Colab untuk mengecek apakah library tersebut telah terinstall

```
import numpy
import pandas
import matplotlib.pyplot
import scipy
import sklearn
Import csv

print('numpy: {}'.format(numpy.__version__)) # numpy
print('pandas: {}'.format(pandas.__version__)) # pandas
print('matplotlib: {}'.format(matplotlib.__version__)) # matplotlib
print('scipy: {}'.format(scipy.__version__)) # scipy
print('sklearn: {}'.format(sklearn.__version__)) # scikit-learn
```

Jika Anda melakukan dengan benar, maka contoh output yang dihasilkan adalah sebagai berikut:

```
numpy: 1.18.5
pandas: 1.1.2
matplotlib: 3.2.2
scipy: 1.4.1
sklearn: 0.22.2.post1
```

## 2. Memuat Dataset

Setelah tools siap dipakai, langkah selanjutnya memanggil dataset yang akan diolah. Terdapat beberapa cara untuk memuat dataset: memuat dari file lokal atau mendownload dari Internet.

### a. Memuat Dataset Lokal

Buatlah sebuah file CSV dengan nama **kontak.csv**. Anda dapat menggunakan aplikasi Notepad atau teks editor lainnya, dan simpan berkas dengan nama **kontak.csv**. Isi berkas **kontak.csv** adalah:


NO,NAMA,TELEPON  
1,Achmad Ali,081234  
2,Budi Utomo,08712333  
3,Toni Saja,08733311  
4,Dewi Utami,0851231

Anda dapat mengunggah (upload) berkas **kontak.csv** ke Google Colab dengan menggunakan kode berikut :

```
from google.colab import files
kontak = files.upload()
for fn in kontak.keys():
    print('Nama file "{name}" dengan panjang {length} bytes'.format(
        name=fn, length=len(kontak[fn])))
```

Contoh keluaran dari proses upload file adalah sebagai berikut:

---

  kontak.csv  
**kontak.csv**(application/vnd.ms-excel) - 103 bytes, last modified: n/a - 100% done  
Saving kontak.csv to kontak.csv  
Nama file "kontak.csv" dengan panjang 103 bytes

---

Berkas **kontak.csv** telah sukses terupload.

#### b. Memuat Dataset dari Internet

Apabila dataset telah tersedia di Internet, maka Anda dapat langsung mengunduh (download) langsung ke Google Colab. Anda dapat memanfaatkan aplikasi **wget** yang tersedia di Google Colab untuk mendownload data. Pada praktikum ini, Anda akan mendownload dataset **iris** yang tersedia di <https://dataset-ppm.s3.amazonaws.com/iris.csv> . Jalankan code di bawah ini untuk mendownload berkas **iris.csv**. Jangan lupa menuliskan tanda seru (!) di depan perintah wget

```
! wget https://dataset-ppm.s3.amazonaws.com/iris.csv
```

Cek menggunakan perintah **ls** untuk mendapatkan file yang tersedia di Google Colab

```
! ls
```

Jika Anda melakukan perintah yang ada dengan urutan yang benar, maka hasilnya adalah sebagai berikut:

```
↳ iris.csv kontak.csv sample_data
```

Artinya, di Google Colab sudah terdapat berkas **iris.csv** dan **kontak.csv**.

### c. Membaca format CSV

File CSV dapat diuraikan (parsing) ke dalam bentuk list atau dictionary. Fungsi di bawah ini digunakan untuk mengubah CSV menjadi list.

```
def csv_list(filename):  
    data = []  
    with open('kontak.csv') as csv_file:  
        csv_reader = csv.reader(csv_file, delimiter=",")  
        for row in csv_reader:  
            data.append(row)  
    data.pop(0)  
    return data
```

Cek apakah fungsi csv\_list dapat berjalan dengan baik

```
list_kontak = csv_list('kontak.csv')  
print(list_kontak)
```

Dictionary adalah struktur data yang tersusun dari *key* dan *value*. Untuk memarsing CSV menjadi dictionary, gunakan fungsi DictReader()

```
def csv_dict(filename):  
    data = []  
    with open(filename) as csv_file:  
        csv_reader = csv.DictReader(csv_file)  
        for row in csv_reader:  
            data.append(row)  
    return data
```

Cek apakah fungsi `csv_dict` dapat berjalan dengan baik

#### d. Membaca CSV menggunakan Pandas

Pandas merupakan library Python yang digunakan untuk membuat dan mengolah *dataframe*. Dataframe merupakan struktur data dua dimensi yang menyerupai tabel pada Microsoft Excel. Dataframe memiliki fitur yang memudahkan pengaksesan dan pemrosesan data.

Pandas memiliki fitur untuk mengambil data dari CSV dengan mudah. Jalankan code di bawah ini untuk membentuk dataframe dari berkas **iris.csv**.

```
iris_df = pandas.read_csv('iris.csv')
```

Anda dapat melihat sebagian isi Dataframe menggunakan property **head** dari dataframe yang bersangkutan

```
iris_df.head()
```

Properti **head** akan menampilkan 5 data teratas dari dataframe

| ↗ | sepal_length | sepal_width | petal_length | petal_width | species     |
|---|--------------|-------------|--------------|-------------|-------------|
| 0 | 5.1          | 3.5         | 1.4          | 0.2         | Iris-setosa |
| 1 | 4.9          | 3.0         | 1.4          | 0.2         | Iris-setosa |
| 2 | 4.7          | 3.2         | 1.3          | 0.2         | Iris-setosa |
| 3 | 4.6          | 3.1         | 1.5          | 0.2         | Iris-setosa |
| 4 | 5.0          | 3.6         | 1.4          | 0.2         | Iris-setosa |

### 3. Meringkas Kumpulan Data

Terdapat beberapa informasi yang dapat diketahui dengan mudah dari struktur dataframe pada Pandas, yaitu:

1. Dimensi dataset
2. Melihat isi dataset
3. Ringkasan statistik dari dataset
4. Rincian data dengan variabel kelas

Tabel berikut berisi fungsi/properti beserta kegunaannya

| Fungsi/properti      | Deskripsi                                              |
|----------------------|--------------------------------------------------------|
| shape                | Menampilkan dimensi dataframe (banyak baris dan kolom) |
| head(n)              | Menampilkan n baris pertama dari data                  |
| describe()           | Menampilkan distribusi data                            |
| groupby(column_name) | Melakukan grouping berdasarkan kolom tertentu          |

Kode berikut menampilkan ringkasan dari dataframe **iris\_df**

```
print(iris_df.head(6)) # menampilkan 6 data pertama
print(iris_df.shape) #menampilkan dimensi dataframe
print(iris_df.describe()) #menampilkan distribusi data
print(iris_df.groupby('species').size()) #menampilkan banyak data per kelas
```

#### 4. Visualisasi Kumpulan Data

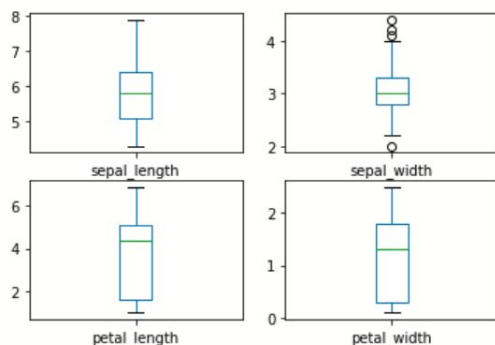
Data dapat divisualiasi menggunakan grafik dengan cara di-ploting. Ada dua cara plotting :

1. Plot univariat untuk lebih memahami setiap atribut
2. Plot multivariasi untuk lebih memahami hubungan antar atribut

Plot univariat merupakan plot dari masing-masing variabel. Mengingat variabel input berupa numerik, kita dapat membuat plot box dan whisker. Kode berikut menampilkan box plot

```
from matplotlib import pyplot
iris_df.plot(kind='box',subplots=True, layout=(2,2), sharex=False,
sharey=False)
pyplot.show()
```

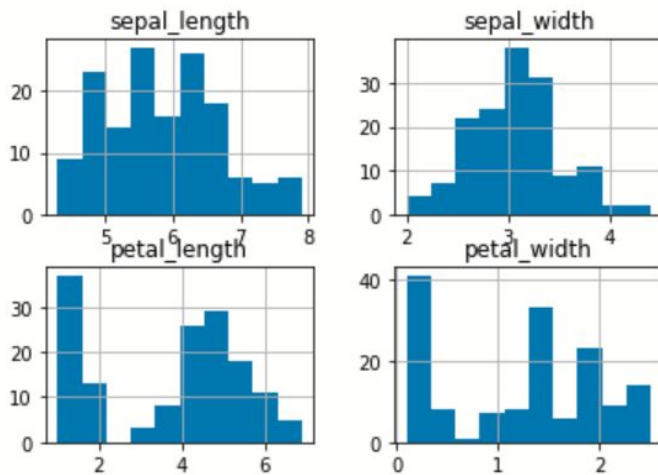
Hasilnya berupa plot box



Kita juga dapat membuat histogram dari setiap variabel input untuk mendapatkan gambaran tentang distribusinya.

```
iris_df.hist()  
pyplot.show()
```

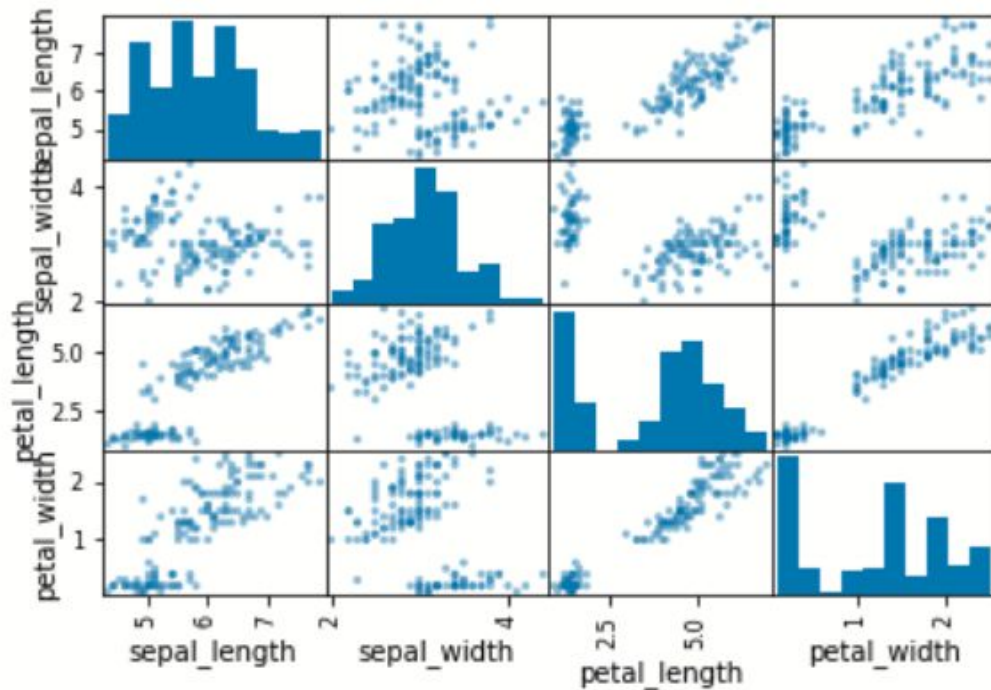
Hasilnya berupa plot histogram



Plot multivariasi dapat memanfaatkan scatter plot untuk melihat interaksi antar variabel.

```
from pandas.plotting import scatter_matrix  
scatter_matrix(iris_df)  
pyplot.show()
```

Hasilnya merupakan scatter plot yang menggambarkan sebaran data antar dua variabel

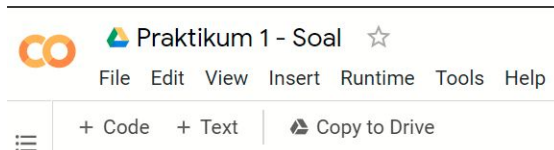


## Tugas

1. Download Glass dataset dari <https://dataset-ppm.s3.amazonaws.com/glass.csv>
2. Masukkan data Glass ke dataframe menggunakan Pandas.
3. Tampilkan 10 data pertama
4. Hitung rata-rata nilai per variabel untuk setiap data
5. Hitung rata-rata nilai per variabel untuk setiap data dikelompokkan berdasarkan Type
6. Buatlah plot bertipe 'line' untuk masing-masing variabelnya. Referensi : [Pandas Plot](#)

Petunjuk pengerjaan soal:

1. Klik link  
[https://colab.research.google.com/drive/1M30zLfbOo5\\_z8Lv8tz1ttACjaT8j32qw?usp=sharing](https://colab.research.google.com/drive/1M30zLfbOo5_z8Lv8tz1ttACjaT8j32qw?usp=sharing)
2. Klik tombol Copy to Drive



3. Beri nama file Praktikum 1 - Nama - NIM
4. Isilah cell yang kosong
5. Download file \*.ipynb dengan cara klik **File -> Download .ipynb**
6. Kumpulkan file \*.ipynb ke asisten