# Modul 1 Pengenalan *Tools* Pembelajaran Mesin

## 1.1 Tujuan

Mahasiswa dapat menggunakan *tools* pemrograman untuk memprogram pembelajaran mesin dengan benar.

## 1.2 Dasar Teori

Python merupakan sebuah bahasa pemrograman yang cukup populer yang dirilis sejak tahun 1991. Bahasa pemrograman ini dapat digunakan untuk pengembangan web, *software*, matematika, serta *scripting* sistem. Python dapat bekerja untuk platform yang berbeda seperti Windows, Mac, Linux, Raspberry Pi, dan lain sebagainya. Selain itu bahasa pemrograman ini memiliki *syntax* yang sederhana sehingga memudahkan *programmer* untuk menuliskan program. Versi terbaru dari Python adalah Python 3, yang telah diupdate dari sisi keamanannya dibandingkan versi sebelumnya.

Python didesain untuk dapat mudah dibaca dan memiliki kemiripan dengan bahasa Inggris. Berbeda dengan *syntax* untuk bahasa pemrograman lain yang menggunakan titik koma (;) atau tanda kurung ([]) untuk mengakhiri program, Python menggunakan baris baru untuk mengakhiri program. Pada Python, lingkup program menggunakan indentasi dan *whitespace*.

```
print("Hello, World!")
```

Gambar 1.1 Contoh program “Hello, World!” pada Python

### A. Instalasi Python

Pada beberapa PC atau Mac terbaru Python telah terinstal langsung. Untuk itu ada baiknya sebelum melakukan instalasi cek dulu apakah Python sudah terinstal. Berikut langkah yang dapat dilakukan untuk memastikan apakah Python sudah terinstal untuk platform Windows.

1. Buka *Command Line* (cmd.exe) pada *start bar*.
2. Ketik *command* sebagai berikut.

```
C:\Users\Your Name>python --version
```

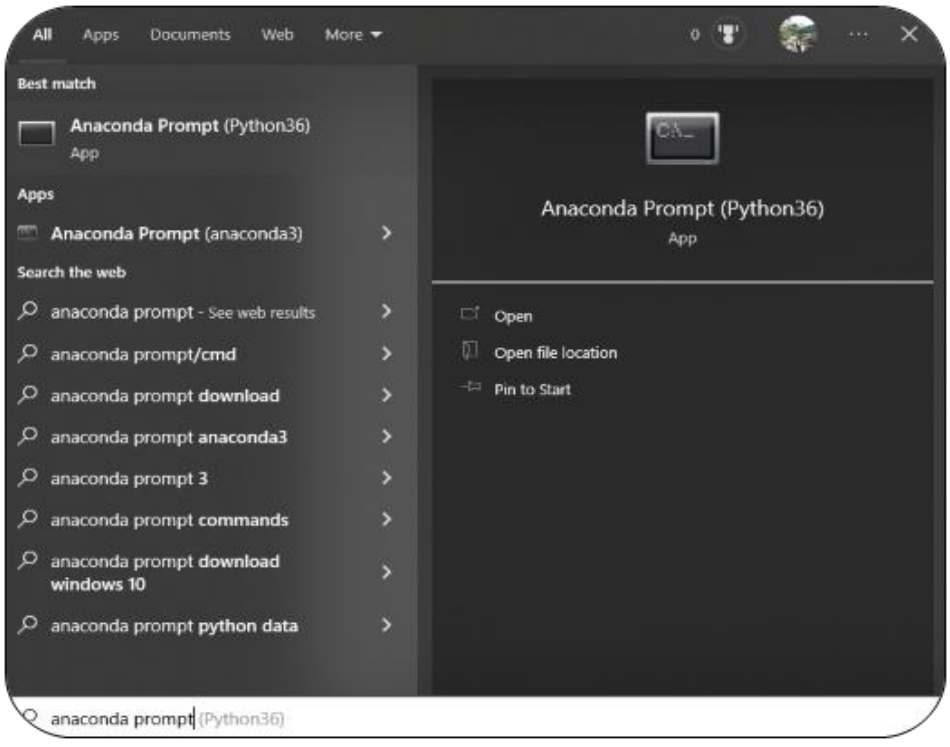
3. Klik ‘enter’.
4. Apabila muncul Python sekaligus versinya, maka dapat dipastikan bahwa Python telah terinstal pada platform yang Anda gunakan.

Apabila tidak muncul versi Python yang digunakan, maka Python dapat diunduh secara gratis pada website www.python.org.

Cara lain yang dapat dilakukan untuk membangun ekosistem pemrograman Python adalah dengan melakukan instalasi Python melalui Anaconda. Metode ini lebih mudah dari pada instalasi Python melalui website resmi Python. Cara instalasinya dapat mengikuti langkah berikut.

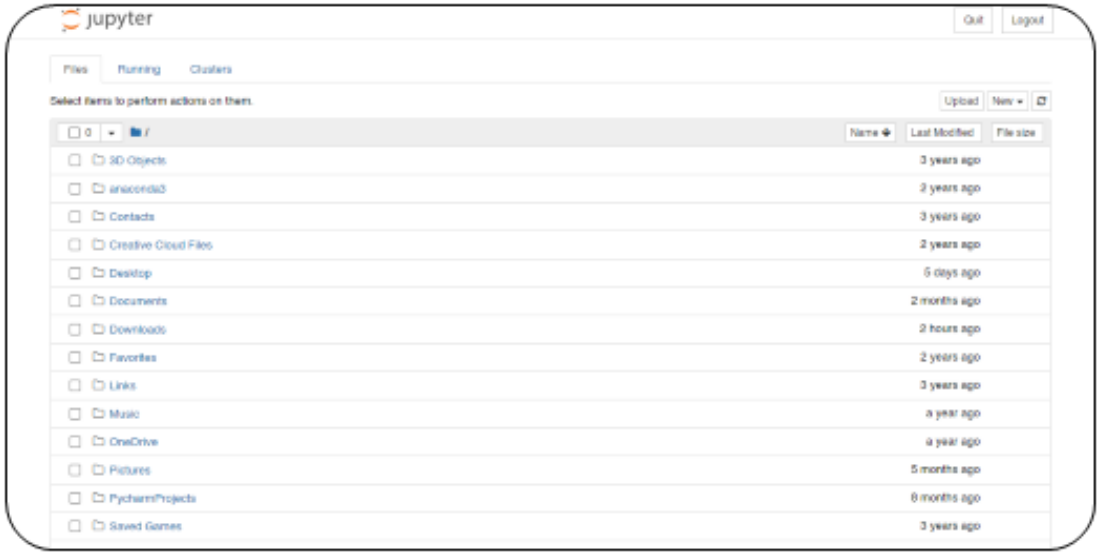
1. Buka www.anaconda.com/products/individual untuk mengunduh versi terbaru Anaconda sesuai dengan platform yang Anda gunakan.


2. Unduh dan jalankan *installer*. Setelah selesai, cek dengan cara mengetikkan kata kunci "Anaconda Prompt" pada *search bar*.



3. Buka Anaconda Prompt, kemudian ketikkan "`jupyter notebook`" pada Anaconda command prompt untuk membuka aplikasi Jupyter Notebook.

Berikut tampilan home untuk Jupyter Notebook. Pada Jupyter Notebook, modul, package, atau library eksternal lain yang dibutuhkan tidak perlu diinstal satu-persatu karena seluruhnya telah tersedia. Selain itu, kita dapat melakukan *coding* secara *online* tanpa harus menginstal IDE atau Python *Interpreter*.




Hanya saja jika dapat memastikan bahwa koneksi cukup cepat dan ingin bisa melakukan pemrograman kolaboratif maka dapat memanfaatkan tools *online* (berbasis *browser*) tanpa harus melakukan instalasi pada PC dapat juga memanfaatkan *tools* Google Collab. Google Collab dapat diakses melalui www.colab.research.google.com.

## B. *Library* dan *Package* Python untuk Pembelajaran Mesin

Terdapat beberapa *Library* dan *Package* pada Python yang dibutuhkan untuk melakukan pemrograman Pembelajaran Mesin, antara lain Pandas, NumPy, SciPy, Matplotlib, dan Scikit-Learn (Gambar 1.2).

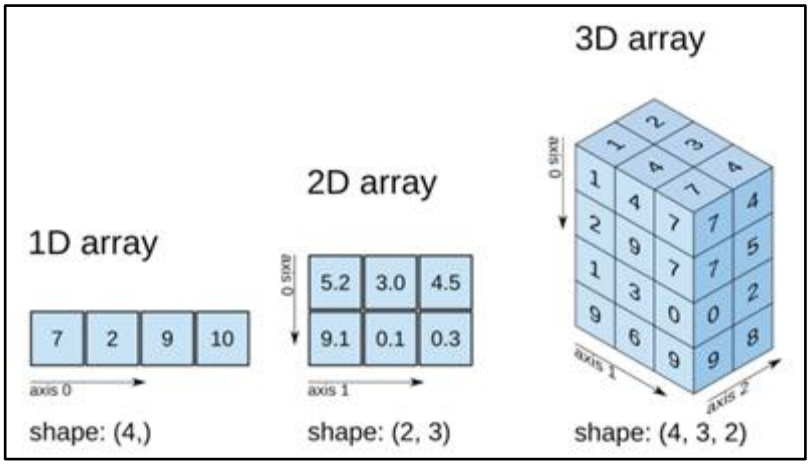

Gambar 1.2 *Library* dan *Package* pada Python

**1. Pandas**

Pandas adalah *library* yang digunakan untuk memanipulasi dan menganalisis data berkinerja tinggi. Pada Pandas, struktur data yang bekerja termasuk data seri dan *frame*. Data serial adalah sebuah *array* 1 dimensi yang terdiri atas data homogen, dimana ukurannya tetap (tidak dapat diubah) tetapi nilainya dapat diubah. Data *frame* adalah data *array* 2 dimensi yang heterogen dan terdiri atas 2 atau lebih seri, yang ukuran dan nilainya dapat diubah.

**2. NumPy**

NumPy atau "Numerical Python" adalah *package* Python yang terdiri atas objek *array* multidimensi dan sekumpulan fungsi untuk memproses *array*. *Library* ini juga berfungsi untuk memproses data pada domain aljabar linear, transformasi fourier, dan matriks. Objek *array* pada NumPy disebut sebagai "ndarray" atau n-dimensional array. Ndarray adalah array homogen yang dioptimalkan untuk pemrosesan data secara cepat.



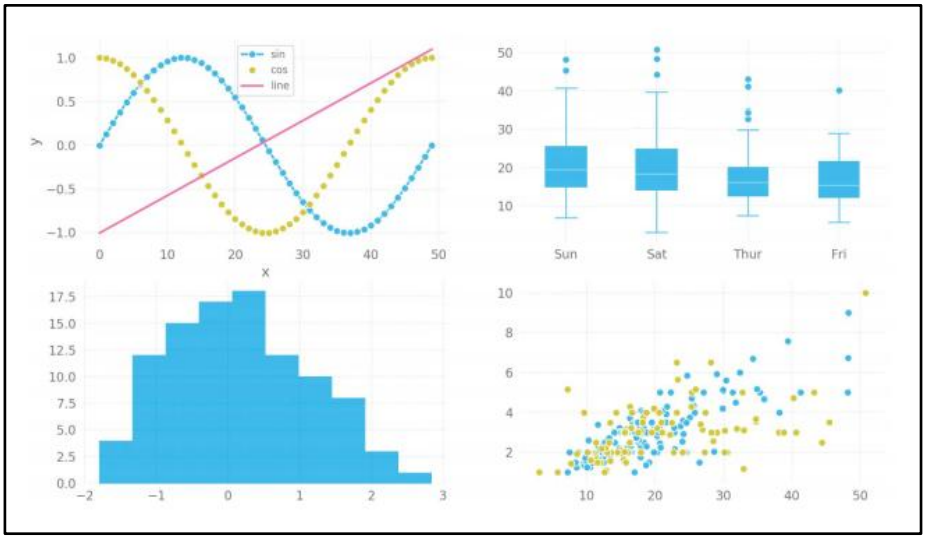

Gambar 1.3 *Array* Multidimensi (n-*Dimensional Array*)

**3. SciPy**

SciPy adalah *library* Python yang dibangun untuk bekerja dengan *array* NumPy guna mengoptimalkan dan meningkatkan efisiensi operasi numerik. Bersama-sama dengan NumPy, SciPy dapat bekerja pada seluruh sistem operasi popular seperti Windows, Mac, dan Linux.

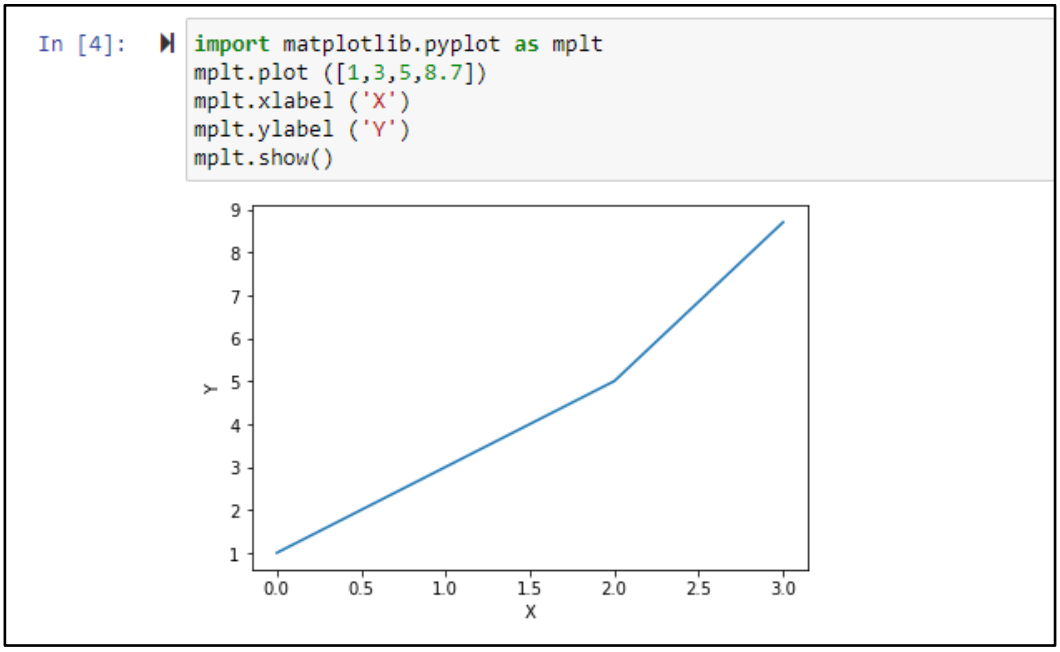
**4. Matplotlib**

Matplotlib adalah *library* Python yang digunakan untuk membuat grafik dan *plot* 2D dengan menggunakan skrip Python. *Library* ini memiliki modul bernama "pyplot" yang memudahkan *plot* dengan menyediakan fitur untuk mengontrol *style* garis, properti huruf, sumbu pemformatan, dan lain sebagainya. Fitur ini sangat bermanfaat untuk visualisasi data. Visualisasi data adalah representasi grafis dari informasi dan data. Dengan menggunakan elemen visual seperti bagan, grafik, dan peta, *tools* visualisasi data menyediakan cara untuk melihat dan memahami tren, *outlier*, dan pola dalam data.




Gambar 1.4 *Plot* data menggunakan Matplotlib

Fungsi "`matplotlib.pyplot`" adalah kumpulan fungsi yang membuat matplotlib berfungsi seperti MATLAB. Setiap fungsi pyplot membuat beberapa perubahan pada gambar: misalnya, membuat gambar, membuat area *plot* dalam gambar, memplot beberapa baris di area *plot*, menghias *plot* dengan label, dan sebagainya. Untuk mengujinya dapat dilakukan dengan masuk ke Jupyter Notebook dan mulai dengan mengimpor modul matplotlib.pyplot (Gambar 1.5).




Gambar 1.5 Import Matplotlib dan plot data (*default*)

## 5. Scikit-Learn

Scikit-learn atau Sklearn adalah *library* Python yang berguna dan *powerful* untuk pembelajaran mesin. *Package* ini menyediakan pilihan fungsi yang efisien untuk pembelajaran mesin dan pemodelan statistik termasuk klasifikasi, regresi, pengelompokan

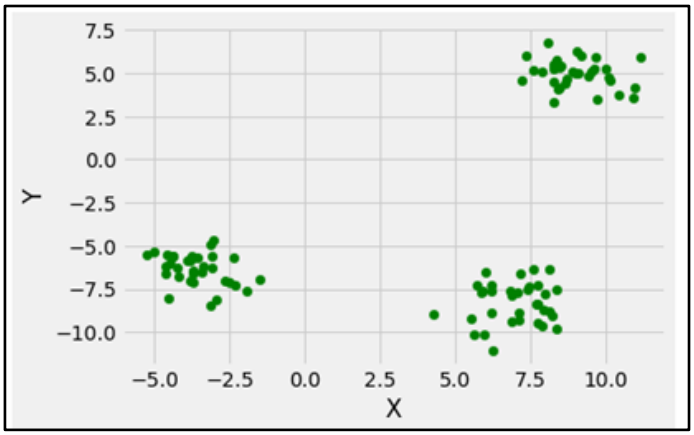
(*clustering*) dan reduksi dimensi melalui antarmuka konsisten dengan Python. *Library* ini, yang sebagian besar ditulis dengan Python, dibangun di atas NumPy, SciPy, dan Matplotlib.

## C. Membuat Dataset Menggunakan Sklearn

Library Python Sklearn menyediakan dataset sampel yang akan membantu dalam pembuatan dataset. Fitur ini cepat dan sangat mudah digunakan. Contoh jenis sampel yang disediakan dapat diakses pada laman https://scikit-learn.org/stable/modules/classes.html?highlight=sklearn+dataset#module-sklearn.datasets. Untuk semua metode di atas, praktikan perlu mengimpor sklearn.datasets.samples_generator.

```
# Creating Test DataSets using sklearn.datasets.make_blobs
from sklearn.datasets import make_blobs
from matplotlib import pyplot as plt
from matplotlib import style

style.use("fivethirtyeight")

X, y = make_blobs(n_samples = 100, centers = 3,
                  cluster_std = 1, n_features = 2)

plt.scatter(X[:, 0], X[:, 1], s = 40, color = 'g')
plt.xlabel("X")
plt.ylabel("Y")

plt.show()
plt.clf()
```

Output dataset disajikan pada Gambar 1.6.




Gambar 1.6 Hasil output dataset

## D. Memulai Dataset untuk Pengujian Pembelajaran Mesin

Ketika berbicara mengenai Pembelajaran Mesin, maka aspek utama yang dibutuhkan adalah sekumpulan data (dataset). Meskipun ada banyak dataset yang bisa ditemukan di berbagai repository data seperti Kaggle, terkadang kita akan punya dataset sendiri yang mungkin diperoleh dari sensor yang kita kembangkan sendiri. Untuk itu kita perlu tahu bagaimana cara membuat dataset yang memungkinkan kita melatih model pembelajaran mesin. Library yang digunakan adalah NumPy, Pandas, dan Matplotlib.

Menurut teori probabilitas, distribusi normal atau Gaussian adalah distribusi probabilitas kontinu yang memiliki *mean* simetris. Hal ini menunjukkan bahwa data yang dekat dengan *mean* lebih sering terjadi daripada data yang jauh dari *mean*. Distribusi normal digunakan dalam statistika data dan sering digunakan untuk mewakili variabel acak bernilai riil.

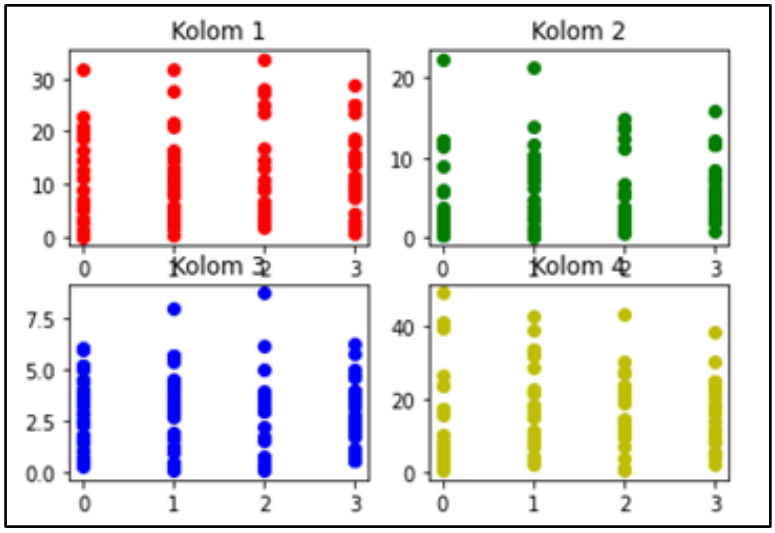
Distribusi normal adalah jenis distribusi yang paling sering dalam analisis statistika. Distribusi normal standar memiliki dua parameter: *mean* dan standar deviasi. *Mean* adalah tendensi sentral dari distribusi. Standar deviasi adalah ukuran variabilitas yang didefinisikan oleh lebar distribusi normal. Standar deviasi menunjukkan seberapa jauh sebuah nilai dari *mean*.

Apabila kita ingin menghasilkan dataset dengan 4 kolom, dimana setiap kolom dalam dataset mewakili sebuah fitur. Kolom ke-5 dari dataset adalah label keluaran (output) yang bervariasi antara 0-3. Maka skrip yang dapat digunakan adalah sebagai berikut.

```
# importing libraries
import numpy as np
import pandas as pd
import math
import random
import matplotlib.pyplot as plt

# defining the columns using normal distribution

# column 1
point1 = abs(np.random.normal(1, 12, 100))
# column 2
point2 = abs(np.random.normal(2, 8, 100))
# column 3
point3 = abs(np.random.normal(3, 2, 100))
# column 4
point4 = abs(np.random.normal(10, 15, 100))

# x contains the features of our dataset
# the points are concatenated horizontally
```

```
# using numpy to form a feature vector.
x = np.c_[point1, point2, point3, point4]

# the output labels vary from 0-3
y = [int(np.random.randint(0, 4)) for i in range(100)]

# defining a pandas data frame to save
# the data for later use
data = pd.DataFrame()

# defining the columns of the dataset
data['col1'] = point1
data['col2'] = point2
data['col3'] = point3
data['col4'] = point4

# plotting the various features (x)
# against the labels (y).
plt.subplot(2, 2, 1)
plt.title('Kolom 1')
plt.scatter(y, point1, color ='r', label ='kolom1')

plt.subplot(2, 2, 2)
plt.title('Kolom 2')
plt.scatter(y, point2, color = 'g', label ='kolom2')

plt.subplot(2, 2, 3)
plt.title('Kolom 3')
plt.scatter(y, point3, color ='b', label ='kolom3')

plt.subplot(2, 2, 4)
plt.title('Kolom 4')
plt.scatter(y, point4, color ='y', label ='kolom4')

# saving the graph
plt.savefig('data_visualization.jpg')

# displaying the graph
plt.show()
```

Output dataset disajikan pada Gambar 1.7.




Gambar 1.7 Hasil output visualisasi dataset

## E. Introduksi: Kaggle

Kaggle adalah sebuah platform global yang dirancang untuk memfasilitasi kolaborasi, pembelajaran, dan eksplorasi di bidang data sains dan pembelajaran mesin. Didirikan pada tahun 2010 dan kini dimiliki oleh Google, Kaggle telah menjadi tempat berkumpul bagi data saintis, analis, dan pengembang dari seluruh dunia untuk berbagi pengetahuan, memecahkan masalah nyata, dan mengembangkan keterampilan mereka.

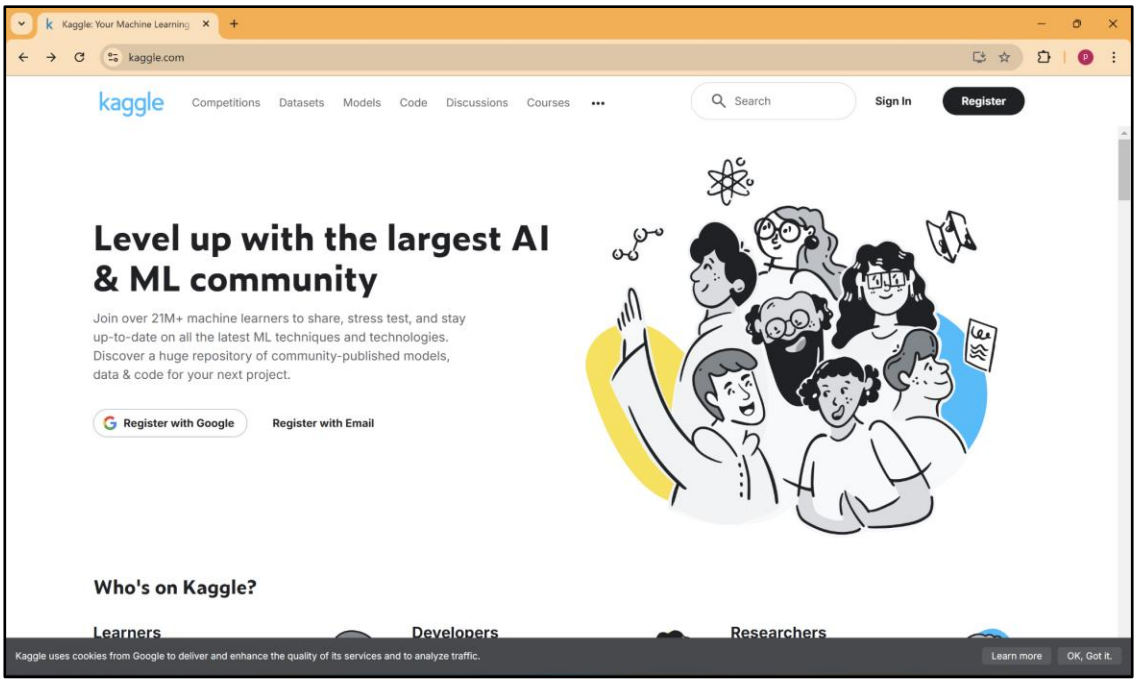
Salah satu fitur unggulan Kaggle adalah kompetisi sains data yang menawarkan tantangan nyata dari berbagai perusahaan dan organisasi. Peserta kompetisi diberikan dataset, deskripsi masalah, dan target yang harus dicapai menggunakan pendekatan data sains atau pembelajaran mesin. Selain sebagai sarana pembelajaran, kompetisi ini sering kali memberikan hadiah menarik, baik berupa uang tunai maupun pengakuan profesional.

Selain kompetisi, Kaggle juga menyediakan *repository* dataset publik yang beragam. Dataset ini mencakup topik-topik seperti kesehatan, keuangan, olahraga, dan media sosial, yang dapat digunakan untuk eksplorasi data, analisis, maupun pelatihan model pembelajaran mesin. Dengan begitu, pengguna dapat mengakses dataset berkualitas tinggi tanpa perlu mencarinya secara manual.

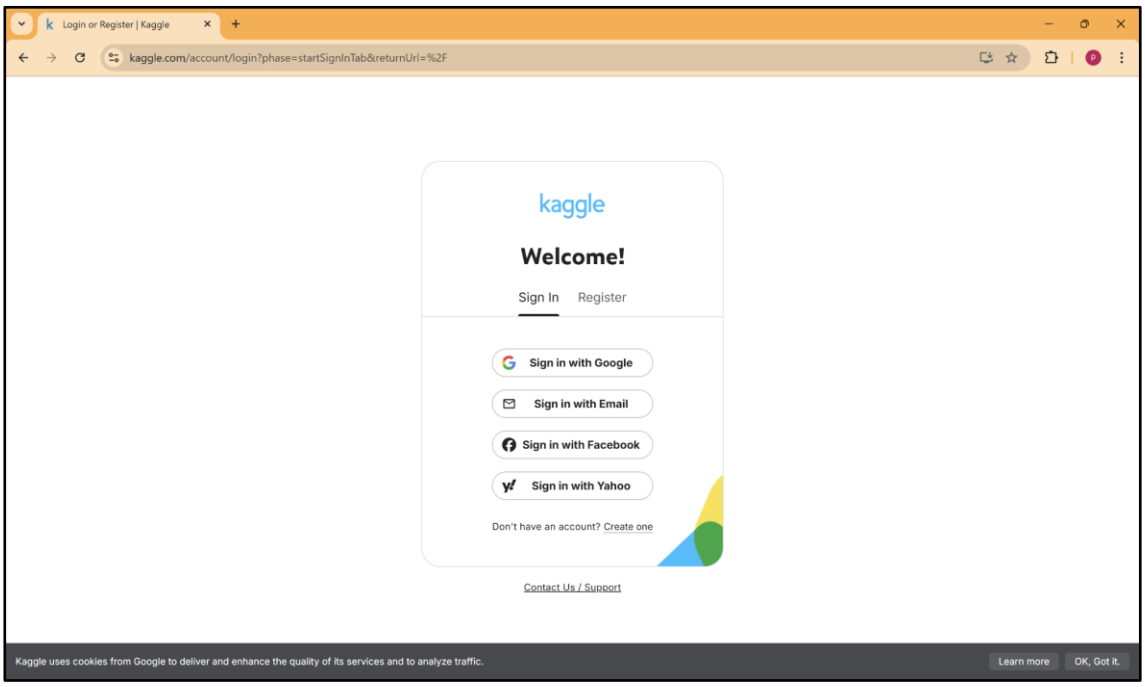
## F. Mendaftar dan Memulai Kompetisi pada Kaggle

Untuk mengakses Kaggle, perlu mendaftarkan diri terlebih dahulu, berikut tutorial mendaftar di Kaggle:

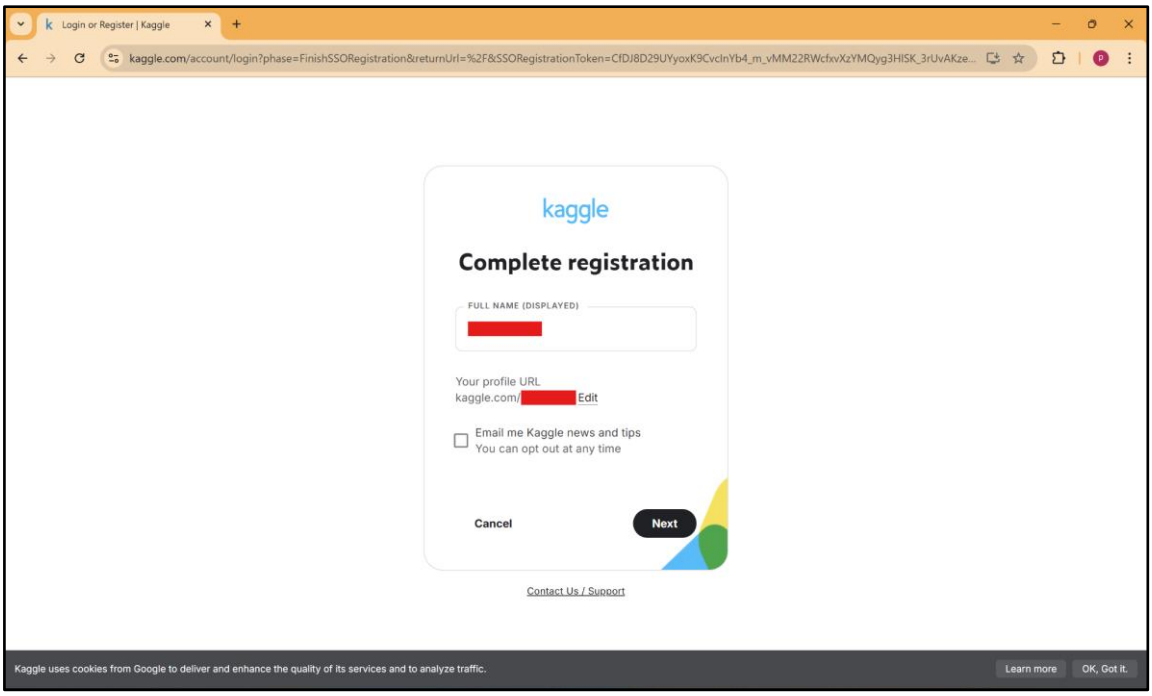
1. Buka Situs Resmi Kaggle (https://www.kaggle.com/) dan Klik “**Sign Up**”




2. Pilih Metode Pendaftaran




3. Isi Informasi Diri




4. Verifikasi Akun
   Setelah berhasil membuat akun, pengguna bisa mengikuti kompetisi yang disediakan di Kaggle dengan mengakses halaman kompetisi dan dapat memilih opsi kompetisi yang tersedia.

## 1.3 Tugas Praktikum

1. *Install tools* Pembelajaran Mesin berbasis bahasa pemrograman Python (Anda dapat menggunakan Google Collab/VS Code/Jupiter Notebook/sejenisnya). Pastikan *library* dasar seperti Pandas, NumPy, SciPy, Matplotlib, dan Scikit-Learn sudah terpasang dan dapat digunakan. Lampirkan bukti bahwa environment pemrograman telah siap dan masing-masing *library* yang dibutuhkan telah terinstal dan siap digunakan.
2. Buatlah sebuah program sederhana untuk mengimpor data sensor akselerometer berformat *.csv (*comma separated value*), kemudian *plot* data x, y, dan z dalam 1 grafik dengan warna yang berbeda, dimana aksisnya berupa data dan ordinatnya adalah nilai vibrasi arah x, y, dan z. Pilih salah satu kecepatan rotasi (variabel pctid) dari 17 variasi kecepatan yang ada*. Jangan lupa berikan judul untuk grafik yang dibangun.
   *) Metode *indexing* dan *slicing* array dapat dipelajari lebih jauh melalui laman berikut https://numpy.org/doc/stable/user/absolute_beginners.html
3. Buatlah sebuah program untuk membuat dataset yang terdiri atas 5 variabel dan 1 output kemudian plot data tersebut menggunakan 3 tipe visualisasi yang berbeda.
4. Buatlah akun pada Kaggle hingga berhasil dengan format NIM_Nama Anda!
5. Akseslah pembelajaran mengenai penggunaan Python pada Kaggle pada *link* berikut:

https://www.kaggle.com/learn/python

https://www.kaggle.com/learn/pandas

Ikuti hingga selesai kursus yang ada beserta latihannya. *Screen shoot* sertifikat penyelesaiannya!

## 1.4 Referensi

Auffarth, Ben, 2021, "*Machine Learning for Time-Series with Python*", PACKT Publishing.

Bilogur, Aleksey, 2025, Kaggle Course: Pandas.

Dangeti, P., 2017, "*Statistics for Machine Learning*", PACKT Publishing

Gopal Sakarkar, Gaurav Patil, dan Prateek Dutta, 2021, "*Machine Learning Algorithms Using Python Programming*", Nova Science Publishers, Inc.

Morris, Colin, 2025, Kaggle Course: Python.