TUGAS PERTEMUAN 1

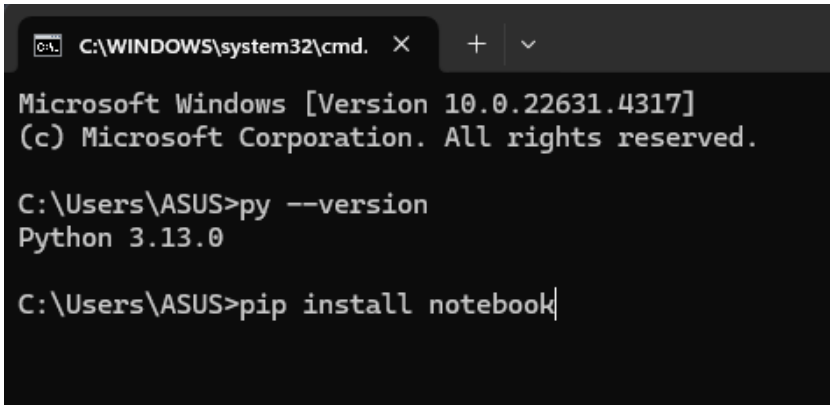
MACHINE LEARNING

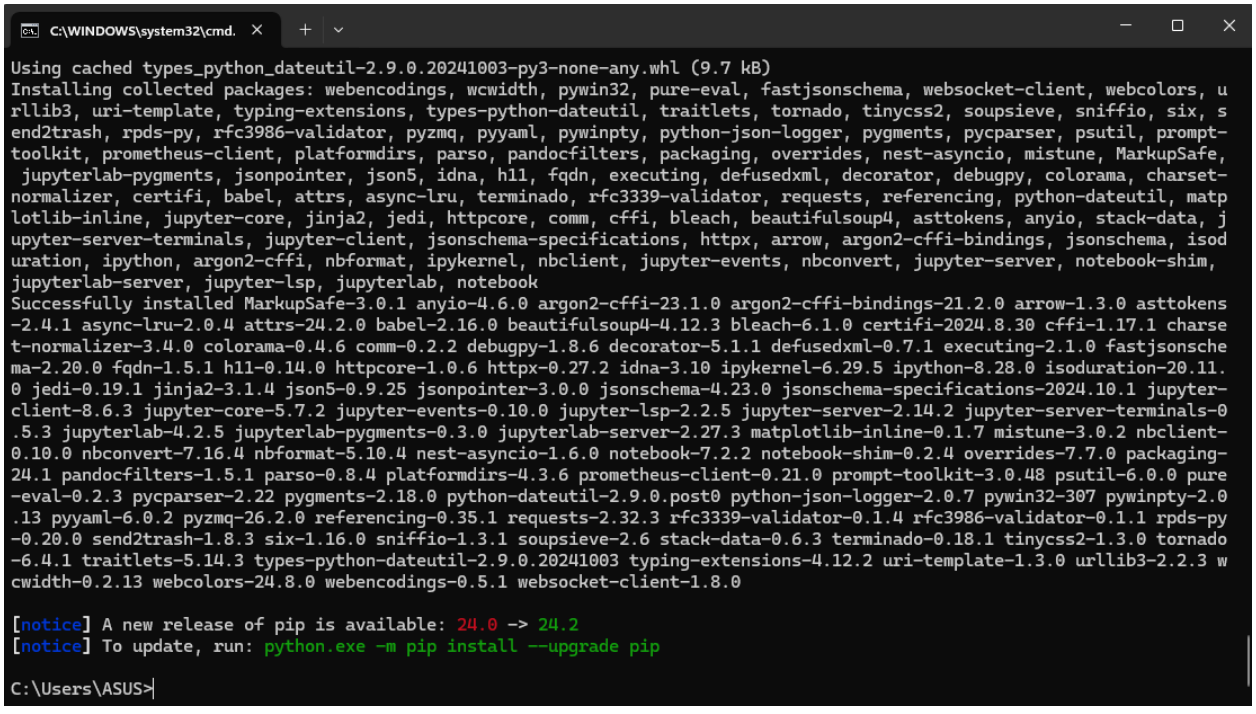
Nama : **Bima Rizki Widiatna**

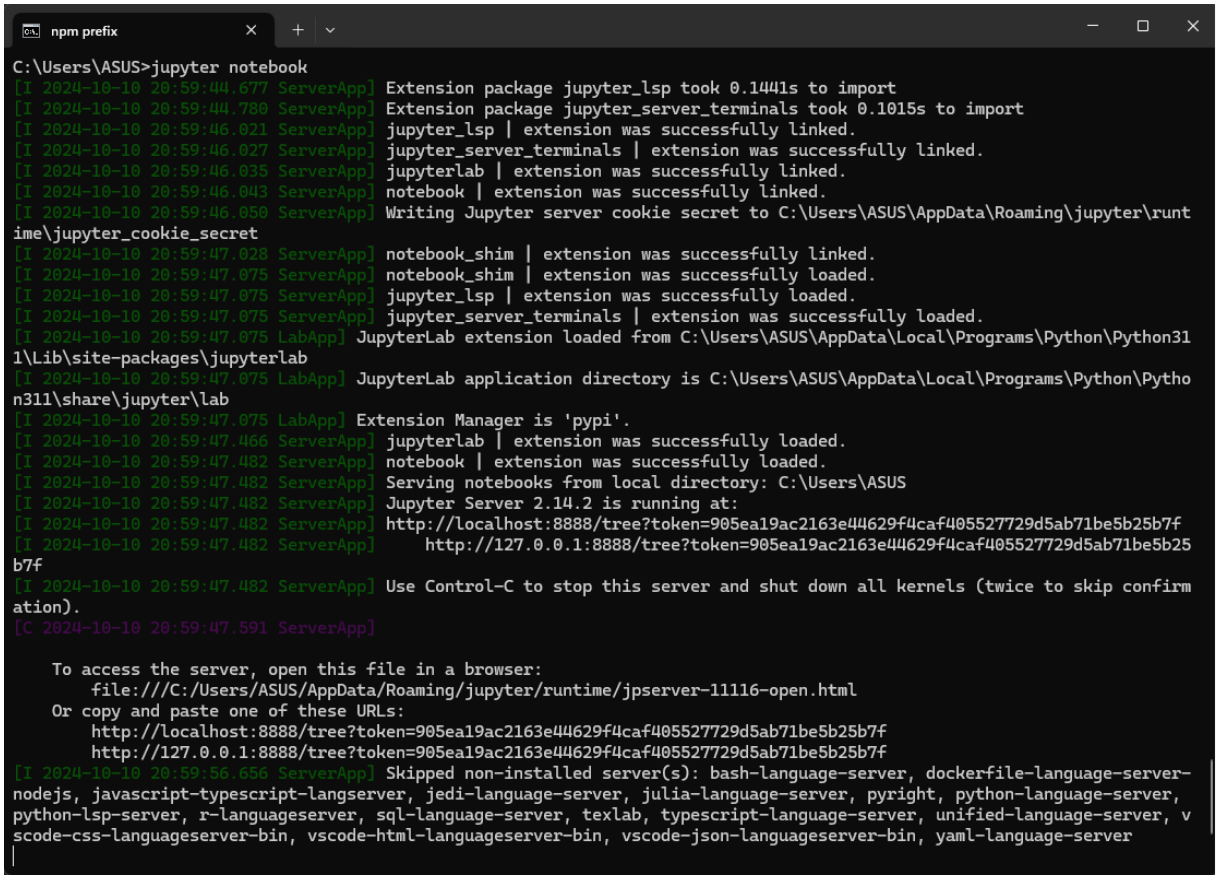
NPM : **41155050210061**

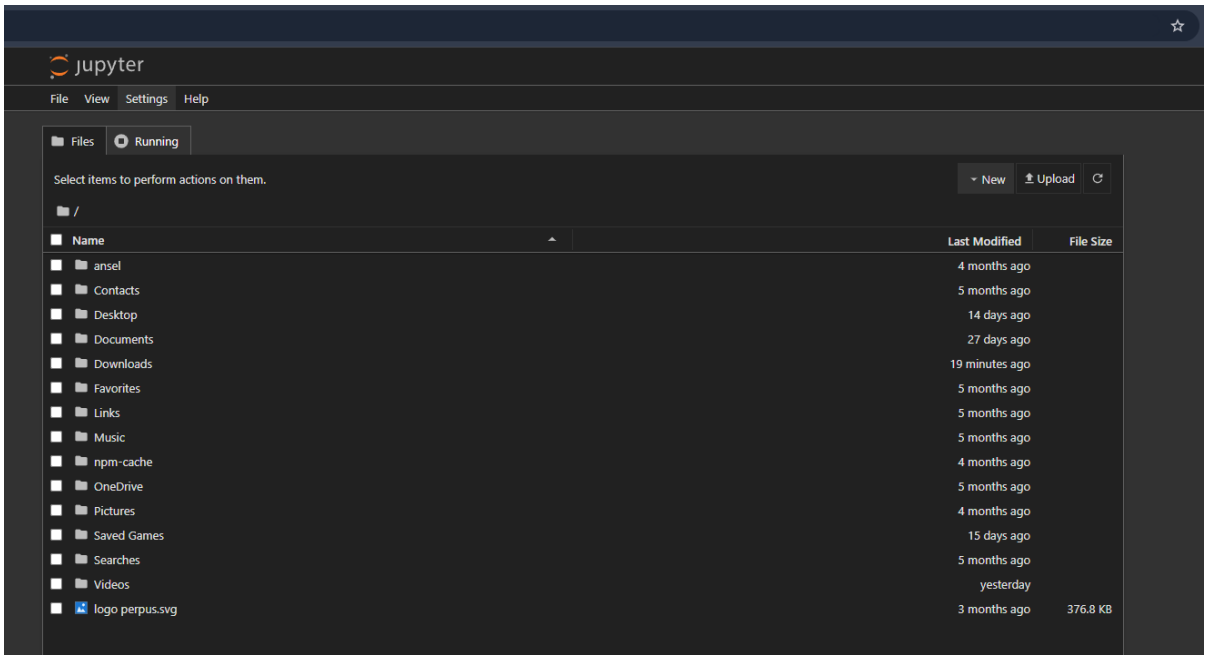
Prodi/Kelas : **Teknik Informatika/A2**

1. Instalasi Jupiter Notebook



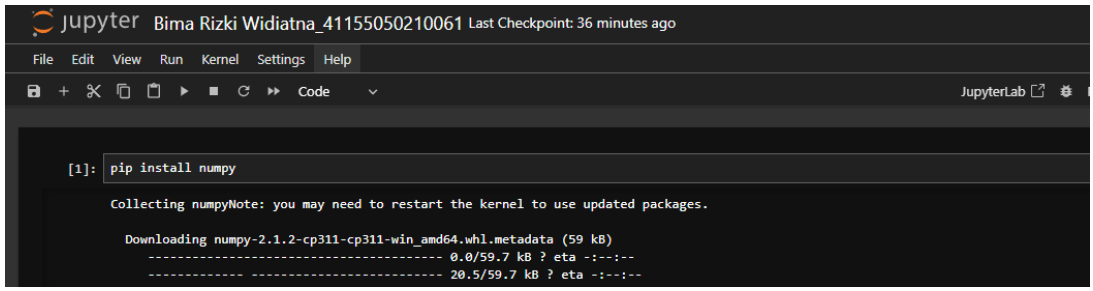


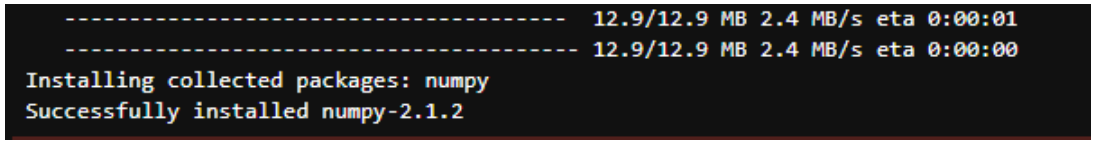

npm prefix
C:\Users\ASUS>jupyter notebook
[I 2024-10-10 20:59:44.677 ServerApp] Extension package jupyter_lsp took 0.1441s to import
[I 2024-10-10 20:59:44.780 ServerApp] Extension package jupyter_server_terminals took 0.1015s to import
[I 2024-10-10 20:59:46.021 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-10-10 20:59:46.027 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-10-10 20:59:46.035 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-10-10 20:59:46.043 ServerApp] notebook | extension was successfully linked.
[I 2024-10-10 20:59:46.050 ServerApp] Writing Jupyter server cookie secret to C:\Users\ASUS\AppData\Roaming\jupyter\runtime\jupyter_cookie_secret
[I 2024-10-10 20:59:47.028 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-10-10 20:59:47.075 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-10-10 20:59:47.075 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-10-10 20:59:47.075 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-10-10 20:59:47.075 LabApp] JupyterLab extension loaded from C:\Users\ASUS\AppData\Local\Programs\Python\Python311\Lib\site-packages\jupyterlab
[I 2024-10-10 20:59:47.075 LabApp] JupyterLab application directory is C:\Users\ASUS\AppData\Local\Programs\Python\Python311\share\jupyter\lab
[I 2024-10-10 20:59:47.075 LabApp] Extension Manager is 'pypi'.
[I 2024-10-10 20:59:47.466 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-10-10 20:59:47.482 ServerApp] notebook | extension was successfully loaded.
[I 2024-10-10 20:59:47.482 ServerApp] Serving notebooks from local directory: C:\Users\ASUS
[I 2024-10-10 20:59:47.482 ServerApp] Jupyter Server 2.14.2 is running at:
[I 2024-10-10 20:59:47.482 ServerApp] http://localhost:8888/tree?token=905ea19ac2163e44629f4caf405527729d5ab71be5b25b7f
[I 2024-10-10 20:59:47.482 ServerApp] http://127.0.0.1:8888/tree?token=905ea19ac2163e44629f4caf405527729d5ab71be5b25b7f
[I 2024-10-10 20:59:47.482 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2024-10-10 20:59:47.591 ServerApp]
To access the server, open this file in a browser:
file:///C:/Users/ASUS/AppData/Roaming/jupyter/runtime/jpserver-11116-open.html
Or copy and paste one of these URLs:
http://localhost:8888/tree?token=905ea19ac2163e44629f4caf405527729d5ab71be5b25b7f
http://127.0.0.1:8888/tree?token=905ea19ac2163e44629f4caf405527729d5ab71be5b25b7f
[I 2024-10-10 20:59:56.656 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langserver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server


jupyter
File View Settings Help
Files Running
Select items to perform actions on them.
New Upload
/
Name Last Modified File Size
ansel 4 months ago
Contacts 5 months ago
Desktop 14 days ago
Documents 27 days ago
Downloads 19 minutes ago
Favorites 5 months ago
Links 5 months ago
Music 5 months ago
npm-cache 4 months ago
OneDrive 5 months ago
Pictures 4 months ago
Saved Games 15 days ago
Searches 5 months ago
Videos yesterday
logo perpus.svg 3 months ago 376.8 KB

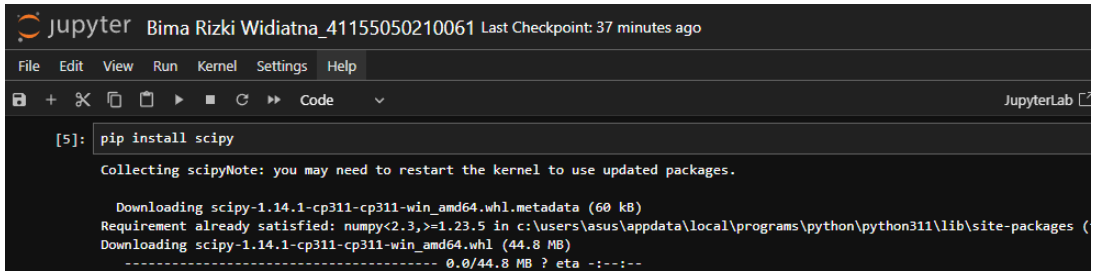
1.1. Download dan Instalasi Library python seperti NumPy, SciPy, Pandas, Matplotlib, Seaborn, Scikit-learn.

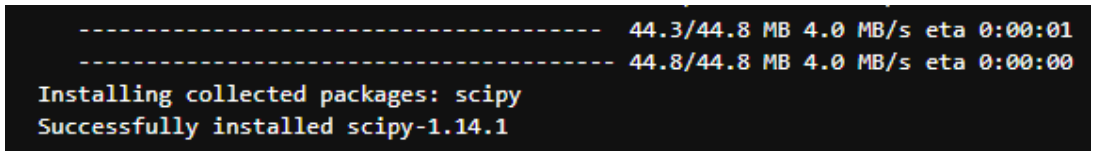
1.1.1. Instalasi Library *numpy*



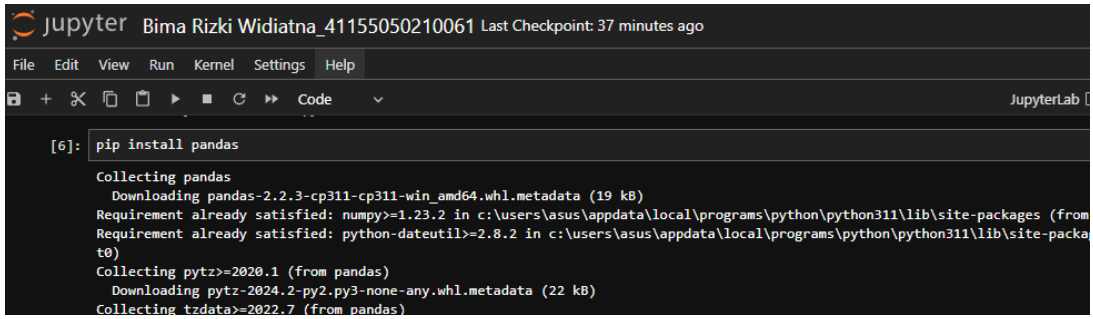


1.1.2. Instalasi Library *scipy*



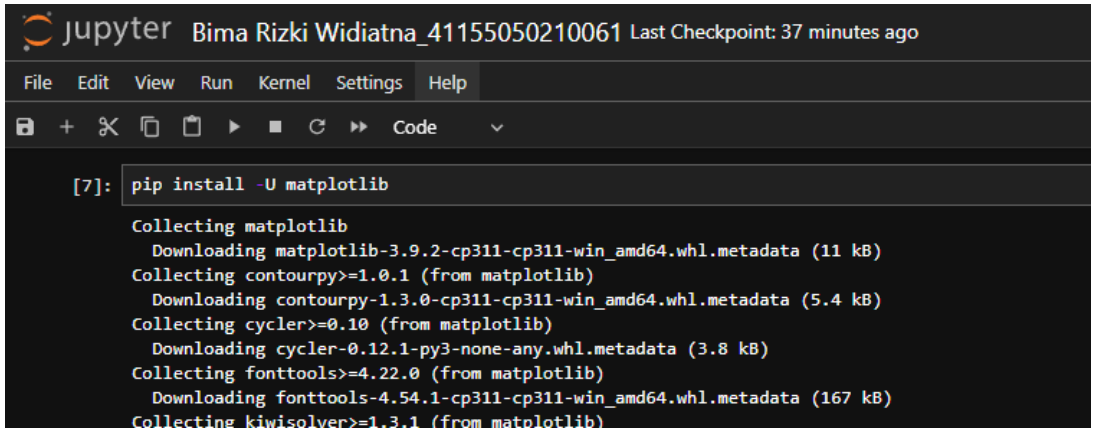


1.1.3. Instalasi Library *pandas*



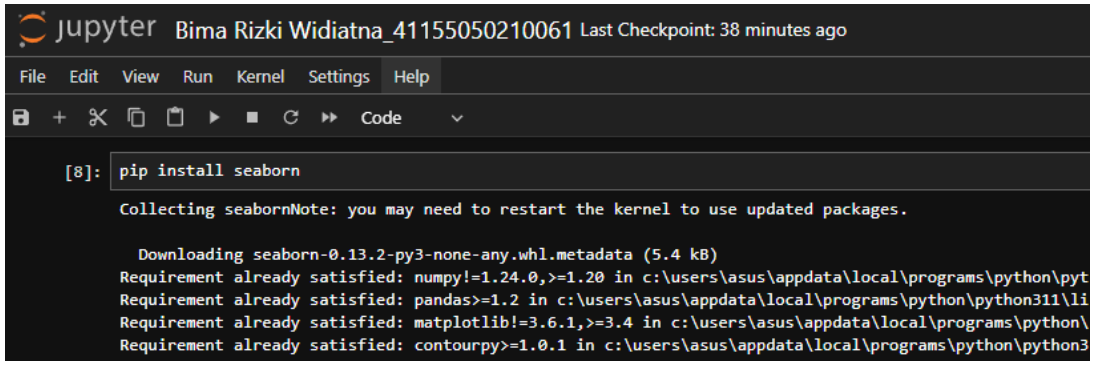
```
   ---------------------------------------- -- 11.0/11.6 MB 2.3 MB/s eta 0:00:01
   ---------------------------------------- 11.6/11.6 MB 2.3 MB/s eta 0:00:00
Downloading pytz-2024.2-py2.py3-none-any.whl (508 kB)
Downloading tzdata-2024.2-py2.py3-none-any.whl (346 kB)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.2.3 pytz-2024.2 tzdata-2024.2
```

1.1.4. Instalasi Library *matplotlib*



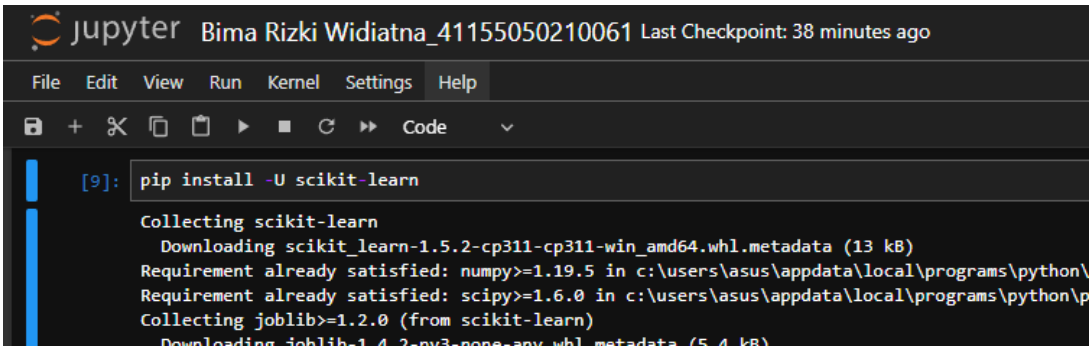

```
   ---------------------------------------- --- 2.4/2.6 MB 3.8 MB/s eta 0:00:01
   ---------------------------------------- 2.6/2.6 MB 3.8 MB/s eta 0:00:00
Downloading pyparsing-3.1.4-py3-none-any.whl (104 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler, contourpy, matplotlib
Successfully installed contourpy-1.3.0 cycler-0.12.1 fonttools-4.54.1 kiwisolver-1.4.7 matplotlib-3.9.2 pillow
```

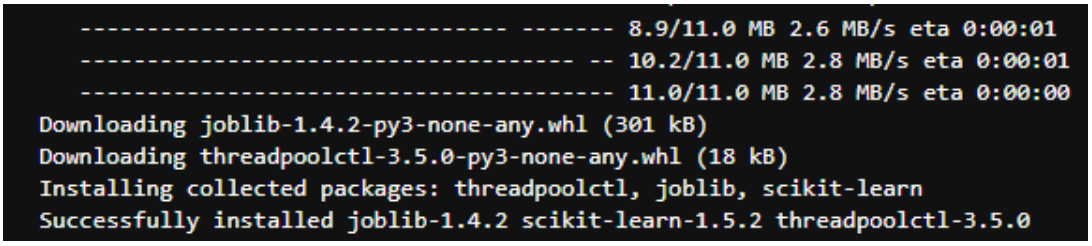
1.1.5. Instalasi Library *seaborn*



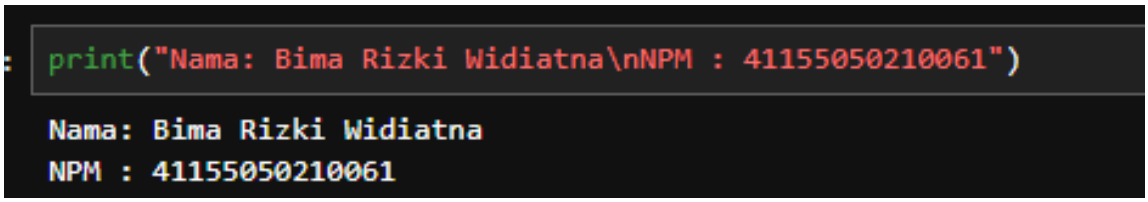

```
Requirement already satisfied: tzdata>=2022.7 in c:\users\asus\ap
4.2)
Requirement already satisfied: six>=1.5 in c:\users\asus\appdata\
b!=3.6.1,>=3.4->seaborn) (1.16.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
```

1.1.6. Instalasi Library *scikit-learn*



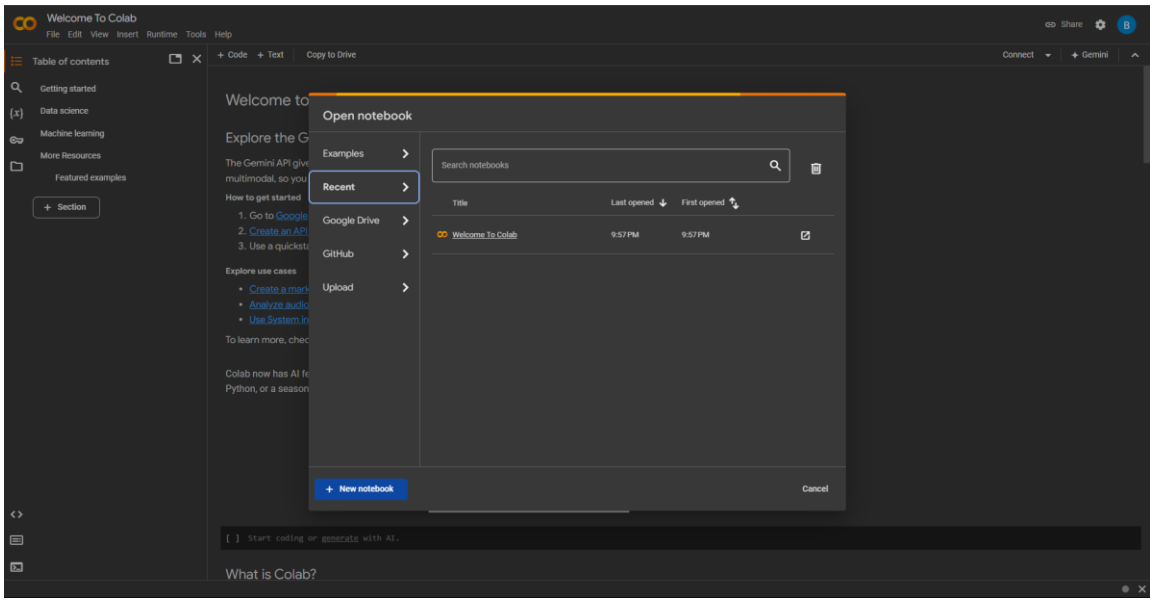


1.2. Tuliskan nama dan nomor NPM anda pada Jupiter Notebook.



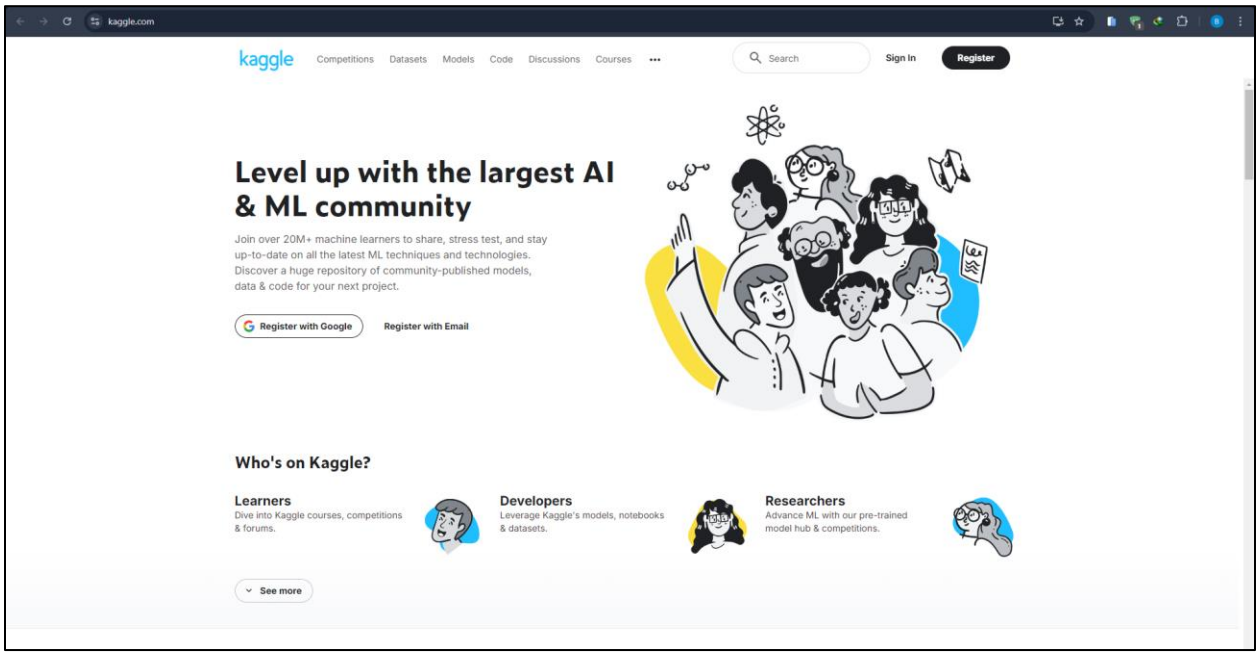
2. Menggunakan *Google Colab*

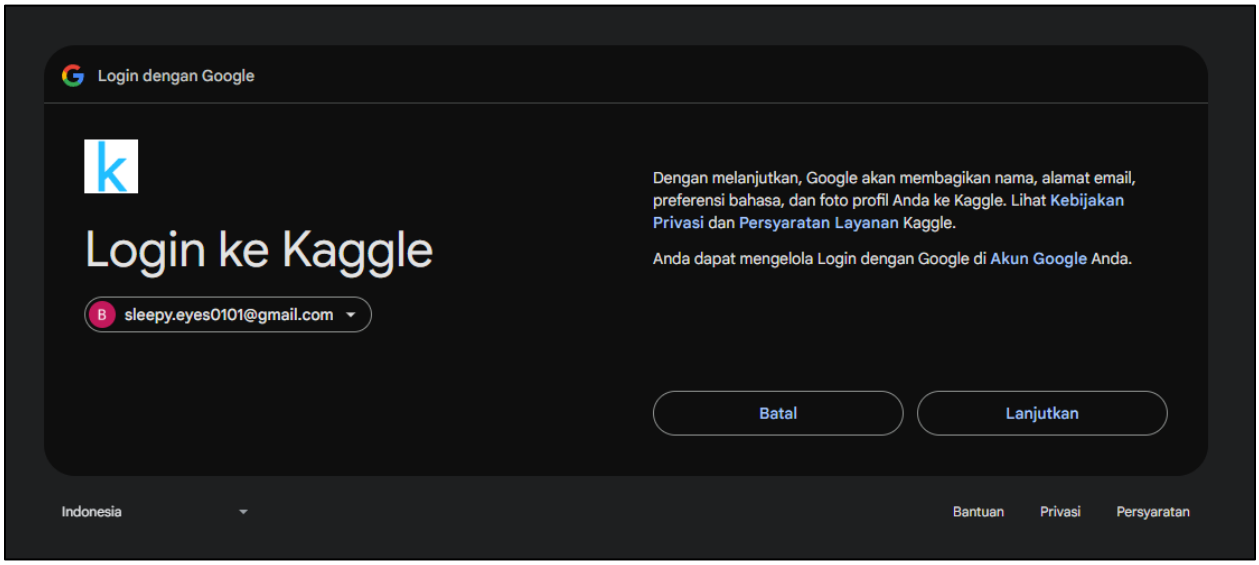
2.1. Gunakan *Google Colab*

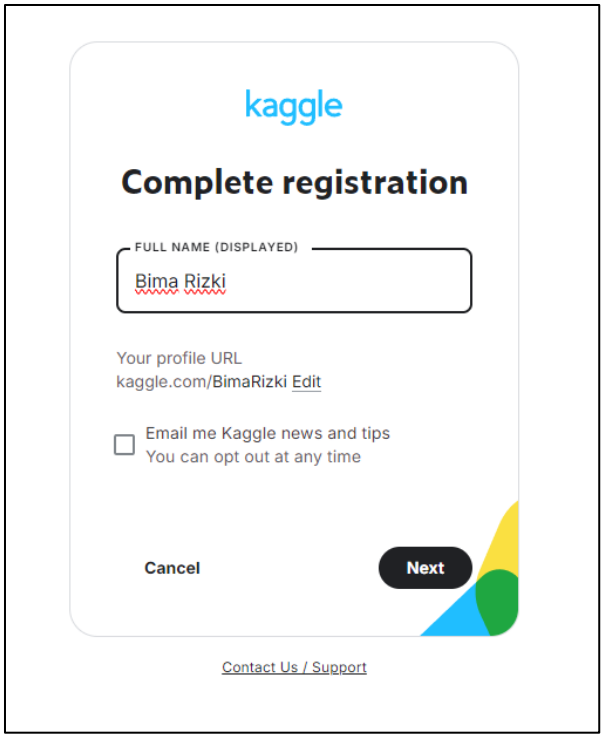


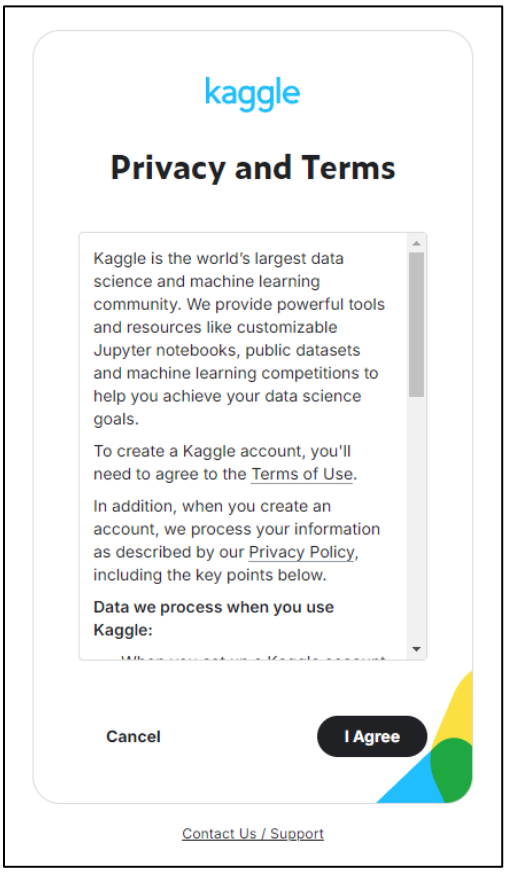
2.2. Tuliskan nama dan nomor NPM anda pada *Google Colab*.

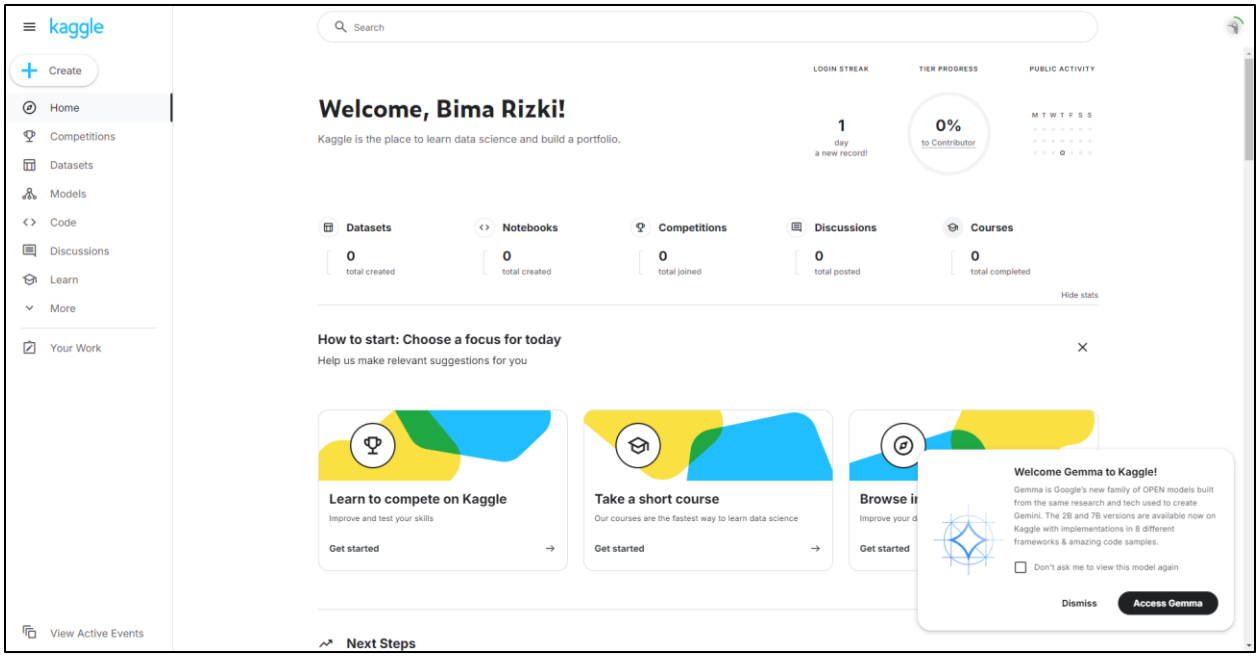
3. Buat Akun *Kaggle*



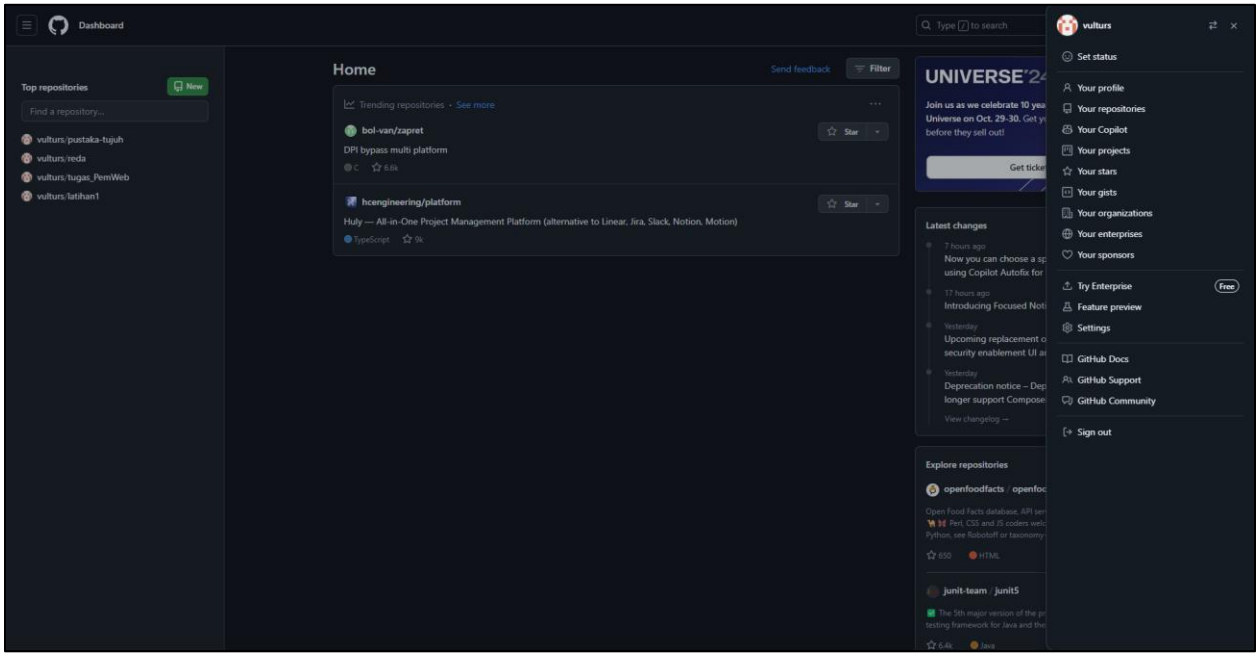



kaggle
Complete registration
FULL NAME (DISPLAYED)
Bima Rizki
Your profile URL
kaggle.com/BimaRizki Edit
Email me Kaggle news and tips
You can opt out at any time
Cancel
Next
Contact Us / Support


kaggle
Privacy and Terms
Kaggle is the world's largest data science and machine learning community. We provide powerful tools and resources like customizable Jupyter notebooks, public datasets and machine learning competitions to help you achieve your data science goals.
To create a Kaggle account, you'll need to agree to the Terms of Use.
In addition, when you create an account, we process your information as described by our Privacy Policy, including the key points below.
Data we process when you use Kaggle:
Cancel
I Agree
Contact Us / Support



4. Buat Akun *Github*



5. Praktek Sample Dataset pada *Scikit-Learn*

5.1. Load Sample Dataset

```
from sklearn.datasets import load_iris

iris = load_iris()
iris
```

```
[12]:


{'data': array([[5.1, 3.5, 1.4, 0.2],
        [4.9, 3. , 1.4, 0.2],
        [4.7, 3.2, 1.3, 0.2],
        [4.6, 3.1, 1.5, 0.2],
        [5. , 3.6, 1.4, 0.2],
        [5.4, 3.9, 1.7, 0.4],
        [4.6, 3.4, 1.4, 0.3],
        [5. , 3.4, 1.5, 0.2],
        [4.4, 2.9, 1.4, 0.2],
        [4.9, 3.1, 1.5, 0.1],
        [5.4, 3.7, 1.5, 0.2],
        [4.8, 3.4, 1.6, 0.2],
        [4.8, 3. , 1.4, 0.1],
        [4.3, 3. , 1.1, 0.1],
```

```
iris.keys()
```

```
[13]:

dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_
module'])
```

Perintah *iris.keys()* digunakan untuk mengetahui *key* apa saja yang terdapat pada *dictionary iris*,

5.2. Metadata | Deskripsi dari sample dataset

```
[14]:
print(iris.DESCR)
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

:Number of Instances: 150 (50 in each of three classes)
:Number of Attributes: 4 numeric, predictive attributes and the class
:Attribute Information:
    - sepal length in cm
    - sepal width in cm
    - petal length in cm
    - petal width in cm
    - class:
            - Iris-Setosa
            - Iris-Versicolour
            - Iris-Virginica

:Summary Statistics:

============== ==== ==== ======= ===== ====================
                Min  Max   Mean    SD   Class Correlation
============== ==== ==== ======= ===== ====================
sepal length:   4.3  7.9   5.84   0.83    0.7826
sepal width:    2.0  4.4   3.05   0.43   -0.4194
petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
============== ==== ==== ======= ===== ====================

:Missing Attribute Values: None
:Class Distribution: 33.3% for each of 3 classes.
:Creator: R.A. Fisher
:Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
:Date: July, 1988

The famous Iris database, first used by Sir R.A. Fisher. The dataset is taken
from Fisher's paper. Note that it's the same as in R, but not as in the UCI
Machine Learning Repository, which has two wrong data points.

This is perhaps the best known database to be found in the
pattern recognition literature.  Fisher's paper is a classic in the field and
is referenced frequently to this day.  (See Duda & Hart, for example.)  The
data set contains 3 classes of 50 instances each, where each class refers to a
type of iris plant.  One class is linearly separable from the other 2; the
latter are NOT linearly separable from each other.

.. dropdown:: References

  - Fisher, R.A. "The use of multiple measurements in taxonomic problems"
    Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to
```

```
      Mathematical Statistics" (John Wiley, NY, 1950).
    - Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis.
      (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 218.
    - Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System
      Structure and Classification Rule for Recognition in Partially Exposed
      Environments".  IEEE Transactions on Pattern Analysis and Machine
      Intelligence, Vol. PAMI-2, No. 1, 67-71.
    - Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".  IEEE Transactions
      on Information Theory, May 1972, 431-433.
    - See also: 1988 MLC Proceedings, 54-64.  Cheeseman et al"s AUTOCLASS II
      conceptual clustering system finds 3 classes in the data.
    - Many, many more ...
```

5.3. Explanatory & Response Variables | Features & Target

5.3.1. Explanatory Variables (Features)

```
[16]:
X = iris.data
X.shape
# X

[16]:

(150, 4)
```

```
[17]:
X = iris.data
# X.shape
X

[17]:


array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
```

5.3.2. Response Variables (Target)

```
[20]:
y = iris.target
y.shape
# y

[20]:

(150,)
```

```
[21]:
y = iris.target
# y.shape
y

[21]:

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

5.3.3. Feature & Target Names

```
[22]:
feature_names = iris.feature_names
feature_names

[22]:

['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

```
[23]:
target_names = iris.target_names
target_names

[23]:

array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

5.3.4. Visualisasi Data

[25]:

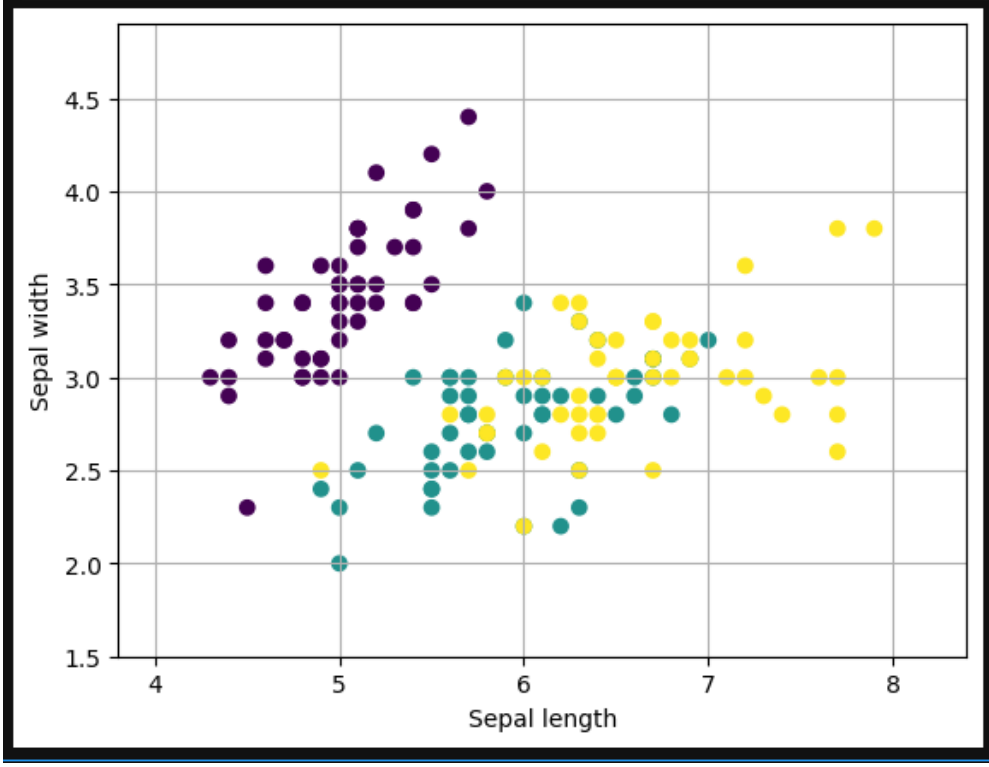
```
import matplotlib.pyplot as plt

X = X[:, :2]

x_min, x_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5
y_min, y_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5

plt.scatter(X[:, 0], X[:, 1], c=y)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')

plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.grid(True)
plt.show()
```



5.3.5. Training Set & Testing Set

[28]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=1)

print(f'X train: {X_train.shape}')
print(f'X test: {X_test.shape}')
print(f'y train: {y_train.shape}')
print(f'y test: {y_test.shape}')
```

```
X train: (105, 2)
X test: (45, 2)
y train: (105,)
y test: (45,)
```

5.3.6. Load Sample Dataset Sebagai *Pandas Data Frame*

[29]:

```
iris = load_iris(as_frame=True)

iris_features_df = iris.data
iris_features_df
```

[29]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

6. Praktek *Machine Learning Workflow* dengan *Scikit-Learn*

6.1. Persiapan Dataset | *Loading & Splitting Dataset*

[1]:

```
from sklearn.datasets import load_iris

iris = load_iris()

X = iris.data
y = iris.target
```

[2]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.4,
                                                    random_state=1)
```

6.2. Training Model *Machine Learning*

[3]:

```
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
```

[3]:

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

6.3. Evaluasi Model *Machine Learning*

[4]:

```
from sklearn.metrics import accuracy_score

y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print(f'Accuracy: {acc}')
```

```
Accuracy: 0.9833333333333333
```

[ ]:

6.4. Pemanfaatan *Trained Model Machine Learning*

[5]:

```
data_baru = [[5, 5, 3, 2],
             [2, 4, 3, 5]]

preds = model.predict(data_baru)
preds
```

[5]:

```
array([1, 2])
```

[6]:

```
pred_species = [iris.target_names[p] for p in preds]
print(f'Hasil Prediksi: {pred_species}')
```

```
Hasil Prediksi: [np.str_('versicolor'), np.str_('virginica')]
```

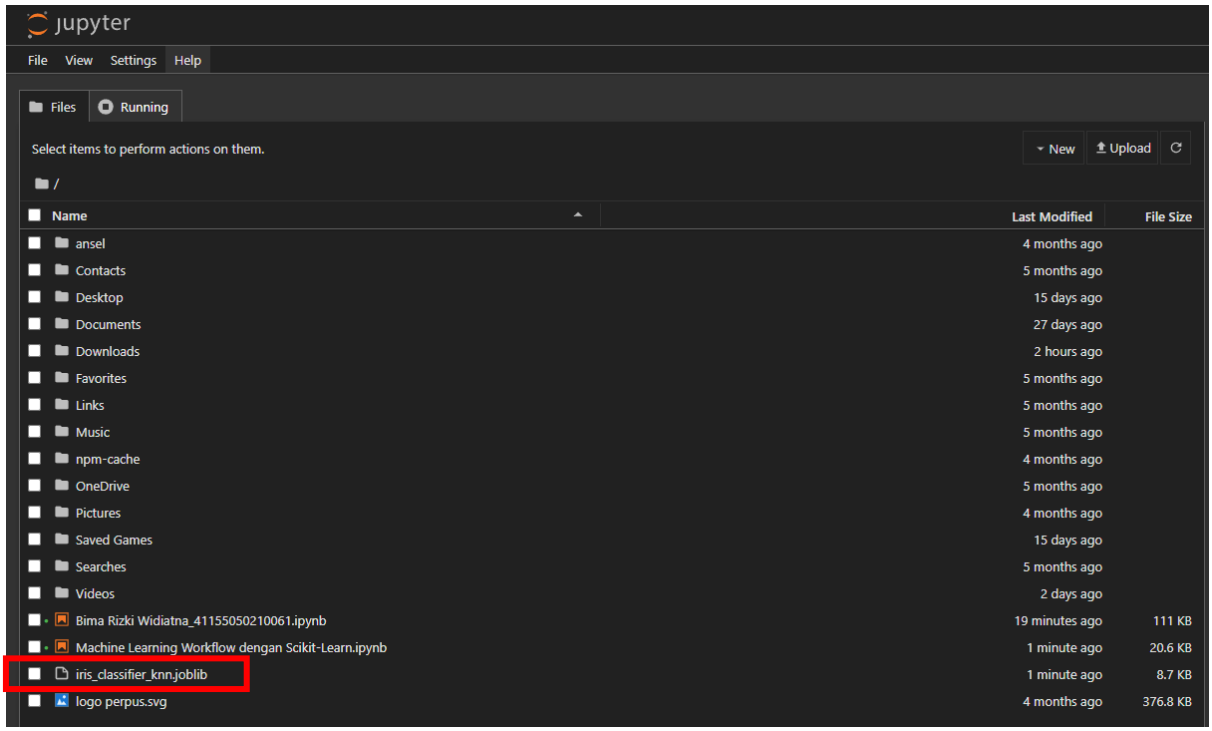
6.5. Deploy Model *Machine Learning* | *Dumping* dan *Loading* Model *Machine Learning*

[7]:

```
import joblib
joblib.dump(model, 'iris_classifier_knn.joblib')
```

[7]:

```
['iris_classifier_knn.joblib']
```



```
[8]:
production_model = joblib.load('iris_classifier_knn.joblib')
```

7. Praktek *Data Preprocessing* dengan *Scikit-Learn*

7.1. Persiapan Sample Dataset

```
[1]:
import numpy as np
from sklearn import preprocessing

sample_data = np.array([[2.1, -1.9, 5.5],
                        [-1.5, 2.4, 3.5],
                        [0.5, -7.9, 5.6],
                        [5.9, 2.3, -5.8]])

sample_data
```

```
[1]:

array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

```
[2]:
sample_data.shape
```

```
[2]:

(4, 3)
```

7.2. Teknik *Data Preprocessing 1: Binarisation*

```
[3]:
sample_data
```

```
[3]:

array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

```
[4]:
preprocessor = preprocessing.Binarizer(threshold=0.5)
binarized_data = preprocessor.transform(sample_data)
binarized_data

[4]:
array([[1., 0., 1.],
       [0., 1., 1.],
       [0., 0., 1.],
       [1., 1., 0.]])
```

7.3. Teknik *Data Preprocessing 2: Scaling*

```
[5]:
sample_data

[5]:
array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

```
[7]:
preprocessor = preprocessing.MinMaxScaler(feature_range=(0, 1))
preprocessor.fit(sample_data)
scaled_data = preprocessor.transform(sample_data)
scaled_data

[7]:
array([[0.48648649, 0.58252427, 0.99122807],
       [0.        , 1.        , 0.81578947],
       [0.27027027, 0.        , 1.        ],
       [1.        , 0.99029126, 0.        ]])
```

```
[9]:
scaled_data = preprocessor.fit_transform(sample_data)
scaled_data

[9]:
array([[0.48648649, 0.58252427, 0.99122807],
       [0.        , 1.        , 0.81578947],
       [0.27027027, 0.        , 1.        ],
       [1.        , 0.99029126, 0.        ]])
```

7.4. Teknik *Data Preprocessing 3: Normalisation*

7.4.1. Normalisasi L1

```
[10]:
sample_data

[10]:
array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

```
[11]:
l1_normalised_data = preprocessing.normalize(sample_data, norm='l1')
l1_normalised_data

[11]:
array([[ 0.22105263, -0.2       ,  0.57894737],
       [-0.2027027 ,  0.32432432,  0.47297297],
       [ 0.03571429, -0.56428571,  0.4       ],
       [ 0.42142857,  0.16428571, -0.41428571]])
```

7.4.2. Normalisasi L2

```
[12]:
sample_data

[12]:
array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

```
[13]:
l2_normalised_data = preprocessing.normalize(sample_data, norm='l2')
l2_normalised_data

[13]:
array([[ 0.33946114, -0.30713151,  0.88906489],
       [-0.33325106,  0.53320169,  0.7775858 ],
       [ 0.05156558, -0.81473612,  0.57753446],
       [ 0.68706914,  0.26784051, -0.6754239 ]])
```