

Nama : Deni Purwanto

NPM : 41155050210017

Kelas : INF A

Mata Kuliah : Machine Learning

Tugas Pertemuan 1 Jawaban Nomor 1

1. Instalasi Jupyter Noterbook, Lakukan download dan instalasi:

- Instalasi Jupyter Notebook

```
Command Prompt - pip insta x + v
Downloading jupyterlab_server-2.27.3-py3-none-any.whl (59 kB)
 59.7/59.7 kB 3.3 MB/s eta 0:00:00
Downloading jupyterlab_widgets-3.0.13-py3-none-any.whl (214 kB)
 214.4/214.4 kB 4.3 MB/s eta 0:00:00
Downloading MarkupSafe-2.1.5-cp312-cp312-win_amd64.whl (17 kB)
Downloading matplotlib_inline-0.1.7-py3-none-any.whl (9.9 kB)
Downloading mistune-3.0.2-py3-none-any.whl (47 kB)
 48.0/48.0 kB 1.2 MB/s eta 0:00:00
Downloading nbclient-0.10.0-py3-none-any.whl (25 kB)
Downloading nbformat-5.10.4-py3-none-any.whl (78 kB)
 78.5/78.5 kB 4.5 MB/s eta 0:00:00
Downloading notebook_shim-0.2.4-py3-none-any.whl (13 kB)
Downloading packaging-24.1-py3-none-any.whl (53 kB)
 54.0/54.0 kB 2.7 MB/s eta 0:00:00
Downloading pandocfilters-1.5.1-py2.py3-none-any.whl (8.7 kB)
Downloading prompt_toolkit-3.0.48-py3-none-any.whl (386 kB)
 386.6/386.6 kB 6.0 MB/s eta 0:00:00
Downloading pygments-2.18.0-py3-none-any.whl (1.2 MB)
 1.2/1.2 MB 5.5 MB/s eta 0:00:00
Downloading pyzmq-26.2.0-cp312-cp312-win_amd64.whl (637 kB)
 637.8/637.8 kB 5.0 MB/s eta 0:00:00
Downloading setuptools-75.1.0-py3-none-any.whl (1.2 MB)
 1.2/1.2 MB 4.7 MB/s eta 0:00:00
Downloading tornado-6.4.1-cp38-abi3-win_amd64.whl (438 kB)
 438.5/438.5 kB 3.1 MB/s eta 0:00:00
Downloading traitlets-5.14.3-py3-none-any.whl (85 kB)
 85.4/85.4 kB 2.4 MB/s eta 0:00:00
Downloading widgetsnbextension-4.0.13-py3-none-any.whl (2.3 MB)
 2.3/2.3 MB 3.0 MB/s eta 0:00:00
Downloading beautifulsoup4-4.12.3-py3-none-any.whl (147 kB)
```

- Download Library NumPy

```
Command Prompt - pip insta x + v
Microsoft Windows [Version 10.0.22631.4249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>pip install numpy
Collecting numpy
  Downloading numpy-2.1.1-cp312-cp312-win_amd64.whl.metadata (59 kB)
    Downloading numpy-2.1.1-cp312-cp312-win_amd64.whl (12.6 MB)
      12.6/12.6 MB 3.4 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-2.1.1
```

- Download Library SciPy

```
C:\Users\ASUS>pip install scipy
Collecting scipy
  Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: numpy<2.3, >=1.23.5 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from scipy) (2.1.1)
  Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl (44.5 MB)
    44.5/44.5 MB 4.5 MB/s eta 0:00:00
Installing collected packages: scipy
Successfully installed scipy-1.14.1
```

- **Download Library Pandas**

```
C:\Users\ASUS>pip install pandas
Collecting pandas
  Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.1.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2024.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl (11.5 MB)
 11.5/11.5 MB 2.6 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
```

- **Download Library Matplotlib**

```
C:\Users\ASUS>pip install -U matplotlib
Collecting matplotlib
  Downloading matplotlib-3.9.2-cp312-cp312-win_amd64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.0-cp312-cp312-win_amd64.whl.metadata (5.4 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.54.1-cp312-cp312-win_amd64.whl.metadata (167 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.7-cp312-cp312-win_amd64.whl.metadata (6.4 kB)
Requirement already satisfied: numpy>=1.23 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (2.1.1)
Requirement already satisfied: packaging>=20.0 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (24.1)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-10.4.0-cp312-cp312-win_amd64.whl.metadata (9.3 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.1.4-py3-none-any.whl.metadata (5.1 kB)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Downloading matplotlib-3.9.2-cp312-cp312-win_amd64.whl (7.8 MB)
 7.8/7.8 MB 4.1 MB/s eta 0:00:00
Downloading contourpy-1.3.0-cp312-cp312-win_amd64.whl (218 kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.54.1-cp312-cp312-win_amd64.whl (2.2 MB)
 2.2/2.2 MB 4.6 MB/s eta 0:00:00
Downloading kiwisolver-1.4.7-cp312-cp312-win_amd64.whl (55 kB)
Downloading pillow-10.4.0-cp312-cp312-win_amd64.whl (2.6 MB)
 2.6/2.6 MB 3.9 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-10.4.0 pyparsing-3.1.4
```

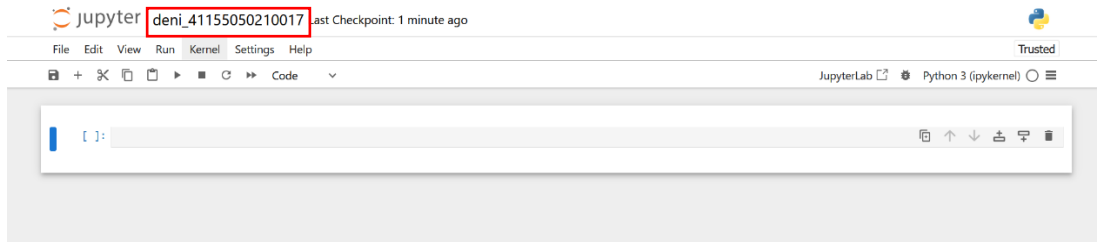
- **Download Library Seaborn**

```
C:\Users\ASUS>pip install seaborn
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from seaborn) (2.1.1)
Requirement already satisfied: pandas>=1.2 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from seaborn) (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.54.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from pandas>=1.2->seaborn) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from pandas>=1.2->seaborn) (2024.2)
Requirement already satisfied: six>=1.5 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.7->matplotlib!=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
```

- Download Library Scikit-Learn

```
C:\Users\ASUS>pip install -U scikit-learn
Collecting scikit-learn
  Downloading scikit_learn-1.5.2-cp312-cp312-win_amd64.whl.metadata (13 kB)
Requirement already satisfied: numpy>=1.19.5 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (2.1.1)
Requirement already satisfied: scipy>=1.6.0 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (1.14.1)
Collecting joblib>=1.2.0 (from scikit-learn)
  Downloading joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Downloading threadpoolctl-3.5.0-py3-none-any.whl.metadata (13 kB)
  Downloading scikit_learn-1.5.2-cp312-cp312-win_amd64.whl (11.0 MB)
    11.0/11.0 MB 2.1 MB/s eta 0:00:00
  Downloading joblib-1.4.2-py3-none-any.whl (301 kB)
  Downloading threadpoolctl-3.5.0-py3-none-any.whl (18 kB)
Installing collected packages: threadpoolctl, joblib, scikit-learn
Successfully installed joblib-1.4.2 scikit-learn-1.5.2 threadpoolctl-3.5.0
```

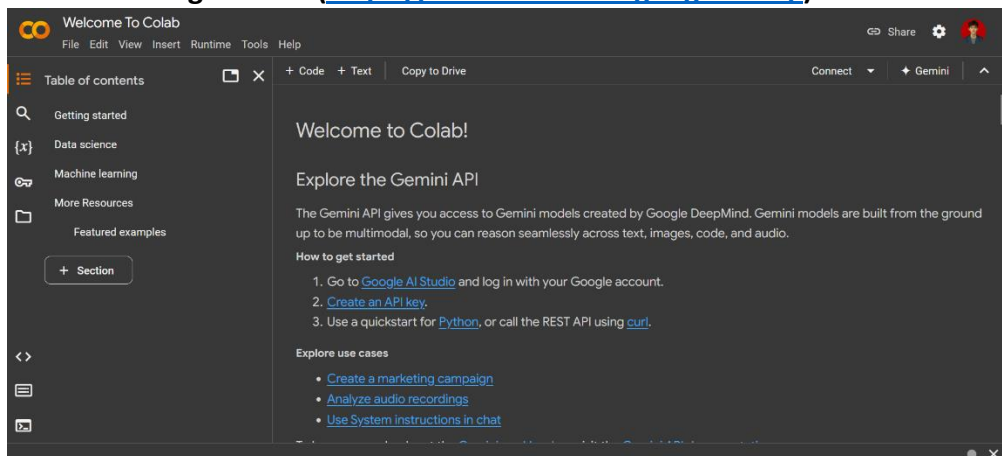
- Tuliskan nama dan nomor NPM anda pada Jupiter Notebook.



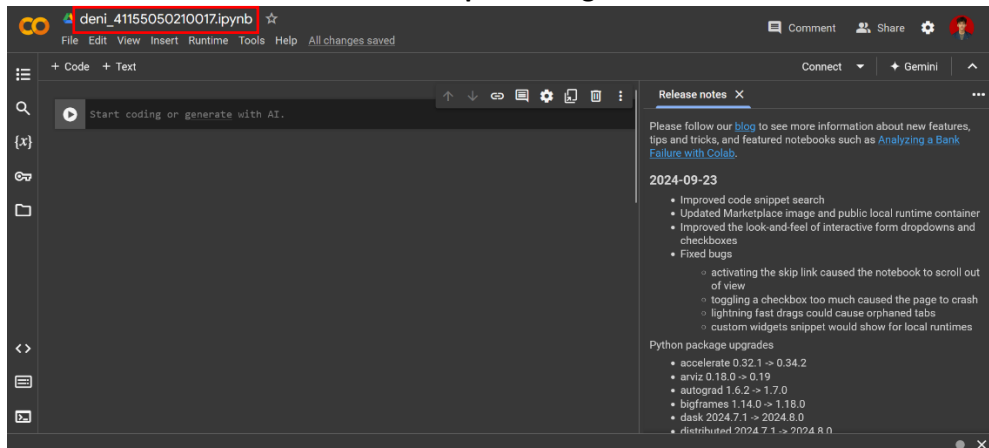
Tugas Pertemuan 1 Jawaban Nomor 2

2. Menggunakan Google Colab

- Gunakan Google Colab (<https://colab.research.google.com/>)



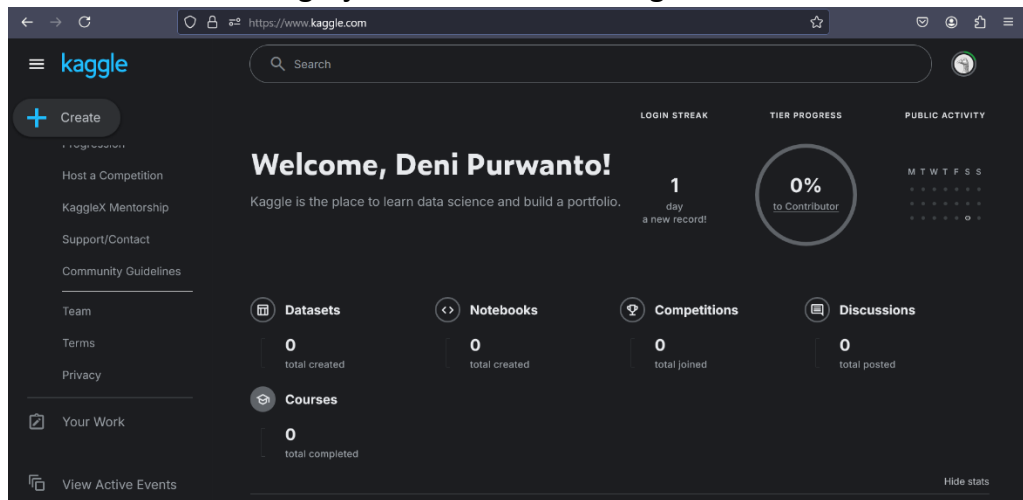
- Tuliskan nama dan nomor NPM anda pada Google Colab.



Tugas Pertemuan 1 Jawaban Nomor 3

3. Buatlah akun di <https://www.kaggle.com/>

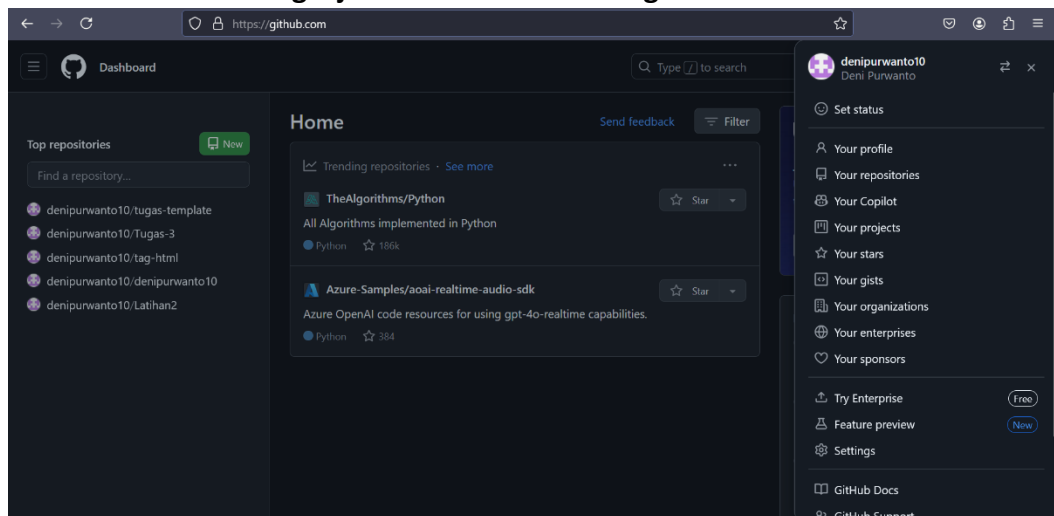
- Buat screenshot sebagai jawaban nomor 3 di Tugas Pertemuan 1



Tugas Pertemuan 1 Jawaban Nomor 4

4. Buatlah akun di <https://github.com/>

- Buat screenshot sebagai jawaban nomor 4 di Tugas Pertemuan 1



Tugas Pertemuan 1 Jawaban Nomor 5

5. Lakukan praktek dari <https://youtu.be/mSO2hJln0OY?feature=shared>

- Load sample dataset

```
jupyter deni_41155050210017 Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
JupyterLab Python 3 (ipykernel)

[3]: from sklearn.datasets import load_iris

iris = load_iris()
iris

=====
Missing Attribute Values: None
Class Distribution: 33.3% for each of 3 classes.
Creator: R.A. Fisher
Donor: Michael Marsha
ll (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 Du
da & 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 s
eparable from the other 2; the latter are NOT linearly separable from each other.
References:
- Fisher, R.A. "The use of multiple mea
surements 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 0-471-22661-1. See page 21
8.
- Dasarthy, 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 Neares
t Neighbor Rule". IEEE Transactions
on Information Theory, May 1972, 431-433.
- See also: 1988 MLC Proceedings, 54-64. Cheeseman et al's AUTOCLAS
S II
conceptual clustering system finds 3 classes in the data.
- Many, many more ...
'feature_names': ['sepal length (cm)',
'sepal width (cm)',
'petal length (cm)',
'petal width (cm)'],
'filename': 'iris.csv',
'data_module': 'sklearn.datasets.data')

[4]: iris.keys()

[4]: dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module'])
```

- Metadata | Deskripsi dari sample dataset

```
jupyter deni_41155050210017 Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
JupyterLab Python 3 (ipykernel)

[5]: 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  6.0   3.76   1.76    0.9490 (high)
```

- Explanatory & Response Variables | Features & Target

```
jupyter deni_41155050210017 Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
JupyterLab Python 3 (ipykernel)

[7]: X = iris.data
     # X.shape
     X

[7]: 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],
        [5.8, 4. , 1.2, 0.2],
        [5.7, 4.4, 1.5, 0.4],
        [5.4, 3.9, 1.3, 0.4],
        [5.1, 3.5, 1.4, 0.2]])
```

```
[8]: y = iris.target  
      # y.shape  
      y  
  
[8]: 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, 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, 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, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```



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

[9]: ['sepal length (cm)',
      'sepal width (cm)',
      'petal length (cm)',
      'petal width (cm)']

[10]: target_names = iris.target_names
    target_names

[10]: array(['setosa', 'versicolour', 'virginica'], dtype='<U10')
```



```
[13]: 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()
```

●

```
[15]: 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,)
```

- Load sample dataset sebagai Pandas Data Frame

```
[17]: iris = load_iris(as_frame=True)
      iris_features_df = iris.data
      iris_features_df
```

	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 x 4 columns

Tugas Pertemuan 1 Jawaban Nomor 6

6. Lakukan praktek dari <https://youtu.be/tiREcHrtDLo?feature=shared>

- Persiapan dataset | Loading & splitting dataset

```
[6]: from sklearn.datasets import load_iris

      iris = load_iris()

      X = iris.data
      y = iris.target

[12]: 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)
```

- Training model Machine Learning

```
[13]: from sklearn.neighbors import KNeighborsClassifier

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

```
[13]: KNeighborsClassifier
      KNeighborsClassifier(n_neighbors=3)
```

- Evaluasi model Machine Learning

```
[14]: 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
```

- Pemanfaatan trained model machine learning

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

preds = model.predict(data_baru)
preds

[15]: array([1, 2])

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

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

- Deploy model Machine Learning | Dumping dan Loading model Machine Learning

```
[17]: import joblib

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

[17]: ['iris_classifier_knn.joblib']

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

Tugas Pertemuan 1 Jawaban Nomor 7

7. Lakukan praktek dari <https://youtu.be/smNnhEd26Ek?feature=shared>

- Persiapan sample dataset


```
[23]: 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

[23]: 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]])

[25]: sample_data.shape

[25]: (4, 3)
```

- Teknik data preprocessing 1: binarization

```
[26]: sample_data

[26]: 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]])

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

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

- Teknik data preprocessing 2: scaling

```
[28]: sample_data

[28]: 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]])

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

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

[32]: scaled_data = preprocessor.fit_transform(sample_data)
      scaled_data

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

- Teknik data preprocessing 3: normalisation

```
[33]: sample_data

[33]: 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]])

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

[34]: 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]])

[35]: sample_data

[35]: 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]])

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

[36]: 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 ]])
```