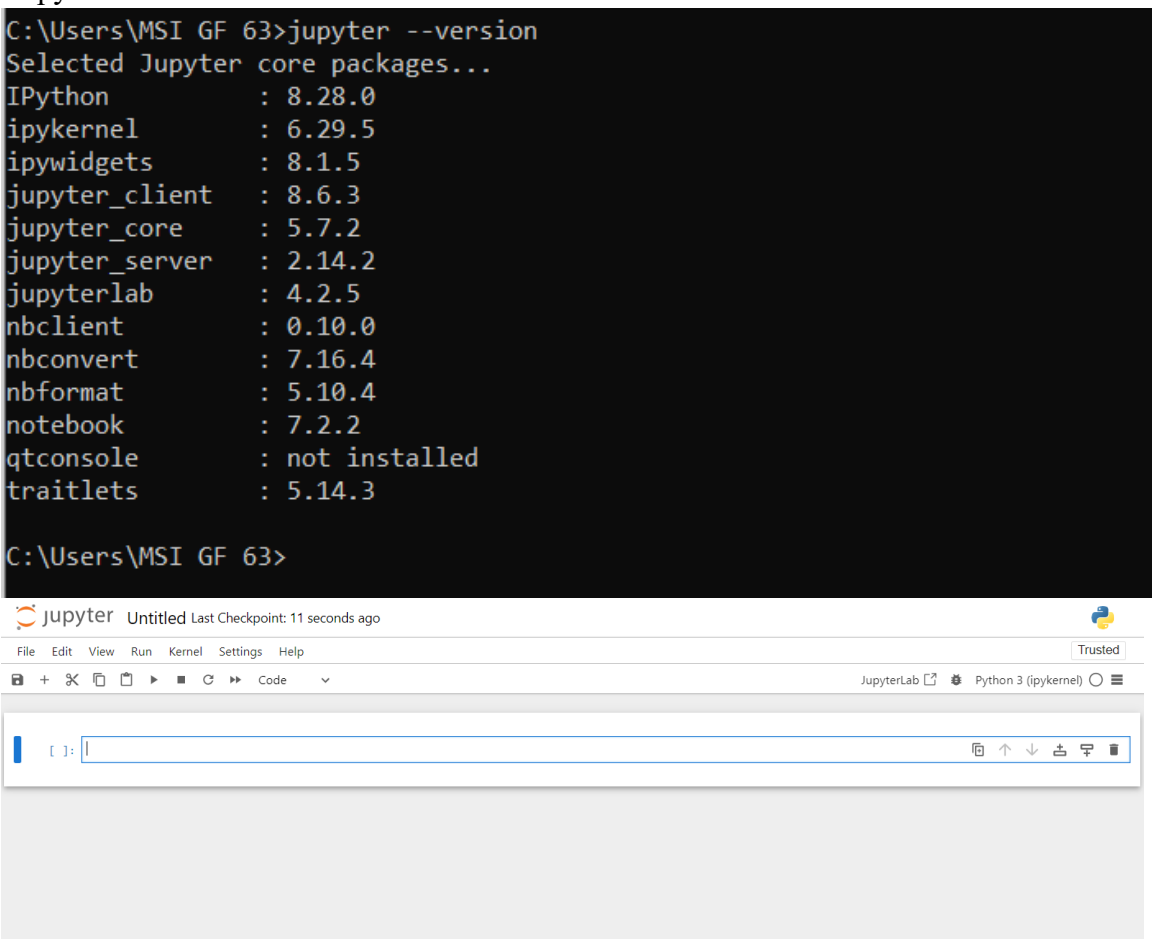
Nama : Mochammad Fajar Maulana

NPM : 41155050210004

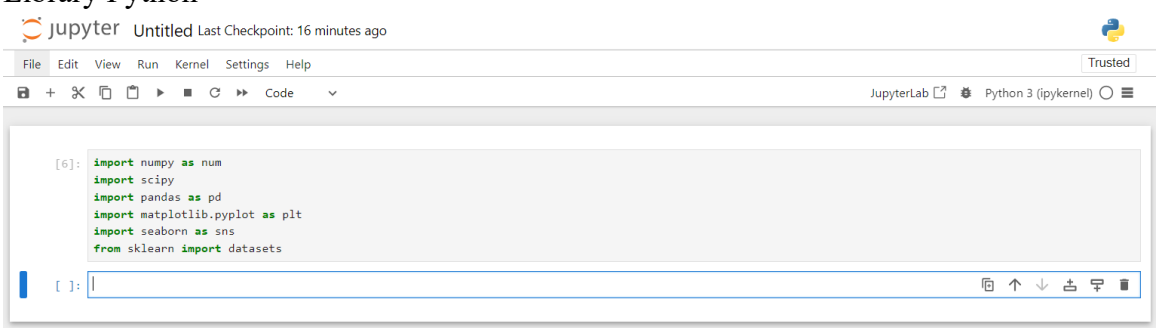
Kelas : INF A1

Tugas Pertemuan 1

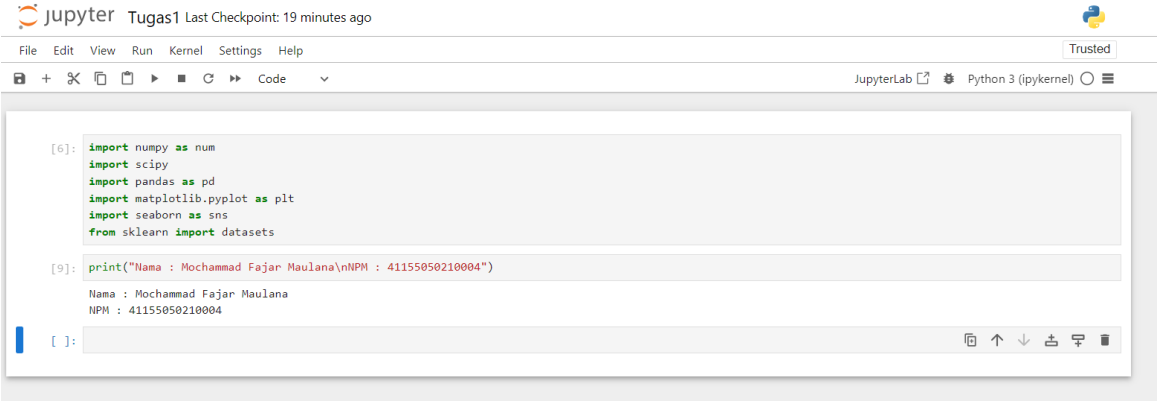
1. Instalasi Jupyter Notebook
    1. Jupyter Notebook



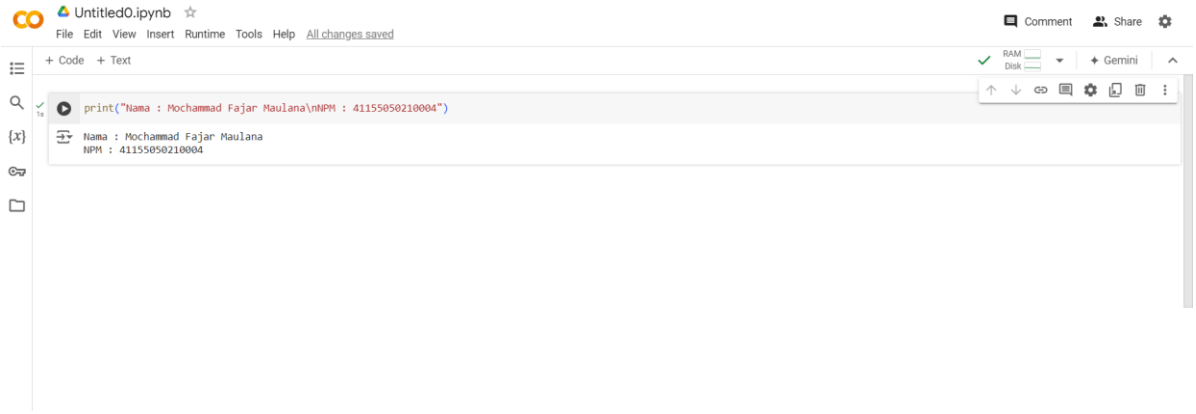
    2. Library Python



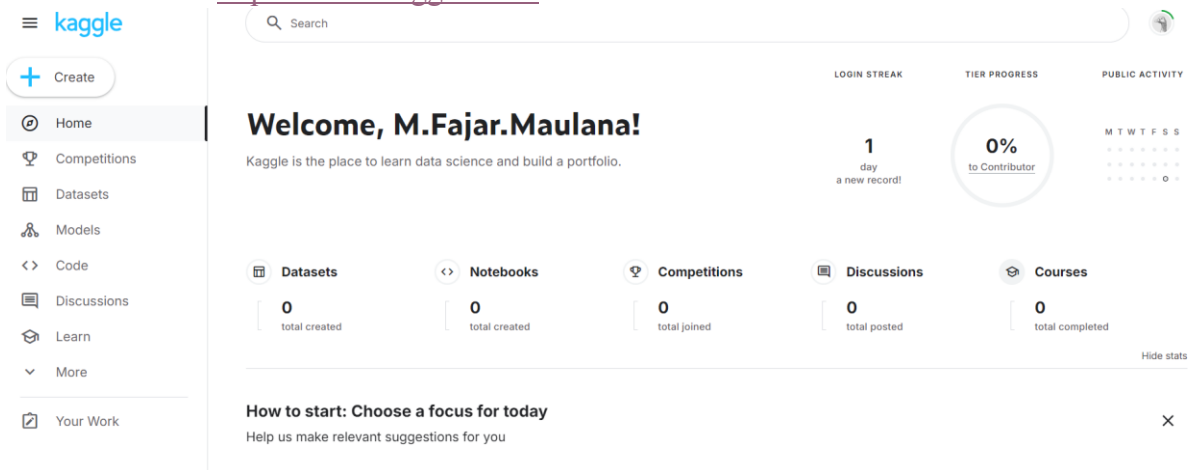
3. Hasil



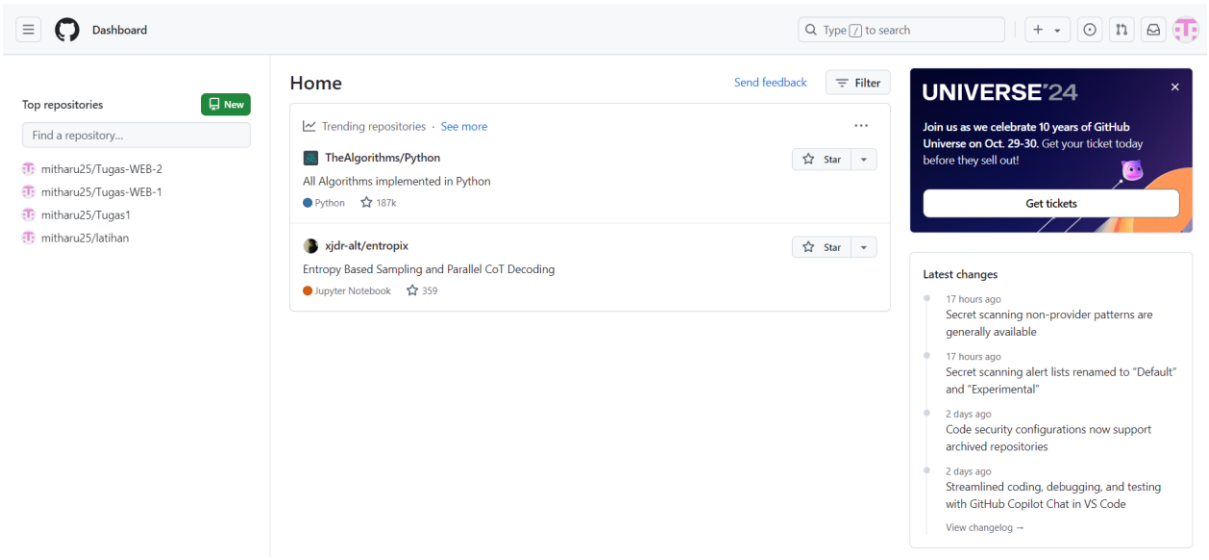
2. Menggunakan Google Collab



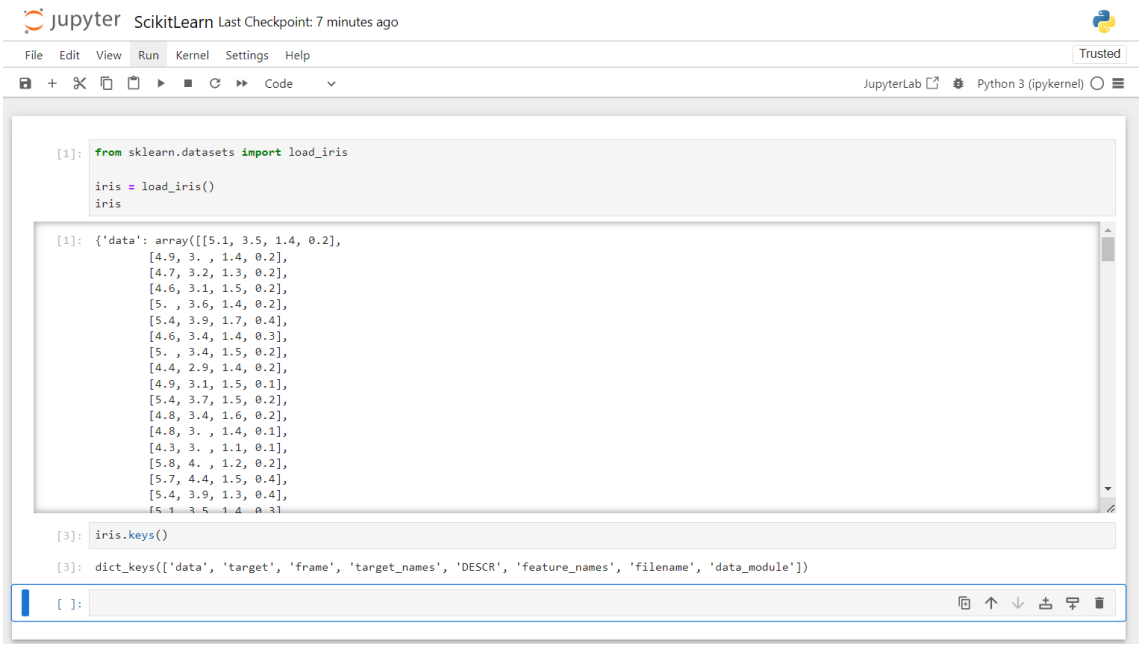
3. Buatlah akun di https://www.kaggle.com/



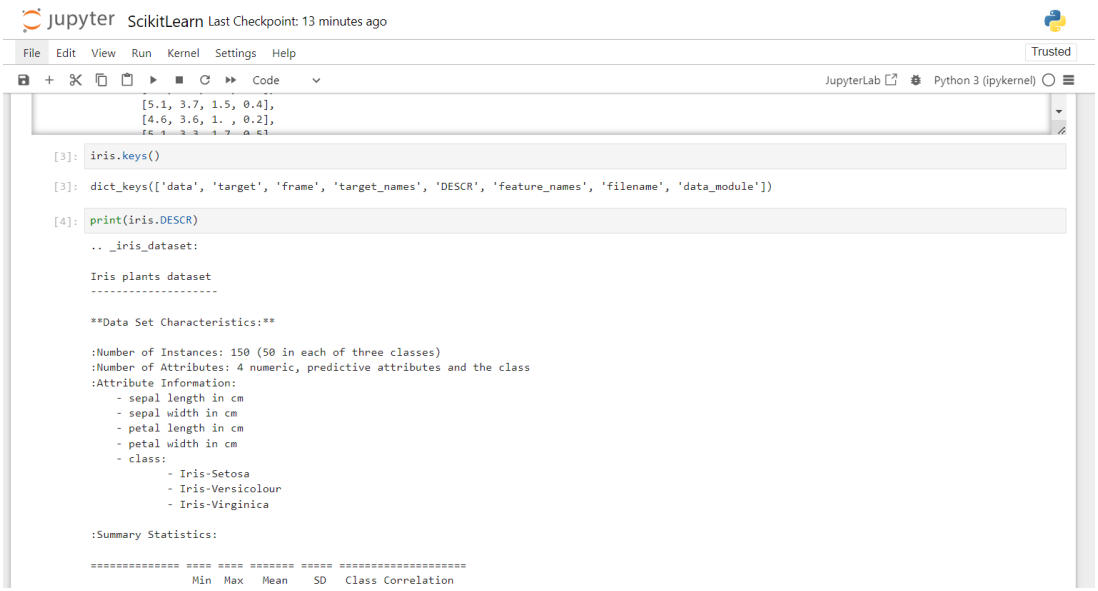
4. Buatlah akun di https://github.com/



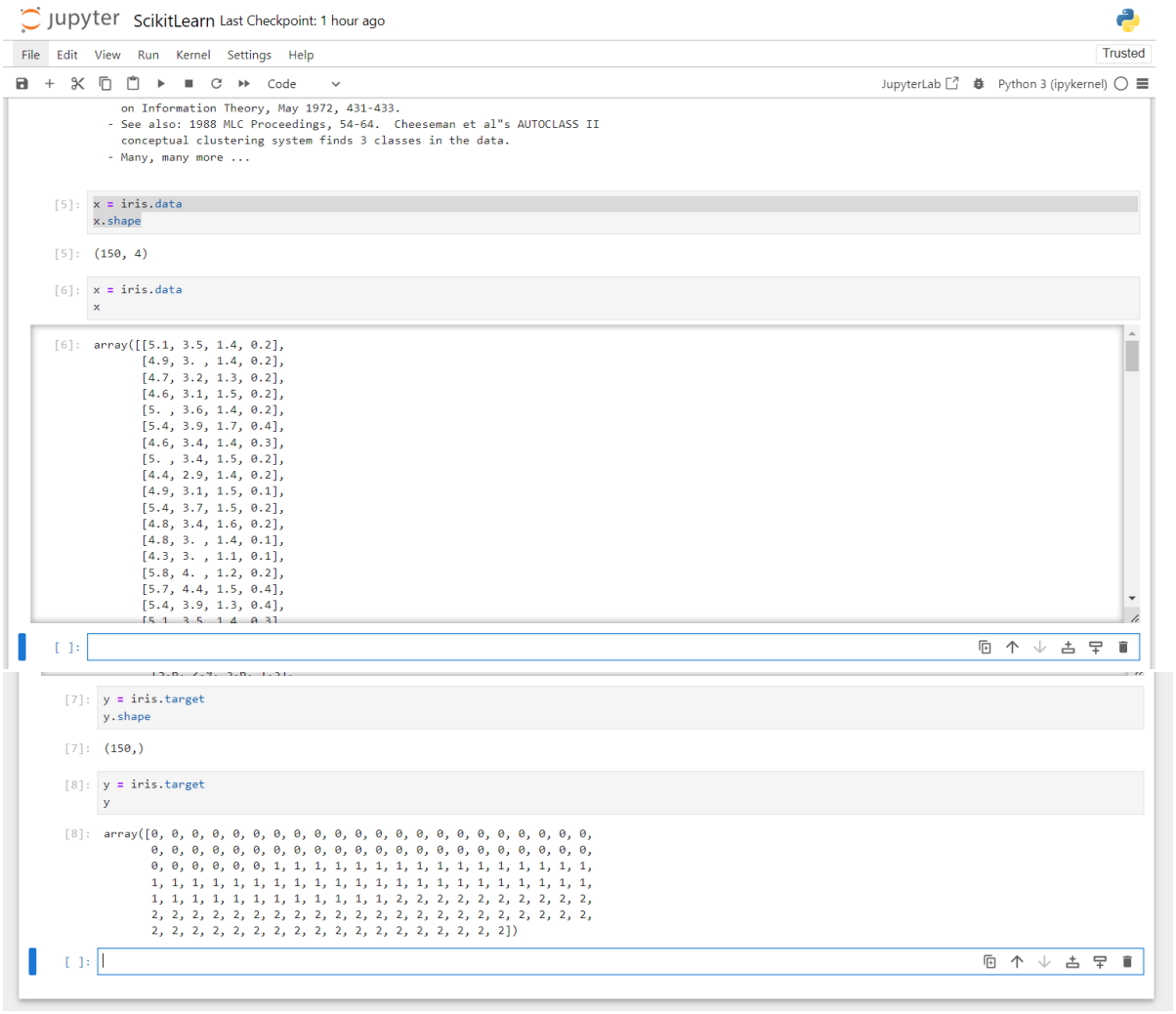
5. Lakukan praktek dari https://youtu.be/mSO2hJln0OY?feature=shared
   1. Load sample dataset



   2. Metadata | Deskripsi dari sample dataset



3. Explanatory & Response Variables | Features & Target



4. Feature & Target Names

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

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

[10]: feature_names = iris.target_names
      feature_names

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

[ ]:
```

5. Visualisasi Data

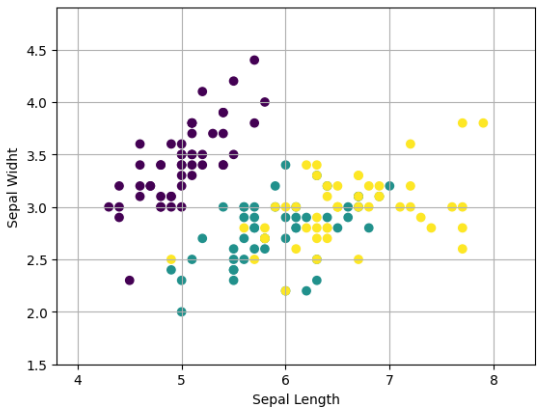
```
[24]: import matplotlib.pyplot as plt

      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 Widht')

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




```
[ ]:
```

6. Training Set & Testing Set

```
[30]: 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,)

[ ]:
```

7. Load sample dataset sebagai Pandas Data Frame

```
[31]: iris = load_iris(as_frame=True)

      iris_feature_df = iris.data
      iris_feature_df
```

[31]:

	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. Lakukan praktek dari https://youtu.be/tiREcHrtDLo?feature=shared
   1. Persiapan dataset | Loading & splitting dataset

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

     iris = load_iris()

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

```
[7]: 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)
```

```
[ ]:
```

   2. Training model Machine Learning

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

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

[8]:

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

```
[ ]:
```

   3. Evaluasi model Machine Learning

```
[9]: 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
```

```
[ ]:
```

   4. Pemanfaatan trained model machine learning

```
[10]: data_baru = [[4, 6, 2, 2],
                   [2, 3, 3, 5]]

      preds = model.predict(data_baru)
      preds

[10]: array([0, 1])

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

      Hasil Prediksi: ['setosa', 'versicolor']

[ ]:
```

5. Deploy model Machine Learning | Dumping dan Loading model Machine Learning

```
[20]: import joblib

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

[20]: ['iris_classifier_knn.joblib']

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

[ ]:
```

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

[ ]:
```

   2. Teknik data preprocessing 1: binarization

```
[3]: preprocessor = preprocessing.Binarizer(threshold=0.5)
     binarised_data = preprocessor.transform(sample_data)
     binarised_data

[3]: array([[1., 0., 1.],
            [0., 1., 1.],
            [0., 0., 1.],
            [1., 1., 0.]])

[ ]:
```

   3. Teknik data preprocessing 2: scaling

```
[4]: sample_data

[4]: 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]])

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

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

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

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

[ ]:
```

4. Teknik data preprocessing 3: normalisation

```
[7]: sample_data

[7]: 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]])

[8]: li_normalised_data = preprocessing.normalize(sample_data, norm='l1')
     li_normalised_data

[8]: 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]])

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

[9]: 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 ]])

[ ]:
```