

Laporan Tugas Machine Learning

Dosen : Estiyan Dwepriyoko, S.kom., M.T



Disusun Oleh :

Rio Putra Anugrah

41155050210044

TIF – A2

PROGRAM STUDI INFORMATIKA

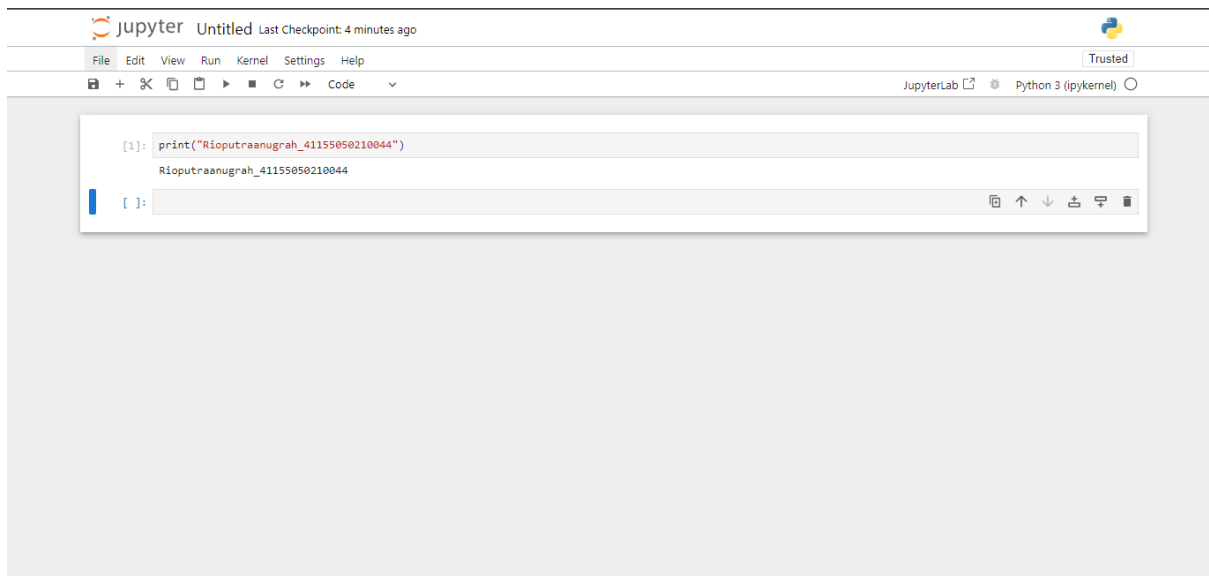
FAKULTAS TEKNIK

UNIVERSITAS LANGLANGBUANA

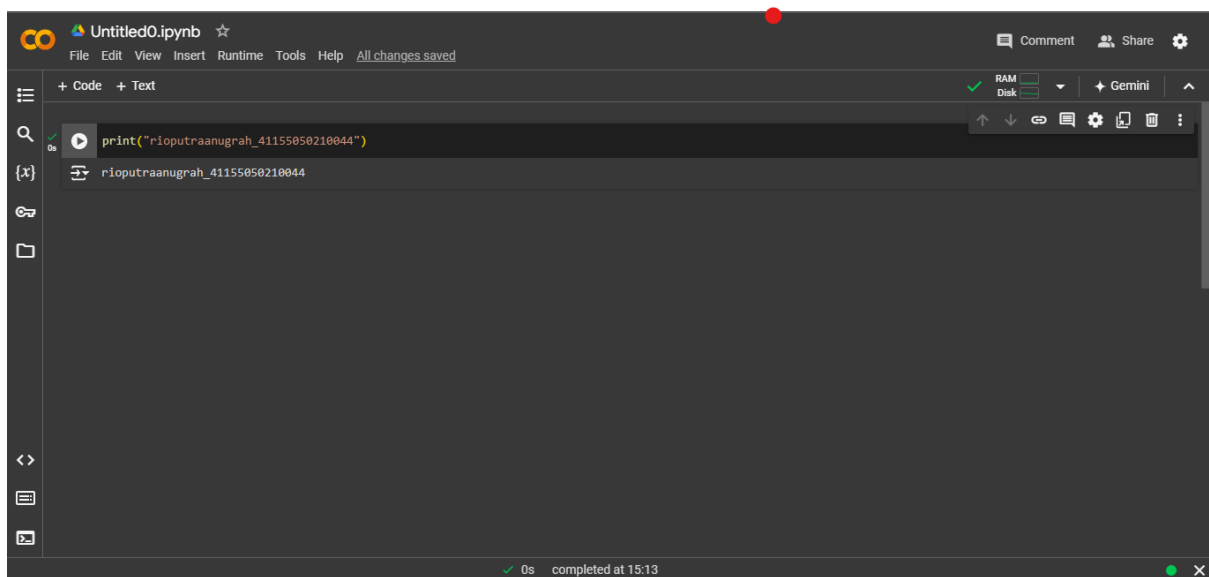
BANDUNG

2023/2024

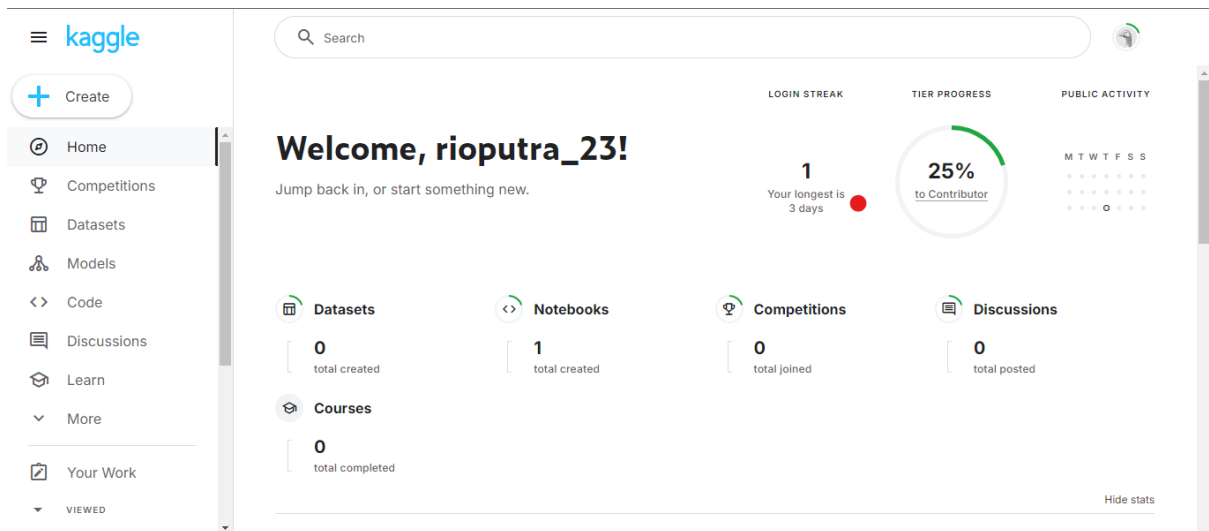
1. Istalasi jupyter notebook



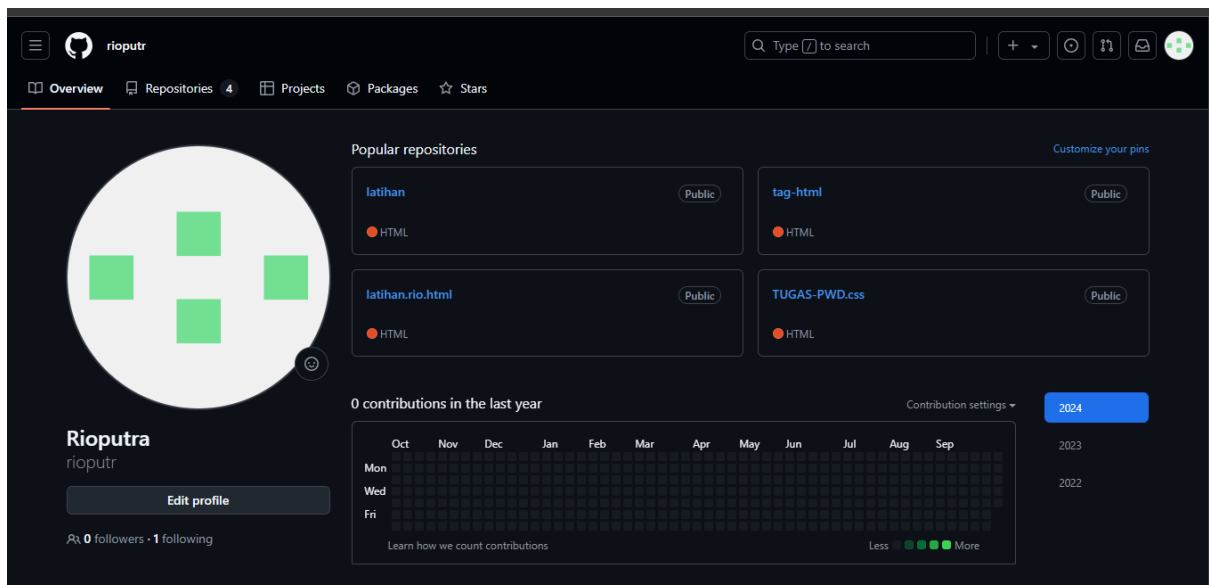
2. Google colab



3. Pembuatan akun kaggle

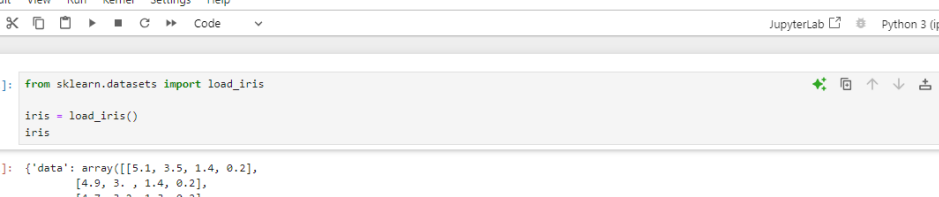


4. Pembuatan akun github



5. Melakukan praktek

- Load sample dataset
- Metadata
- Explanatoru dan response variables
- Features dan target names
- Visualisasi data
- Training set dan testing set
- Load sample dataset



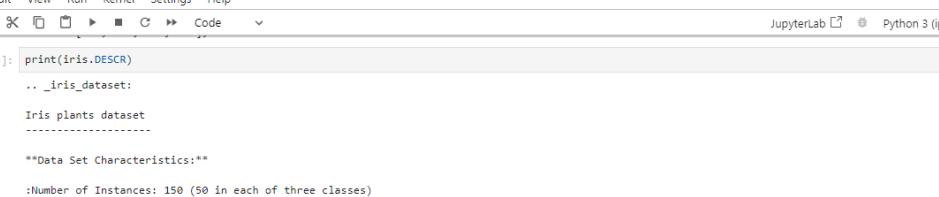
The screenshot shows a Jupyter Notebook window titled "Untitled1" with a "Last Checkpoint: 2 minutes ago" status. The interface includes a top menu bar with options: File, Edit, View, Run, Kernel, Settings, and Help. On the right, there is a "Trusted" badge and a "Python 3 (ipykernel)" label. Below the menu bar is a toolbar with icons for file operations, execution, and code editing. The main area contains a code cell with the following Python code:

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

iris = load_iris()
iris
```

The output of the code cell is displayed below the code, showing the loaded Iris dataset as a dictionary:

```
[1]: {'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],
[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, 0.3]]),
'feature_names': ['sepal_length', 'sepal_width', 'petal_length', 'petal_width'],
'target_names': ['species'],
'target': array(['setosa', 'versicolour', 'versicolour'], dtype=object),
'filename': 'iris.csv'}
```



The screenshot shows a JupyterLab window with a code editor and a console. The code cell contains a print statement for the Iris dataset. The output in the console shows the following information:

```
[3]: 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!)
=====
```

[]:

[]:

[]:

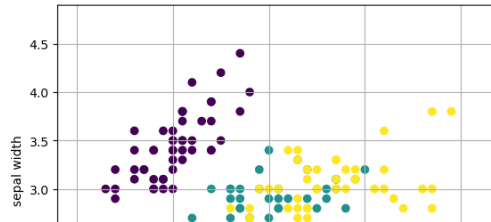
```
[70]: import matplotlib.pyplot as plt

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()
```



```
[74]: 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,)
```

x test: (45,)

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

iris_features_df = iris.data
iris_features_df
```

```
[76]:
```

	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. Melakukan praktek

- Persiapan dataset
- Training model machine learning
- Evaluasi model machine learning
- Pemanfaatan trained model machine learning
- Deploy model machine learning

[94]:

```
from sklearn.datasets import load_iris  
  
iris = load_iris()  
  
x = iris.data  
y = iris.target
```

[92]:

```
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)
```

[90]:

```
from sklearn.neighbors import KNeighborsClassifier  
  
model = KNeighborsClassifier(n_neighbors=3)  
model.fit(x_train,y_train)
```

[90]:

```
▼ KNeighborsClassifier ⓘ ?  
KNeighborsClassifier(n_neighbors=3)
```

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

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

```
preds
```

```
[130]:
```

```
pred_species = [iris.target_names[p] for p in preds]
```

```
print(f'Hasil Prediksi: {pred_species}')
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[130], line 1
----> 1 pred_species = [iris.target_names[p] for p in preds]
      2 print(f'Hasil Prediksi: {pred_species}')
```

NameError: name 'preds' is not defined

```
[ ]:
```

```
[132]:
```

```
import joblib
```

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

```
[132]:
```

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

```
[138]:
```

```
production_model = joblib.load('iris_classifier_knn.joblib')
```

```
[ ]:
```


7. Melakukan praktek

- Persiapan sample dataset
- Teknik data preprocessing 1. Binarization
- Teknik data preprocessing 2. Scaling
- Teknik data preprocessing 3. Normalisation

[11]:

```
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
```

[11]:

```
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.shape
```

[18]:

```
sample_data.shape
```

[18]:

(4, 3)

[24]:

```
sample_data
```

[24]:

```
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]])
```

[66]:

```
preprocessor = preprocessing.binarizer(threshold=0.5)
binarised_data = preprocessor.transform(sample_data)
binarised_data
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[66], line 1
----> 1 preprocessor = preprocessing.binarizer(threshold=0.5)
      2 binarised_data = preprocessor.transform(sample_data)
      3 binarised_data

AttributeError: module 'sklearn.preprocessing' has no attribute 'binarizer'
```

[88]:

```
sample_data
```

[88]:

```
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]])
```

[128]:

```
preprocessor = preprocessing.MinMaxScaler(feature_range=(0, 1))
preprocessor.fit(sample_data)
scaled_data = preprocessor.fit_transform(sample_data)
scaled_data
```

```
-----
NameError                                    Traceback (most recent call last)
Cell In[128], line 1
----> 1 preprocessor = preprocessing.MinMaxScaler(feature_range=(0, 1))
      2 preprocessor.fit(sample_data)
      3 scaled_data = preprocessor.fit_transform(sample_data)

NameError: name 'preprocessing' is not defined
```

[106]:

```
scaled_data = preprocessor.fit_transform(sample_data)
scaled_data
```

```
-----
NameError                                    Traceback (most recent call last)
Cell In[106], line 1
----> 1 scaled_data = preprocessor.fit_transform(sample_data)
```

```
[136]: sample_data
```

```
[136]: 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]])
```

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

```
[138]: 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]])
```

```
[ ]:
```

```
[144]:
```

```
sample_data
```

```
[144]:
```

```
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]])
```

```
[146]:
```

```
l2_normalised_data = preprocessing.normalize(sample_data, norm='l2')
l2_normalised_data
```

```
[146]:
```

```
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 ]])
```

```
[ ]:
```