MACHINE LEARNING

(Tugas 2)

Disusun Oleh:

Nama: Sifa Maryam Rahman

NPM: 41155050210031

Kelas: TIF A1

TEKNIK INFORMATIKA

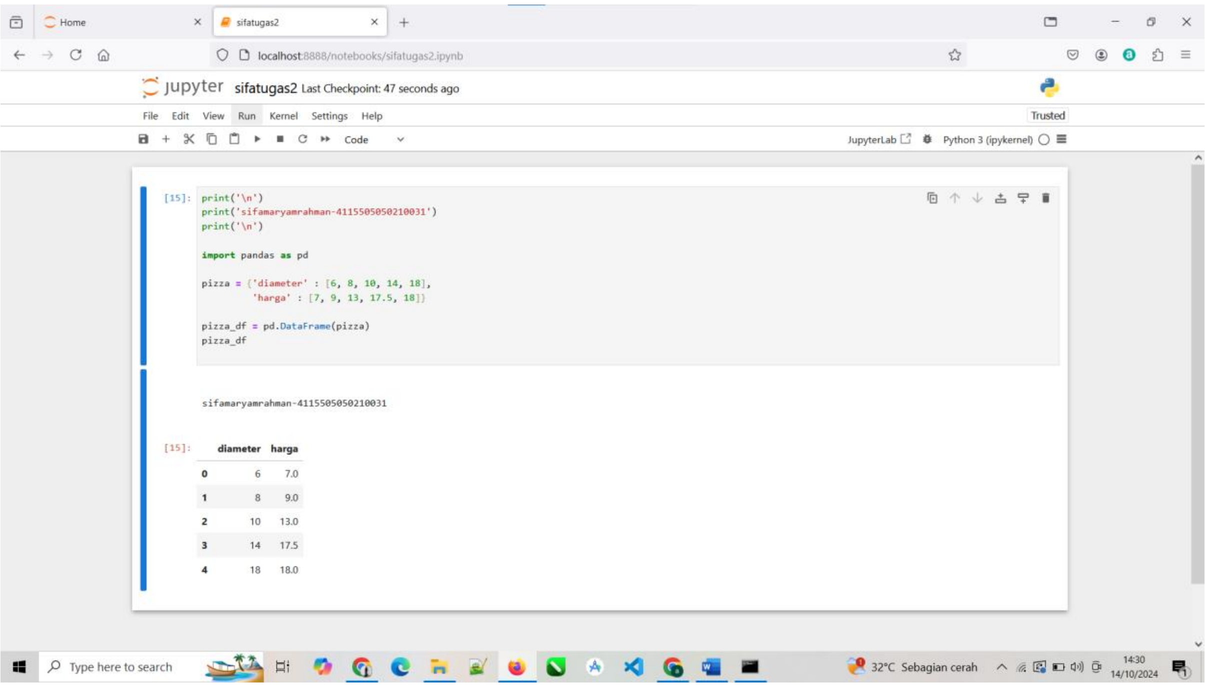
FAKULTAS TEKNIK

UNIVERSITAS LANGLANGBUANA

2024

1. Berikut adalah hasil praktik dari video youtube https://youtu.be/lcjq7-2zMSA?si=f4jWJR6lY8y0BZKl

1.1. Sample dataset


Home
sifatugas2
localhost:8888/notebooks/sifatugas2.ipynb
jupyter sifatugas2 Last Checkpoint: 47 seconds ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
32°C Sebagian cerah
14:30
14/10/2024

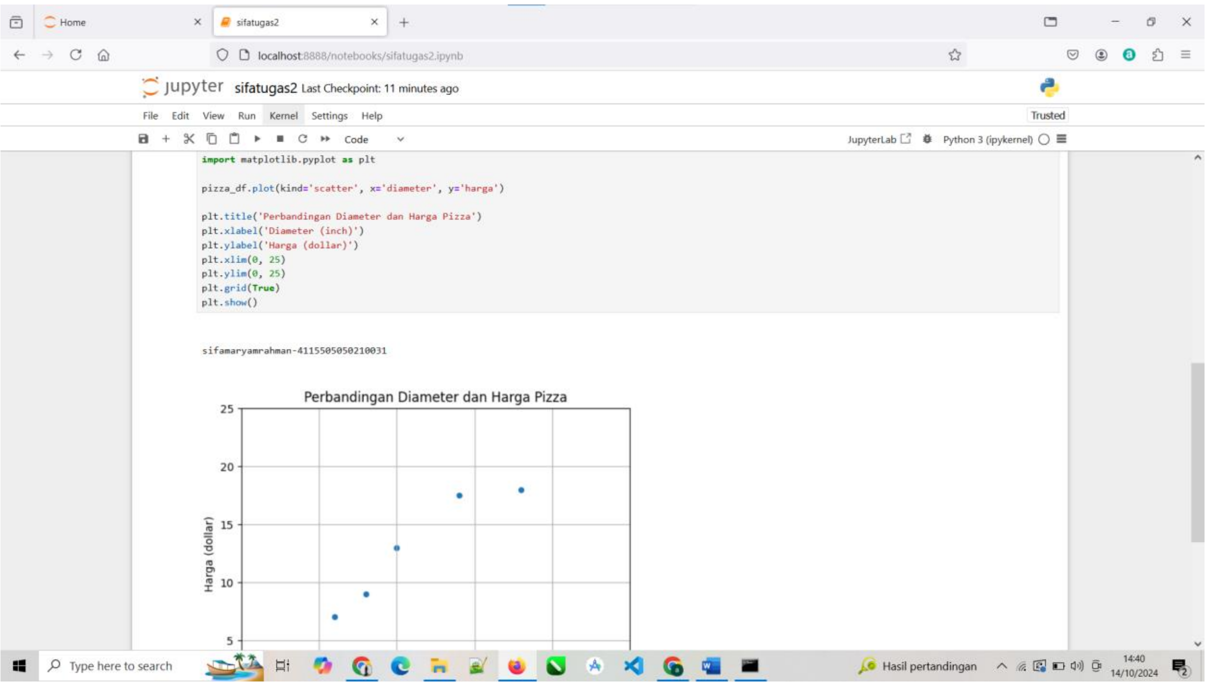
```
[15]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      import pandas as pd

      pizza = {'diameter' : [6, 8, 10, 14, 18],
               'harga' : [7, 9, 13, 17.5, 18]}

      pizza_df = pd.DataFrame(pizza)
      pizza_df
```

```
sifamaryamrahman-4115505050210031
```

[15]:

	diameter	harga
0	6	7.0
1	8	9.0
2	10	13.0
3	14	17.5
4	18	18.0

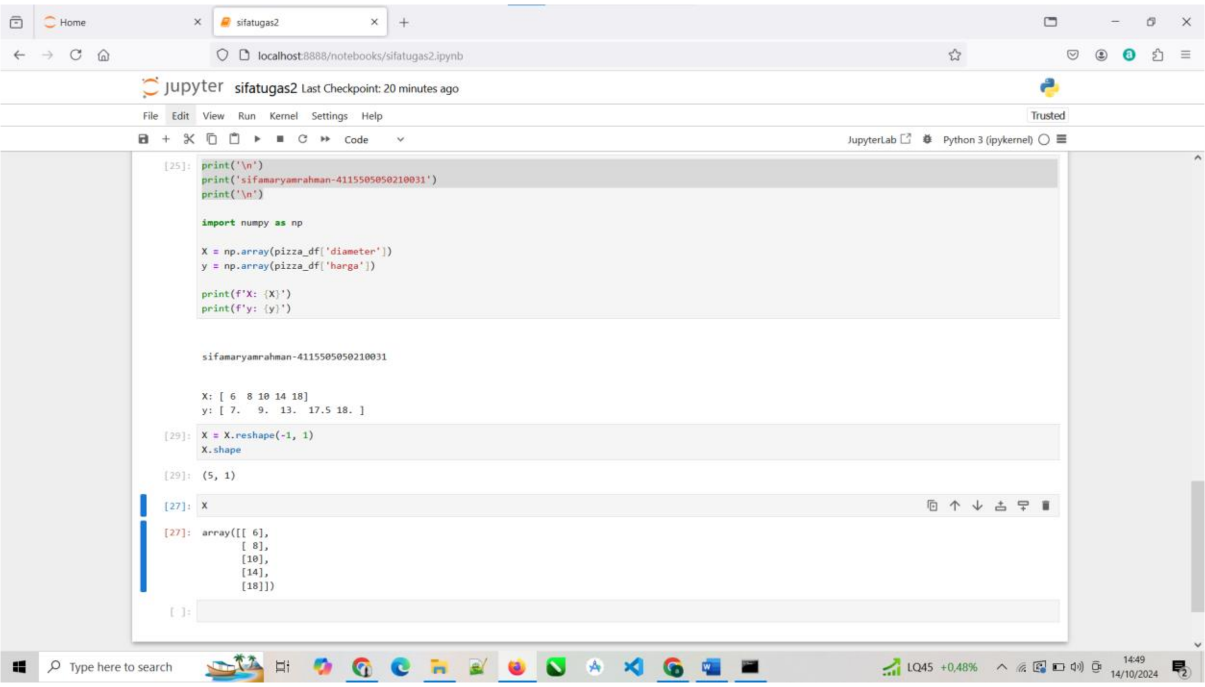
1.2.Visualisasi dataset


Home
sifatugas2
localhost:8888/notebooks/sifatugas2.ipynb
jupyter sifatugas2 Last Checkpoint: 11 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Perbandingan Diameter dan Harga Pizza
25
20
15
10
5
Harga (dollar)
Type here to search
Hasil pertandingan
14:40
14/10/2024

```
import matplotlib.pyplot as plt

pizza_df.plot(kind='scatter', x='diameter', y='harga')

plt.title('Perbandingan Diameter dan Harga Pizza')
plt.xlabel('Diameter (inch)')
plt.ylabel('Harga (dollar)')
plt.xlim(0, 25)
plt.ylim(0, 25)
plt.grid(True)
plt.show()
```

```
sifamaryamrahman-4115505050210031
```

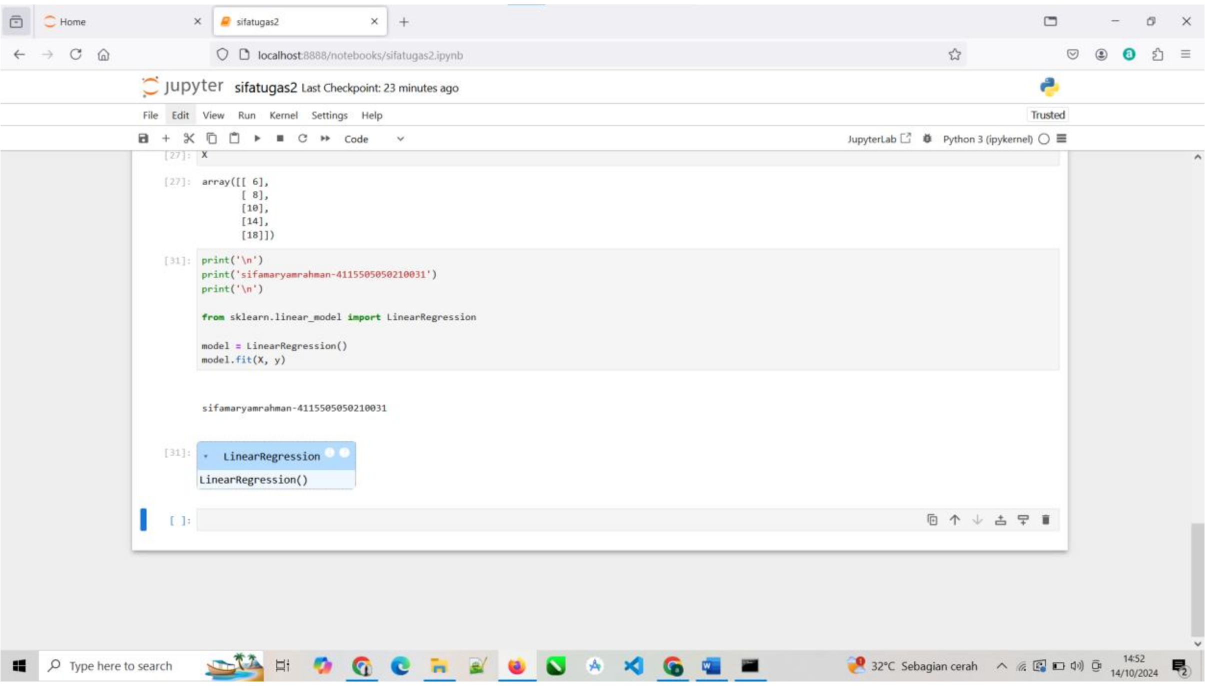
1.3.Transformasi dataset



Home
sifatugas2
localhost:8888/notebooks/sifatugas2.ipynb
jupyter sifatugas2 Last Checkpoint: 20 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
LQ45 +0,48%
14:49
14/10/2024

```
[25]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      import numpy as np

      X = np.array(pizza_df['diameter'])
      y = np.array(pizza_df['harga'])

      print(f'X: {X}')
      print(f'y: {y}')


      sifamaryamrahman-4115505050210031


      X: [ 6  8 10 14 18]
      y: [ 7.   9.  13.  17.5 18. ]

[29]: X = X.reshape(-1, 1)
      X.shape

[29]: (5, 1)

[27]: X

[27]: array([[ 6],
             [ 8],
             [10],
             [14],
             [18]])

[ ]:
```

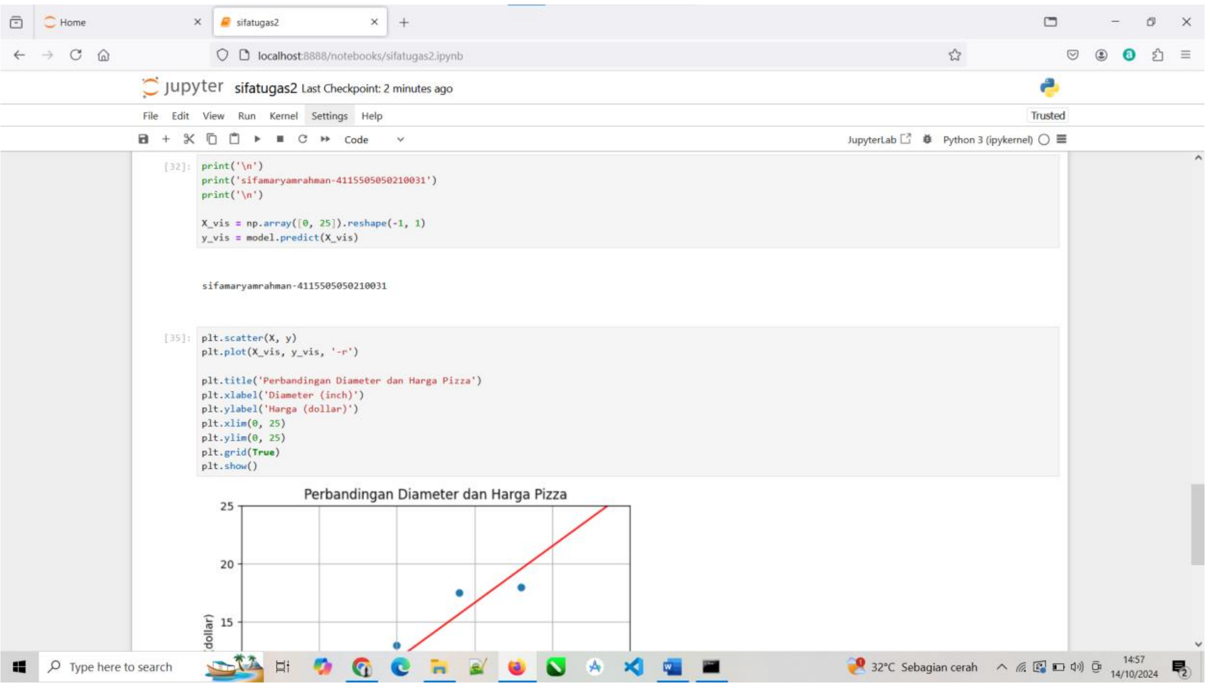
1.4.Training Simple Linear Regression Model



Home
sifatugas2
localhost:8888/notebooks/sifatugas2.ipynb
jupyter sifatugas2 Last Checkpoint: 23 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
32°C Sebagian cerah
14:52
14/10/2024

```
[27]: X

[27]: array([[ 6],
             [ 8],
             [10],
             [14],
             [18]])

[31]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.linear_model import LinearRegression

      model = LinearRegression()
      model.fit(X, y)


      sifamaryamrahman-4115505050210031


[31]: ▾ LinearRegression
      LinearRegression()

[ ]:
```

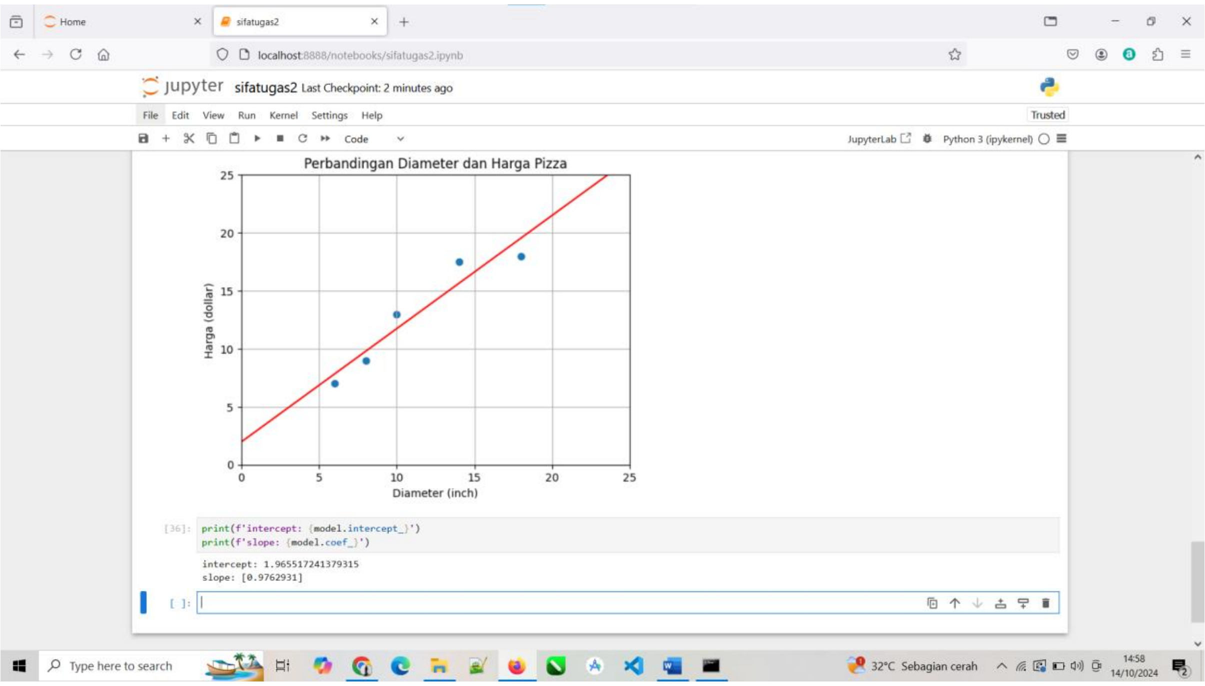
1.5.Visualisasi Simple Linear Regression Model | Penjelasan persamaan garis linear


Home
sifatugas2
localhost:8888/notebooks/sifatugas2.ipynb
jupyter sifatugas2 Last Checkpoint: 2 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Perbandingan Diameter dan Harga Pizza
25
20
15
dollar)
Type here to search
32°C Sebagian cerah
14:57
14/10/2024

```
[32]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      X_vis = np.array([0, 25]).reshape(-1, 1)
      y_vis = model.predict(X_vis)
```

```


sifamaryamrahman-4115505050210031


```

```
[35]: plt.scatter(X, y)
      plt.plot(X_vis, y_vis, '-r')

      plt.title('Perbandingan Diameter dan Harga Pizza')
      plt.xlabel('Diameter (inch)')
      plt.ylabel('Harga (dollar)')
      plt.xlim(0, 25)
      plt.ylim(0, 25)
      plt.grid(True)
      plt.show()
```

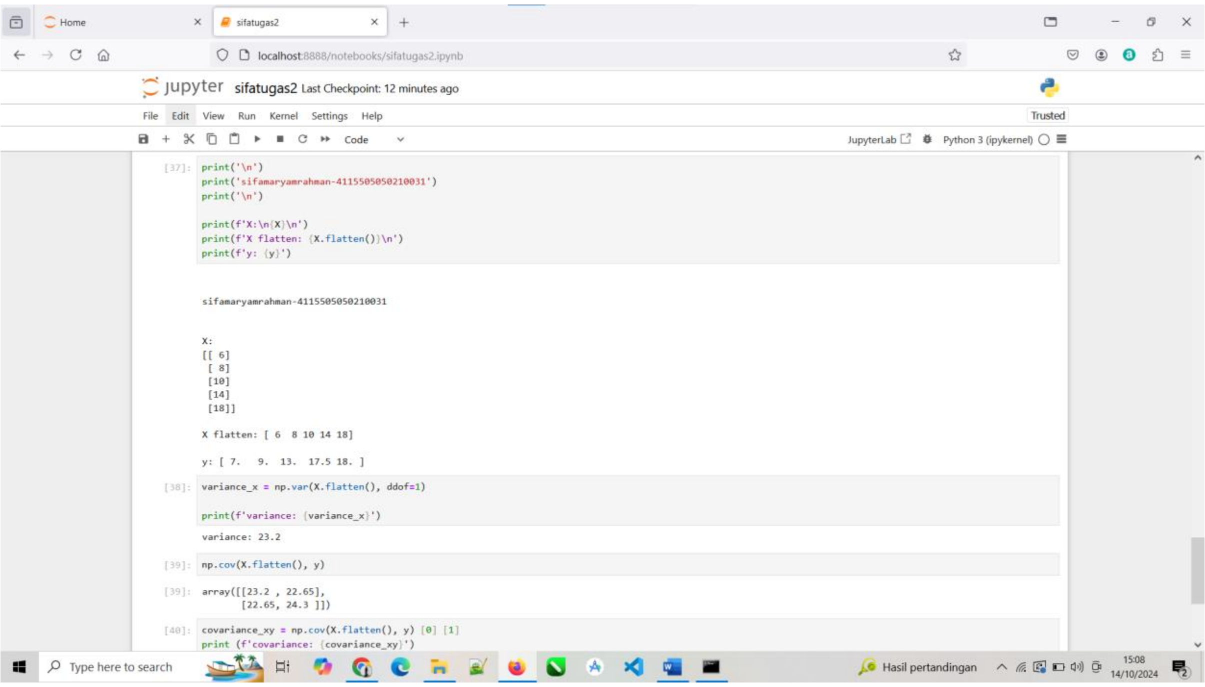

Home
sifatugas2
localhost:8888/notebooks/sifatugas2.ipynb
jupyter sifatugas2 Last Checkpoint: 2 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Perbandingan Diameter dan Harga Pizza
25
20
15
10
5
0
Harga (dollar)
0
5
10
15
20
25
Diameter (inch)
Type here to search
32°C Sebagian cerah
14:58
14/10/2024

```
[36]: print(f'intercept: {model.intercept_}')
      print(f'slope: {model.coef_}')
```

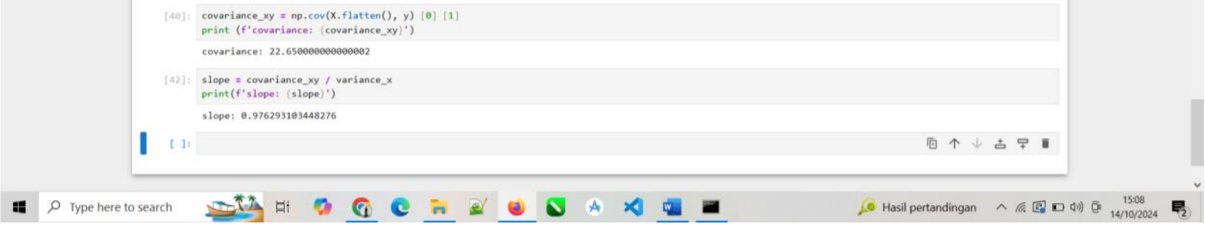
```
intercept: 1.965517241379315
slope: [0.9762931]
```

```
[ ]:
```

1.6.Kalkulasi nilai slope

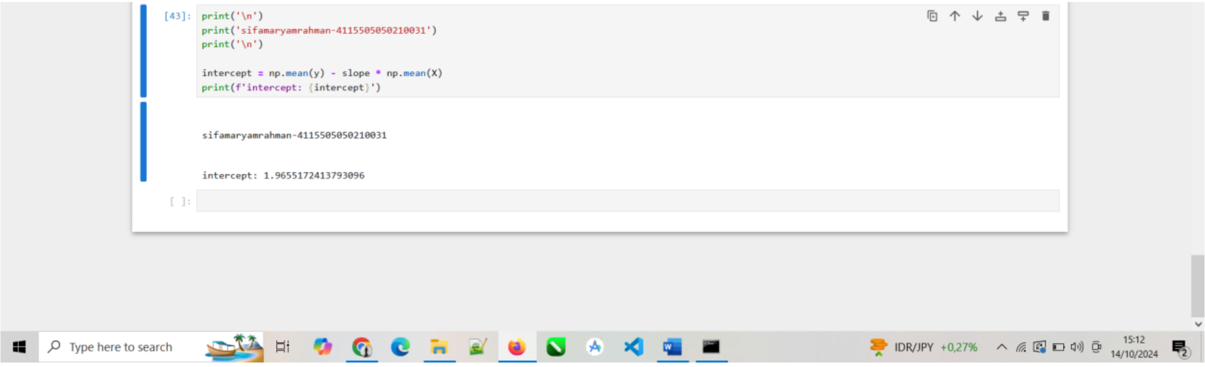

Home
sifatugas2
localhost:8888/notebooks/sifatugas2.ipynb
jupyter sifatugas2 Last Checkpoint: 12 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
Hasil pertandingan
15:08
14/10/2024

```
[37]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      print(f'X:\n{X}\n')
      print(f'X flatten: {X.flatten()}\n')
      print(f'y: {y}')


      sifamaryamrahman-4115505050210031


      X:
      [[ 6]
       [ 8]
       [10]
       [14]
       [18]]

      X flatten: [ 6  8 10 14 18]

      y: [ 7.   9.  13.  17.5 18. ]
[38]: variance_x = np.var(X.flatten(), ddof=1)

      print(f'variance: {variance_x}')
      variance: 23.2
[39]: np.cov(X.flatten(), y)
[39]: array([[23.2 , 22.65],
             [22.65, 24.3 ]])
[40]: covariance_xy = np.cov(X.flatten(), y) [0] [1]
      print (f'covariance: {covariance_xy}')
```



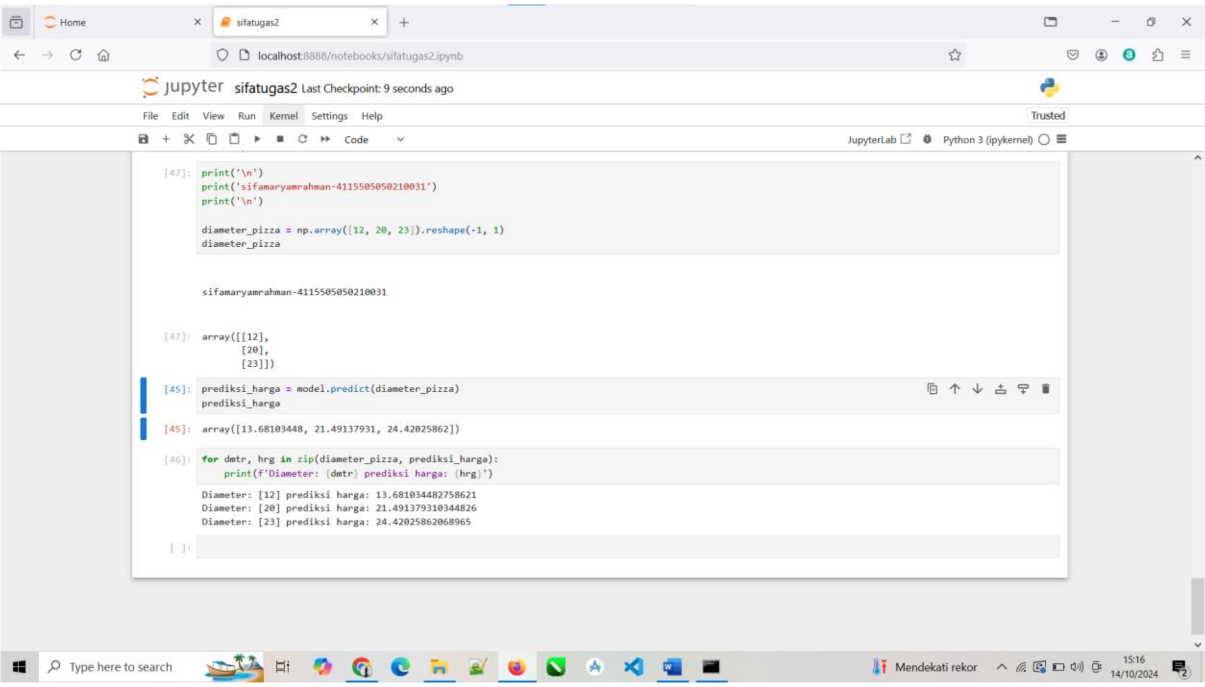
```
[40]: covariance_xy = np.cov(X.flatten(), y) [0] [1]
      print (f'covariance: {covariance_xy}')
      covariance: 22.650000000000002
[42]: slope = covariance_xy / variance_x
      print(f'slope: {slope}')
      slope: 0.976293103448276
[ ]:
```

1.7.Kalkukasi nilai intercept



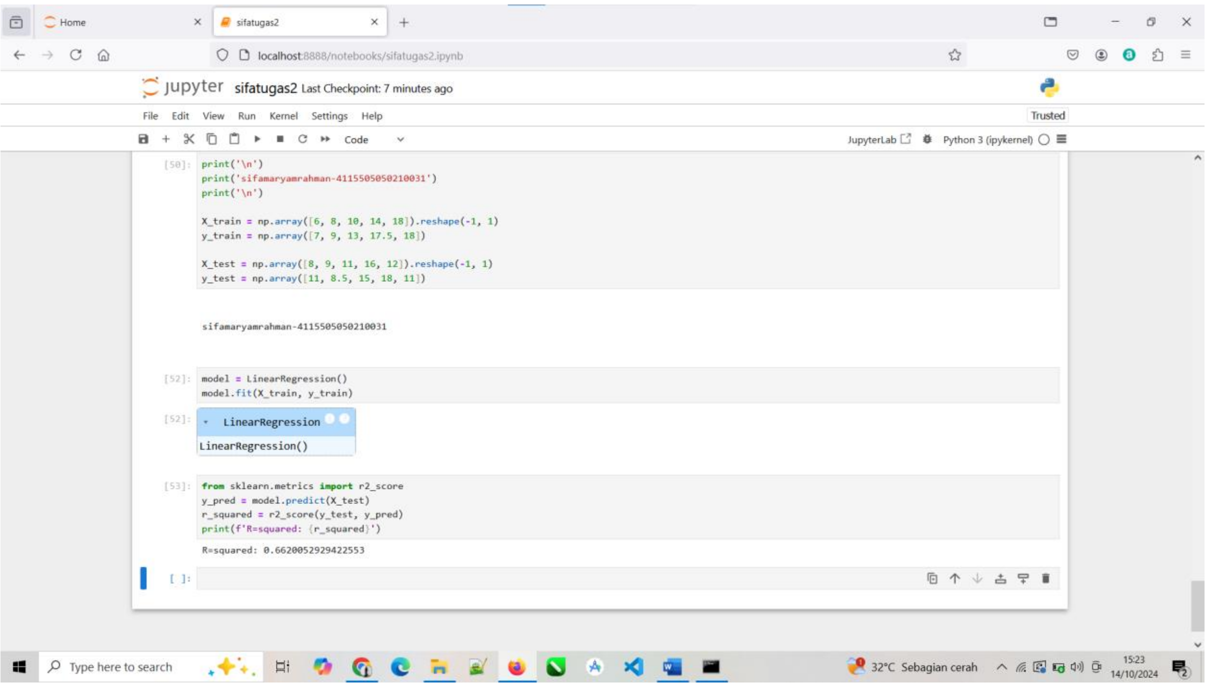
```
[43]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      intercept = np.mean(y) - slope * np.mean(X)
      print(f'intercept: {intercept}')


      sifamaryamrahman-4115505050210031


      intercept: 1.9655172413793096

[ ]:
```

1.8.Prediksi harga pizza dengan Simple Linear Regression Model



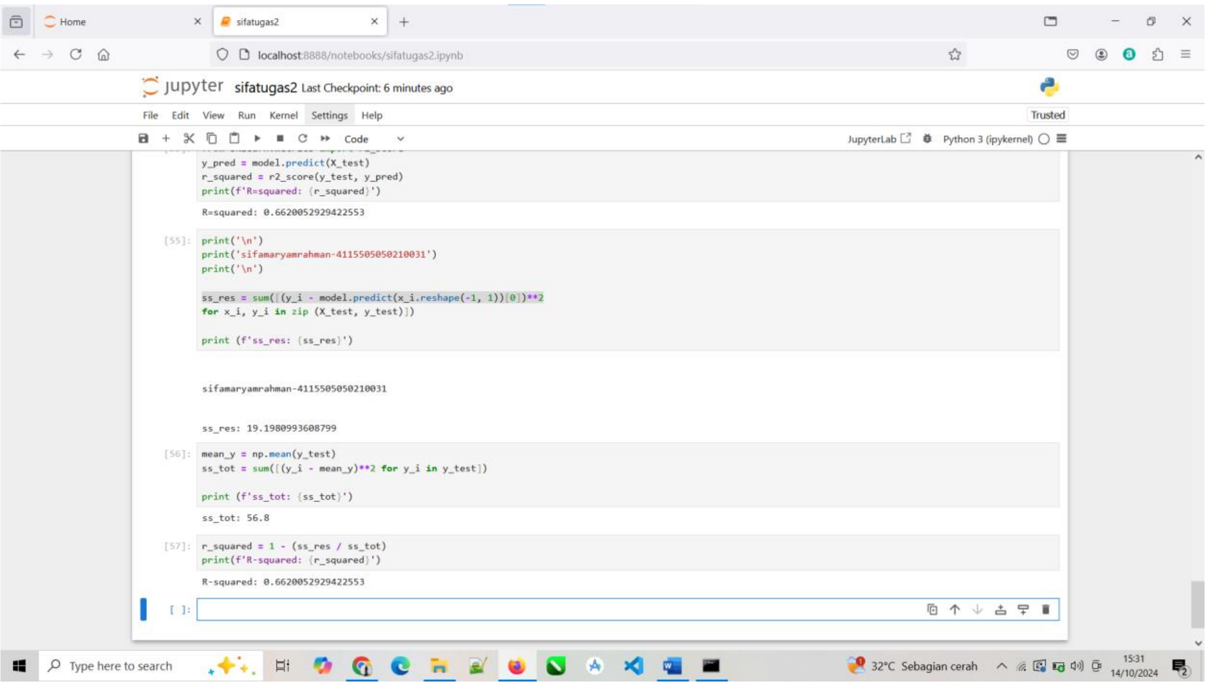
```
[47]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      diameter_pizza = np.array([12, 20, 23]).reshape(-1, 1)
      diameter_pizza


      sifamaryamrahman-4115505050210031


[47]: array([[12],
             [20],
             [23]])

[45]: prediksi_harga = model.predict(diameter_pizza)
      prediksi_harga

[45]: array([13.68103448, 21.49137931, 24.42025862])

[46]: for dmtr, hrg in zip(diameter_pizza, prediksi_harga):
          print(f'Diameter: {dmtr} prediksi harga: {hrg}')

      Diameter: [12] prediksi harga: 13.681034482758621
      Diameter: [20] prediksi harga: 21.491379310344826
      Diameter: [23] prediksi harga: 24.42025862068965

[ ]:
```

1.9. Evaluasi model dengan Coefficient of Determination | R Squared



```
[50]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      X_train = np.array([6, 8, 10, 14, 18]).reshape(-1, 1)
      y_train = np.array([7, 9, 13, 17.5, 18])

      X_test = np.array([8, 9, 11, 16, 12]).reshape(-1, 1)
      y_test = np.array([11, 8.5, 15, 18, 11])


      sifamaryamrahman-4115505050210031


[52]: model = LinearRegression()
      model.fit(X_train, y_train)

[52]: LinearRegression
      LinearRegression()

[53]: from sklearn.metrics import r2_score
      y_pred = model.predict(X_test)
      r_squared = r2_score(y_test, y_pred)
      print(f'R=squared: {r_squared}')

      R=squared: 0.6620052929422553

[ ]:
```

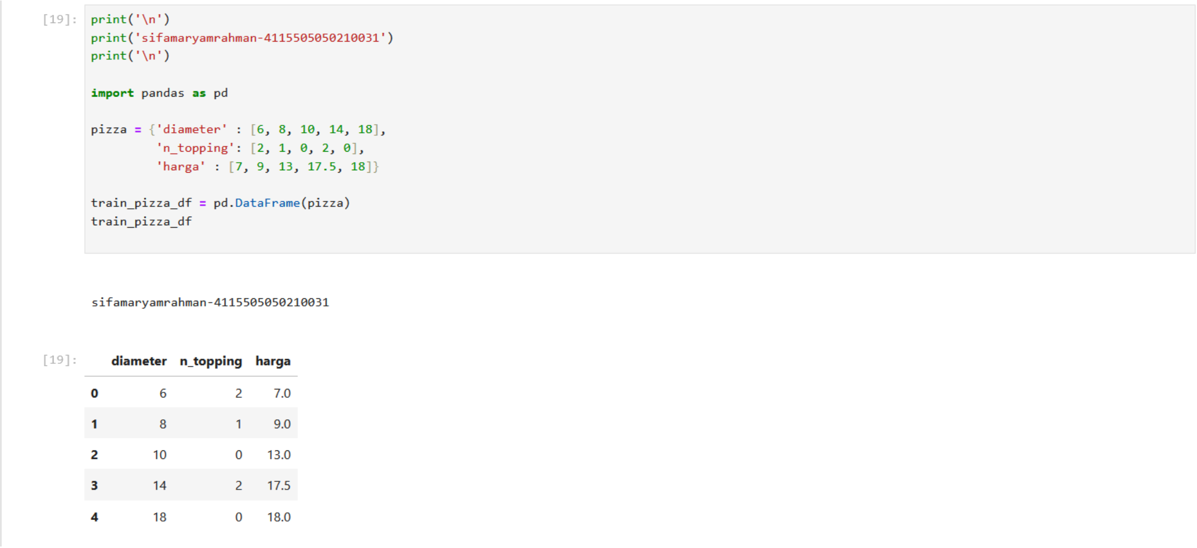
1.10. Kalkulasi nilai R Squared | Coefficient of Determination



```
      y_pred = model.predict(X_test)
      r_squared = r2_score(y_test, y_pred)
      print(f'R=squared: {r_squared}')

      R=squared: 0.6620052929422553

[55]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      ss_res = sum([(y_i - model.predict(x_i.reshape(-1, 1))[0])**2
      for x_i, y_i in zip (X_test, y_test)])

      print (f'ss_res: {ss_res}')


      sifamaryamrahman-4115505050210031


      ss_res: 19.1980993608799

[56]: mean_y = np.mean(y_test)
      ss_tot = sum([(y_i - mean_y)**2 for y_i in y_test])

      print (f'ss_tot: {ss_tot}')

      ss_tot: 56.8

[57]: r_squared = 1 - (ss_res / ss_tot)
      print(f'R-squared: {r_squared}')

      R-squared: 0.6620052929422553

[ ]:
```

2. Berikut adalah hasil praktik dari video youtube https://youtu.be/nWJUJenAyB8?si=BQDzWwrMnr8jtzpV

2.1. Persiapan sample dataset



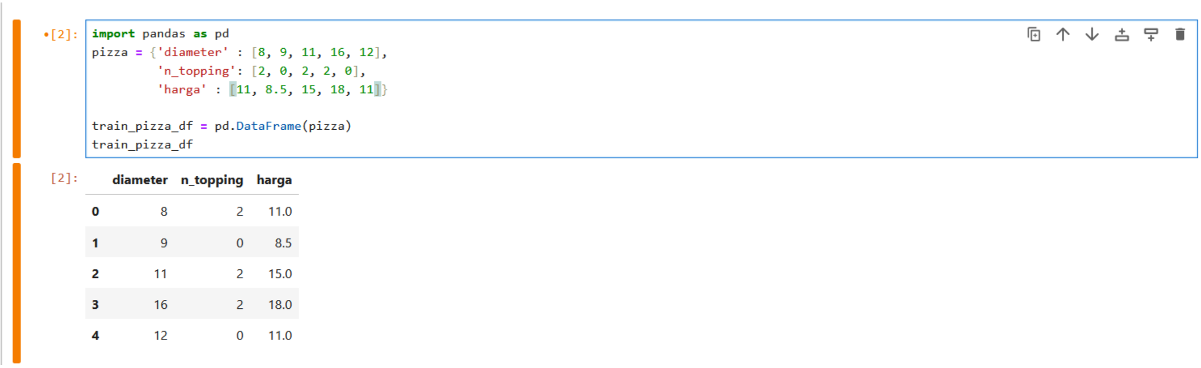
[19]:

```
print('\n')
print('sifamaryamrahman-4115505050210031')
print('\n')

import pandas as pd

pizza = {'diameter' : [6, 8, 10, 14, 18],
         'n_topping': [2, 1, 0, 2, 0],
         'harga' : [7, 9, 13, 17.5, 18]}

train_pizza_df = pd.DataFrame(pizza)
train_pizza_df
```

```


sifamaryamrahman-4115505050210031


```

[19]:

	diameter	n_topping	harga
0	6	2	7.0
1	8	1	9.0
2	10	0	13.0
3	14	2	17.5
4	18	0	18.0



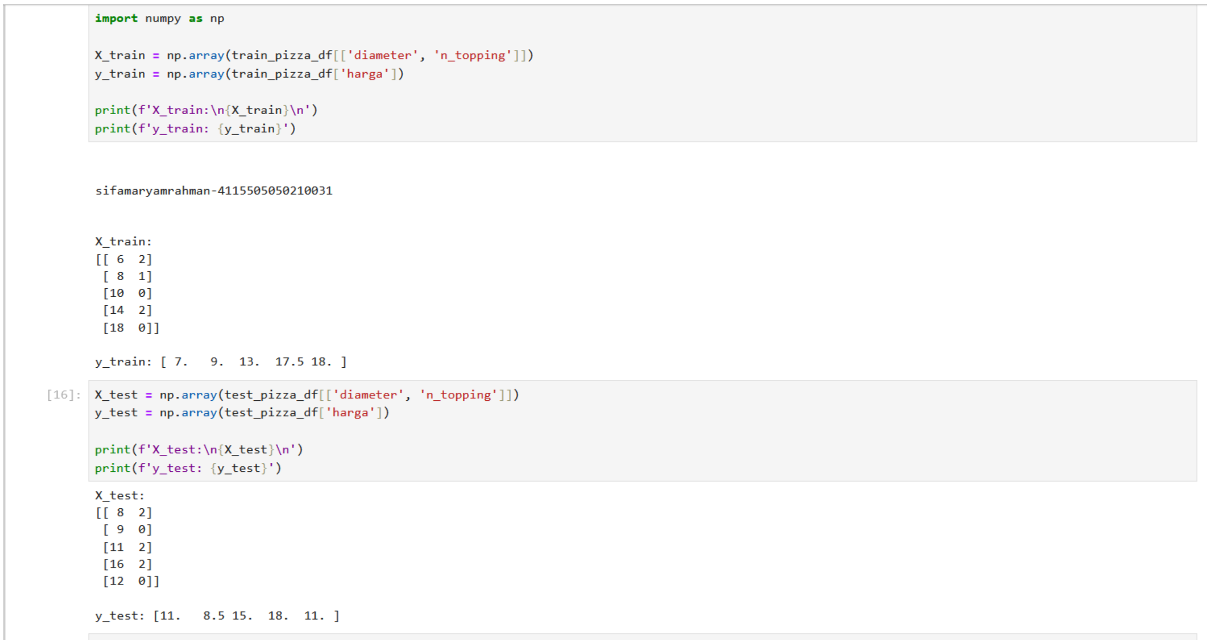
•[2]:

```
import pandas as pd
pizza = {'diameter' : [8, 9, 11, 16, 12],
         'n_topping': [2, 0, 2, 2, 0],
         'harga' : [11, 8.5, 15, 18, 11]}

train_pizza_df = pd.DataFrame(pizza)
train_pizza_df
```

[2]:

	diameter	n_topping	harga
0	8	2	11.0
1	9	0	8.5
2	11	2	15.0
3	16	2	18.0
4	12	0	11.0

2.2.Preprocessing dataset



```
import numpy as np

X_train = np.array(train_pizza_df[['diameter', 'n_topping']])
y_train = np.array(train_pizza_df['harga'])

print(f'X_train:\n{X_train}\n')
print(f'y_train: {y_train}')
```

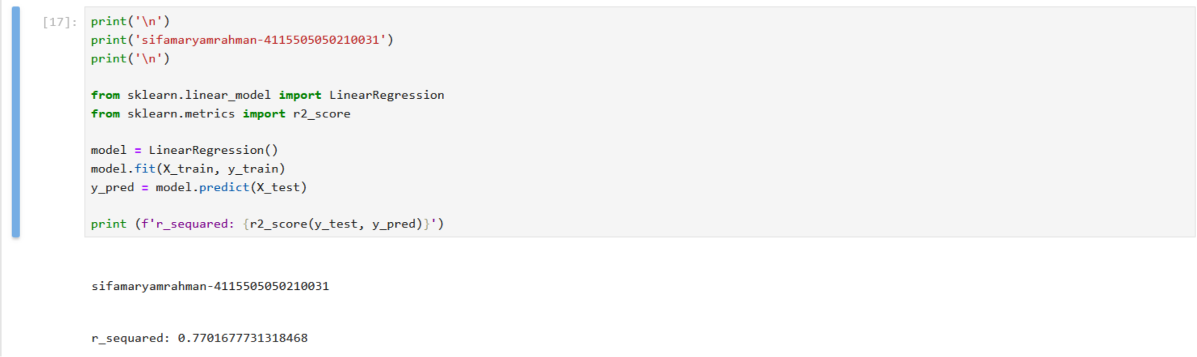
```
sifamaryamrahman-4115505050210031


X_train:
[[ 6  2]
 [ 8  1]
 [10  0]
 [14  2]
 [18  0]]

y_train: [ 7.   9.  13.  17.5 18. ]
```

[16]:

```
X_test = np.array(test_pizza_df[['diameter', 'n_topping']])
y_test = np.array(test_pizza_df['harga'])

print(f'X_test:\n{X_test}\n')
print(f'y_test: {y_test}')
```

```
X_test:
[[ 8  2]
 [ 9  0]
 [11  2]
 [16  2]
 [12  0]]

y_test: [11.   8.5 15.  18.  11. ]
```

2.3. Pengenalan Multiple Linear Regression | Apa itu Multiple Linear Regression?



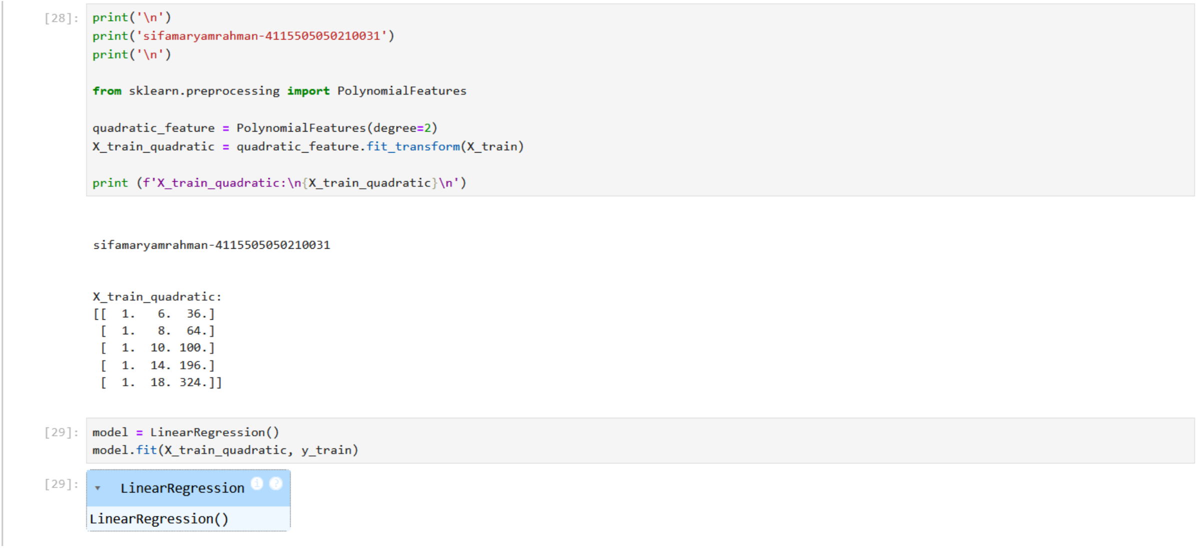
[17]:

```
print('\n')
print('sifamaryamrahman-4115505050210031')
print('\n')

from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print (f'r_sequared: {r2_score(y_test, y_pred)}')
```

```
sifamaryamrahman-4115505050210031


r_sequared: 0.7701677731318468
```

2.4.Pengenalan Polynomial Regression | Apa itu Polynomial Regression?

2.5. Quadratic Polynomial Regression

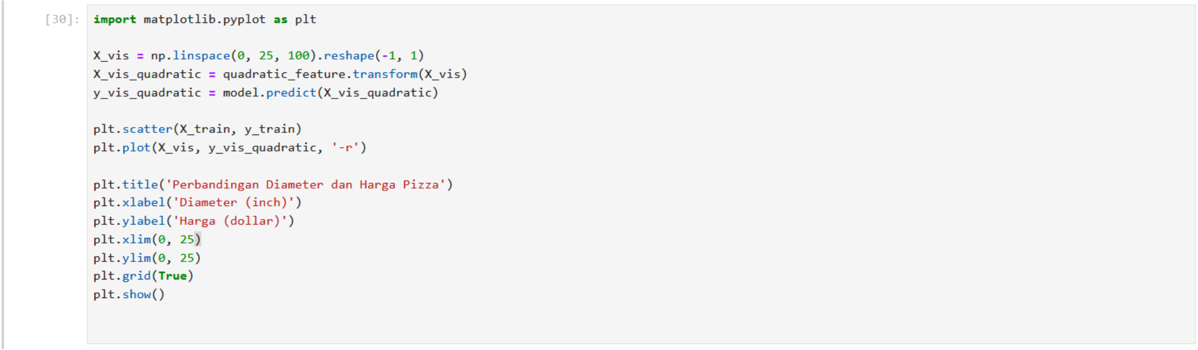


```
[28]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.preprocessing import PolynomialFeatures

      quadratic_feature = PolynomialFeatures(degree=2)
      X_train_quadratic = quadratic_feature.fit_transform(X_train)

      print (f'X_train_quadratic:\n{X_train_quadratic}\n')
```

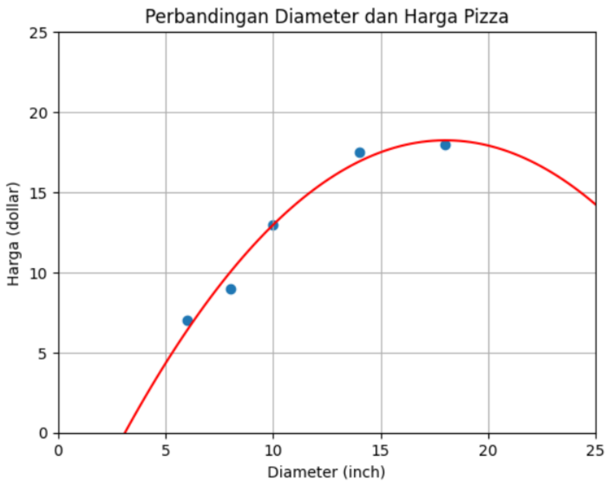
```


sifamaryamrahman-4115505050210031


X_train_quadratic:
[[  1.   6.  36.]
 [  1.   8.  64.]
 [  1.  10. 100.]
 [  1.  14. 196.]
 [  1.  18. 324.]]
```

```
[29]: model = LinearRegression()
      model.fit(X_train_quadratic, y_train)
```

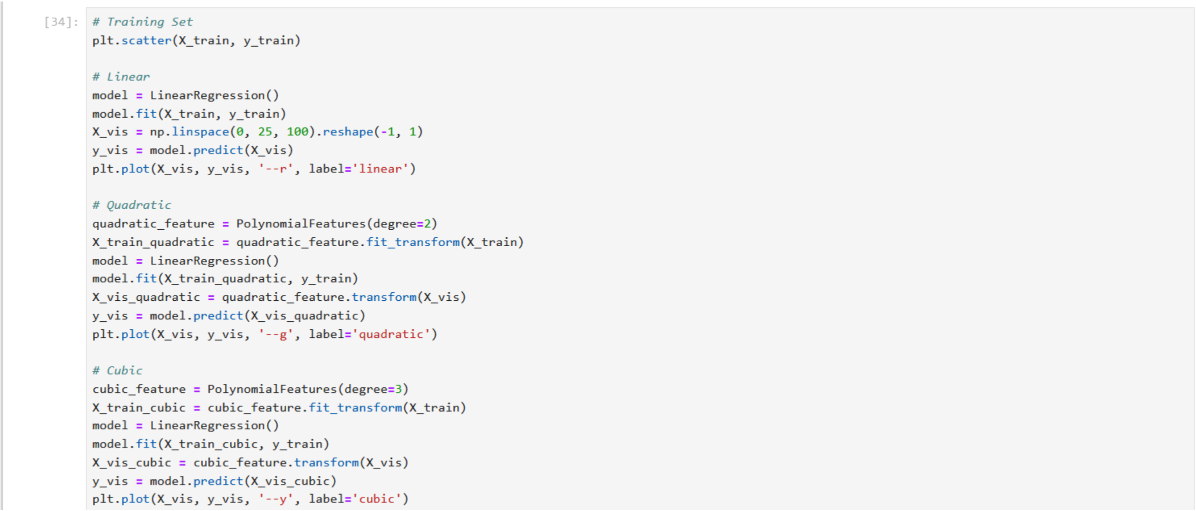
```
[29]: LinearRegression
      LinearRegression()
```



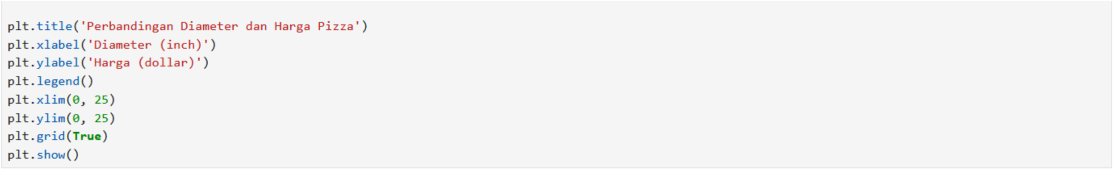
```
[30]: import matplotlib.pyplot as plt

      X_vis = np.linspace(0, 25, 100).reshape(-1, 1)
      X_vis_quadratic = quadratic_feature.transform(X_vis)
      y_vis_quadratic = model.predict(X_vis_quadratic)

      plt.scatter(X_train, y_train)
      plt.plot(X_vis, y_vis_quadratic, '-r')

      plt.title('Perbandingan Diameter dan Harga Pizza')
      plt.xlabel('Diameter (inch)')
      plt.ylabel('Harga (dollar)')
      plt.xlim(0, 25)
      plt.ylim(0, 25)
      plt.grid(True)
      plt.show()
```



2.6. Linear Regression vs Quadratic Polynomial Regression vs Cubic Polynomial Regression

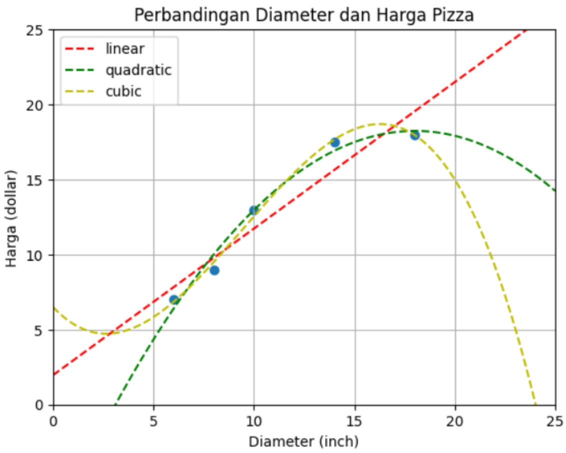


```
[34]: # Training Set
      plt.scatter(X_train, y_train)

      # Linear
      model = LinearRegression()
      model.fit(X_train, y_train)
      X_vis = np.linspace(0, 25, 100).reshape(-1, 1)
      y_vis = model.predict(X_vis)
      plt.plot(X_vis, y_vis, '--r', label='linear')

      # Quadratic
      quadratic_feature = PolynomialFeatures(degree=2)
      X_train_quadratic = quadratic_feature.fit_transform(X_train)
      model = LinearRegression()
      model.fit(X_train_quadratic, y_train)
      X_vis_quadratic = quadratic_feature.transform(X_vis)
      y_vis = model.predict(X_vis_quadratic)
      plt.plot(X_vis, y_vis, '--g', label='quadratic')

      # Cubic
      cubic_feature = PolynomialFeatures(degree=3)
      X_train_cubic = cubic_feature.fit_transform(X_train)
      model = LinearRegression()
      model.fit(X_train_cubic, y_train)
      X_vis_cubic = cubic_feature.transform(X_vis)
      y_vis = model.predict(X_vis_cubic)
      plt.plot(X_vis, y_vis, '--y', label='cubic')
```



```
plt.title('Perbandingan Diameter dan Harga Pizza')
plt.xlabel('Diameter (inch)')
plt.ylabel('Harga (dollar)')
plt.legend()
plt.xlim(0, 25)
plt.ylim(0, 25)
plt.grid(True)
plt.show()
```

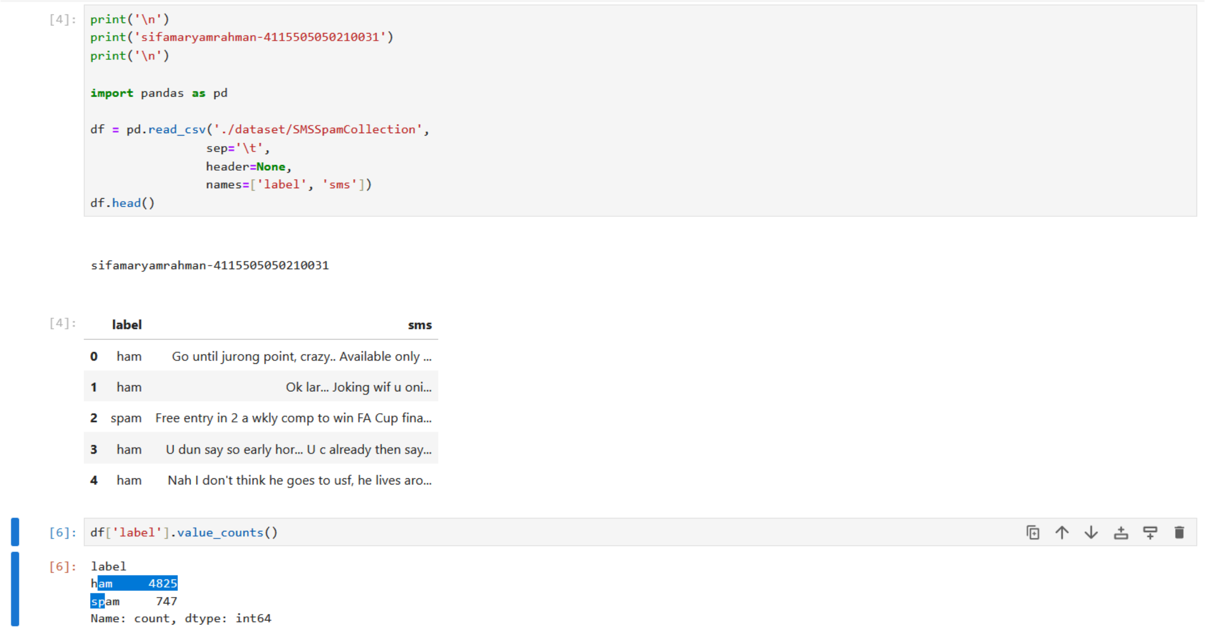
```
sifamaryamrahman-4115505050210031
```



3. Berikut adalah hasil praktik dari video youtube https://youtu.be/oe7DW4rSH1o?si=H-PZJ9rs9-Kab-Ln

3.1.Formula dasar pembentuk Logistic Regression | Fungsi Sigmoid

3.2.Persiapan dataset | SMS Spam Collection Dataset



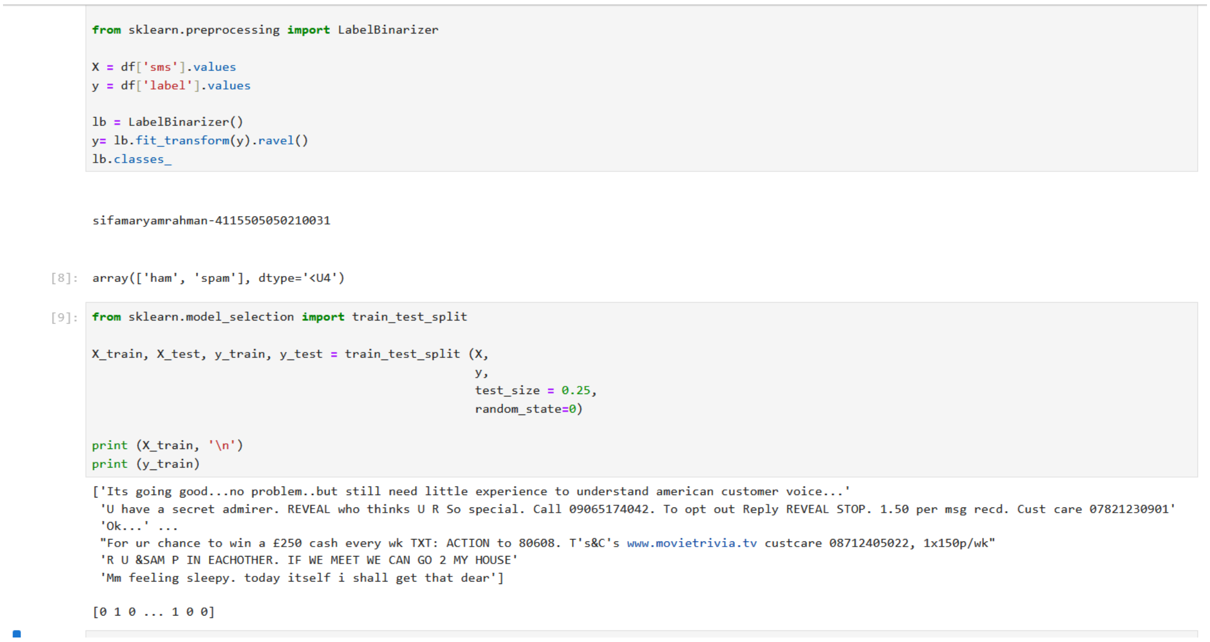
```
[4]: print('\n')
     print('sifamaryamrahman-4115505050210031')
     print('\n')

     import pandas as pd

     df = pd.read_csv('./dataset/SMSSpamCollection',
                      sep='\t',
                      header=None,
                      names=['label', 'sms'])
     df.head()


     sifamaryamrahman-4115505050210031
```

[4]:

	label	sms
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

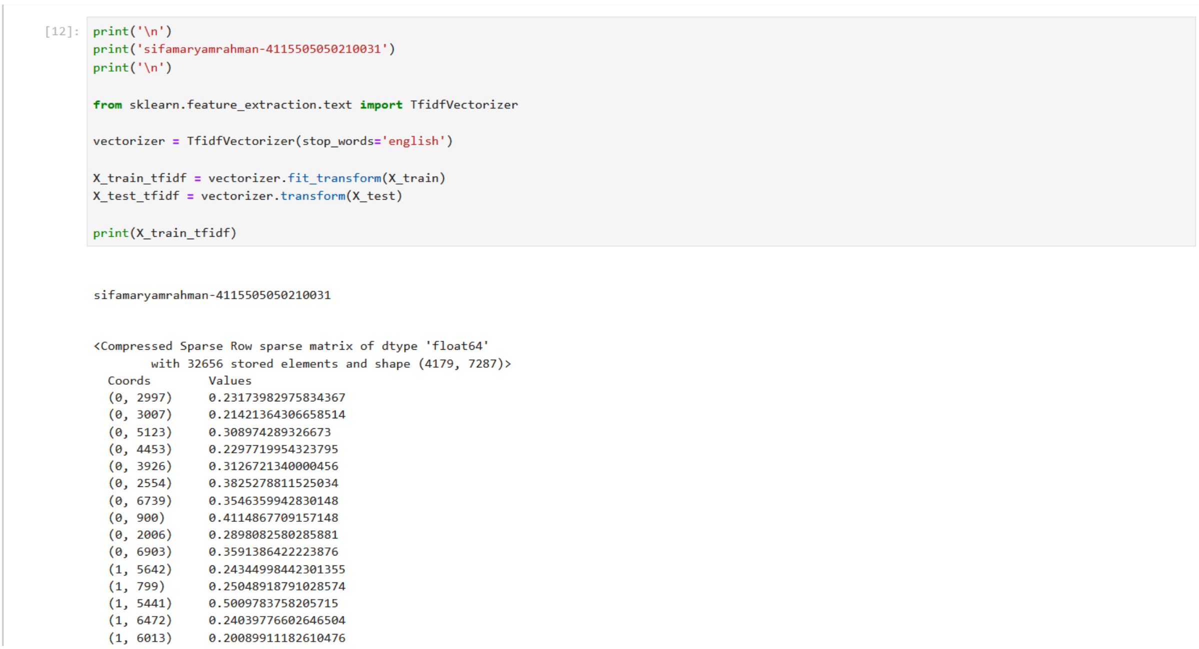
```
[6]: df['label'].value_counts()

[6]: label
     ham      4825
     spam      747
     Name: count, dtype: int64
```

3.3.Pembagian training dan testing set



```
     from sklearn.preprocessing import LabelBinarizer

     X = df['sms'].values
     y = df['label'].values

     lb = LabelBinarizer()
     y= lb.fit_transform(y).ravel()
     lb.classes_


     sifamaryamrahman-4115505050210031

[8]: array(['ham', 'spam'], dtype='<U4')

[9]: from sklearn.model_selection import train_test_split

     X_train, X_test, y_train, y_test = train_test_split (X,
                                                          y,
                                                          test_size = 0.25,
                                                          random_state=0)

     print (X_train, '\n')
     print (y_train)
     ['Its going good...no problem..but still need little experience to understand american customer voice...'
      'U have a secret admirer. REVEAL who thinks U R So special. Call 09065174042. To opt out Reply REVEAL STOP. 1.50 per msg recd. Cust care 07821230901'
      'Ok...' ...
      "For ur chance to win a £250 cash every wk TXT: ACTION to 80608. T's&C's www.movietrivia.tv custcare 08712405022, 1x150p/wk"
      'R U &SAM P IN EACHOTHER. IF WE MEET WE CAN GO 2 MY HOUSE'
      'Mm feeling sleepy. today itself i shall get that dear']

     [0 1 0 ... 1 0 0]
```

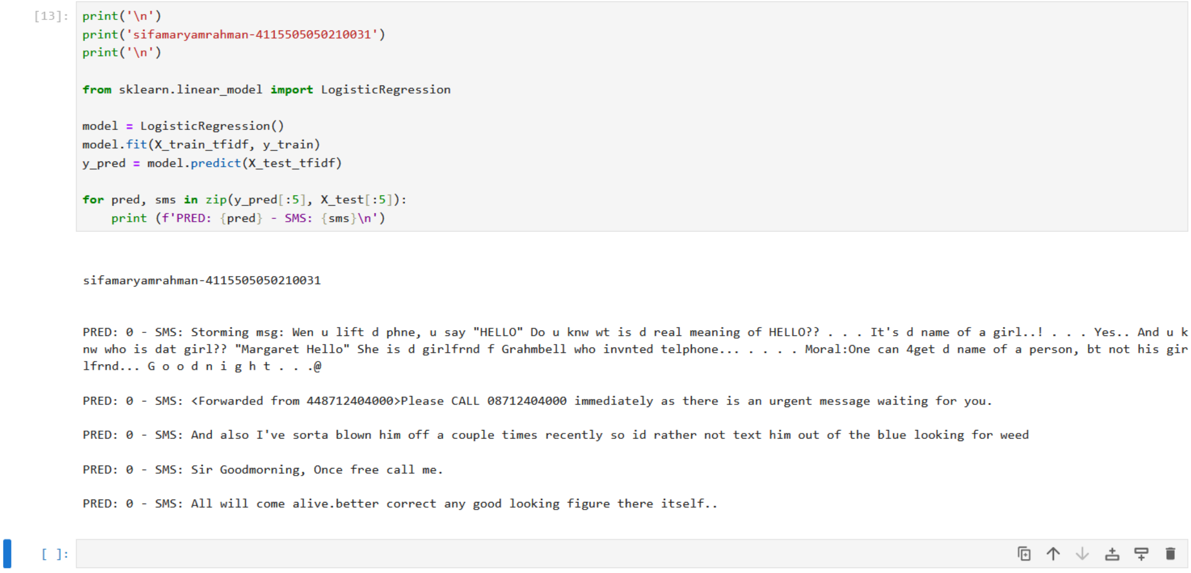
3.4.Feature extraction dengan TF-IDF



```
[12]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.feature_extraction.text import TfidfVectorizer

      vectorizer = TfidfVectorizer(stop_words='english')

      X_train_tfidf = vectorizer.fit_transform(X_train)
      X_test_tfidf = vectorizer.transform(X_test)

      print(X_train_tfidf)
```

```


sifamaryamrahman-4115505050210031


<Compressed Sparse Row sparse matrix of dtype 'float64'
	with 32656 stored elements and shape (4179, 7287)>
  Coords	Values
  (0, 2997)	0.23173982975834367
  (0, 3007)	0.21421364306658514
  (0, 5123)	0.308974289326673
  (0, 4453)	0.2297719954323795
  (0, 3926)	0.3126721340000456
  (0, 2554)	0.3825278811525034
  (0, 6739)	0.3546359942830148
  (0, 900)	0.4114867709157148
  (0, 2006)	0.2898082580285881
  (0, 6903)	0.3591386422223876
  (1, 5642)	0.24344998442301355
  (1, 799)	0.25048918791028574
  (1, 5441)	0.5009783758205715
  (1, 6472)	0.24039776602646504
  (1, 6013)	0.20089911182610476
```

3.5.Binary Classification dengan Logistic Regression



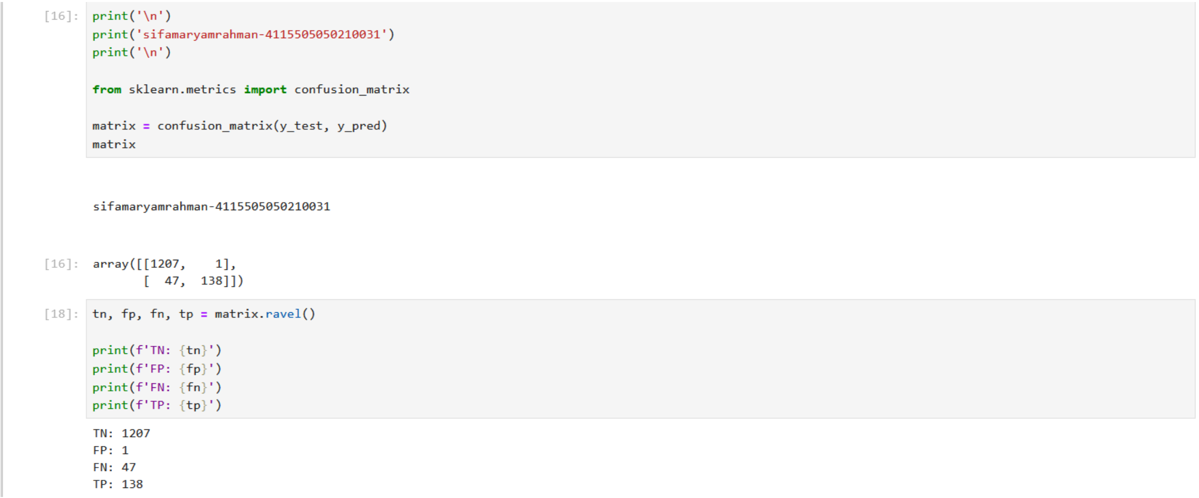
```
[13]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.linear_model import LogisticRegression

      model = LogisticRegression()
      model.fit(X_train_tfidf, y_train)
      y_pred = model.predict(X_test_tfidf)

      for pred, sms in zip(y_pred[:5], X_test[:5]):
          print (f'PRED: {pred} - SMS: {sms}\n')
```

```


sifamaryamrahman-4115505050210031


PRED: 0 - SMS: Storming msg: Wen u lift d phne, u say "HELLO" Do u knw wt is d real meaning of HELLO?? . . . It's d name of a girl..! . . . Yes.. And u k
nw who is dat girl?? "Margaret Hello" She is d girlfrnd f Grahmbell who invnted telphone... . . . . Moral:One can 4get d name of a person, bt not his gir
lfrnd... G o o d n i g h t . . .@

PRED: 0 - SMS: <Forwarded from 448712404000>Please CALL 08712404000 immediately as there is an urgent message waiting for you.

PRED: 0 - SMS: And also I've sorta blown him off a couple times recently so id rather not text him out of the blue looking for weed

PRED: 0 - SMS: Sir Goodmorning, Once free call me.

PRED: 0 - SMS: All will come alive.better correct any good looking figure there itself..
```

```
[ ]:
```

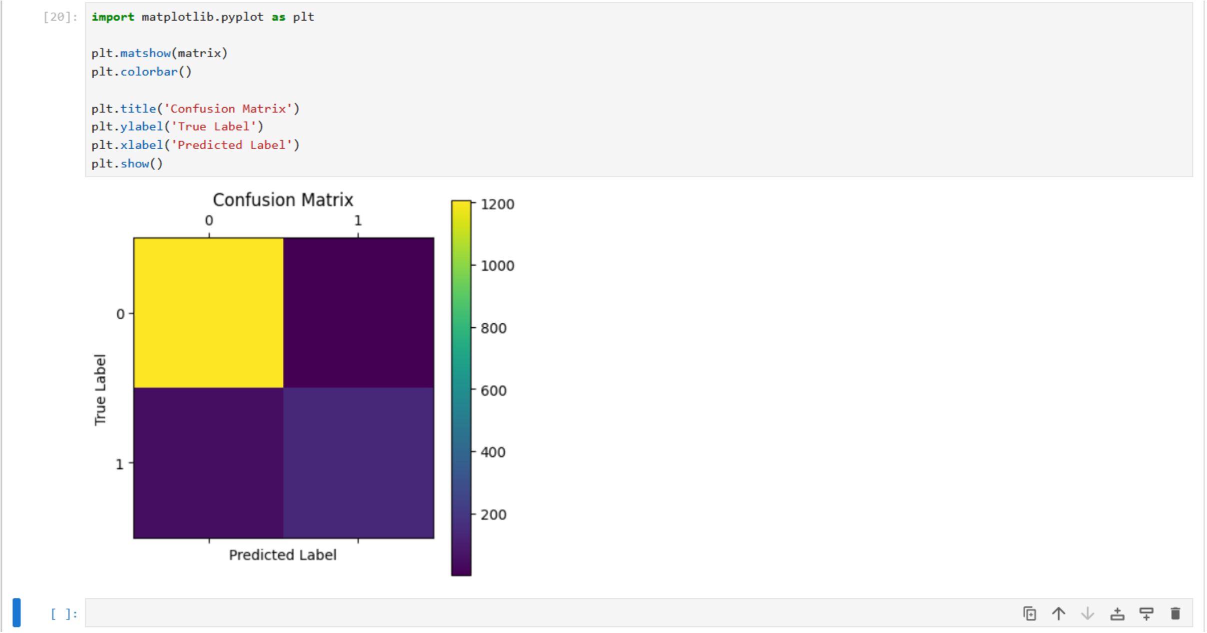
3.6.Evaluation Metrics pada Binary Classification Task

3.7.Pengenalan Confusion Matrix



```
[16]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.metrics import confusion_matrix

      matrix = confusion_matrix(y_test, y_pred)
      matrix


      sifamaryamrahman-4115505050210031


[16]: array([[1207,    1],
             [  47,  138]])
```

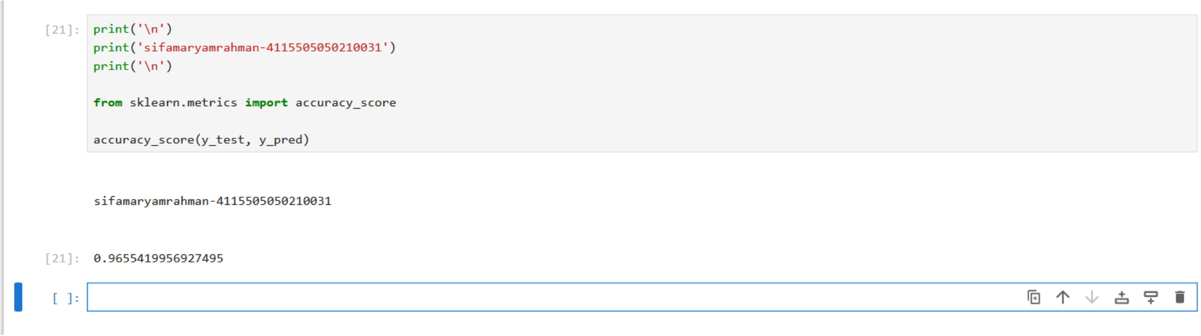
```
[18]: tn, fp, fn, tp = matrix.ravel()

      print(f'TN: {tn}')
      print(f'FP: {fp}')
      print(f'FN: {fn}')
      print(f'TP: {tp}')

      TN: 1207
      FP: 1
      FN: 47
      TP: 138
```



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

      plt.matshow(matrix)
      plt.colorbar()

      plt.title('Confusion Matrix')
      plt.ylabel('True Label')
      plt.xlabel('Predicted Label')
      plt.show()
```

```
[ ]:
```

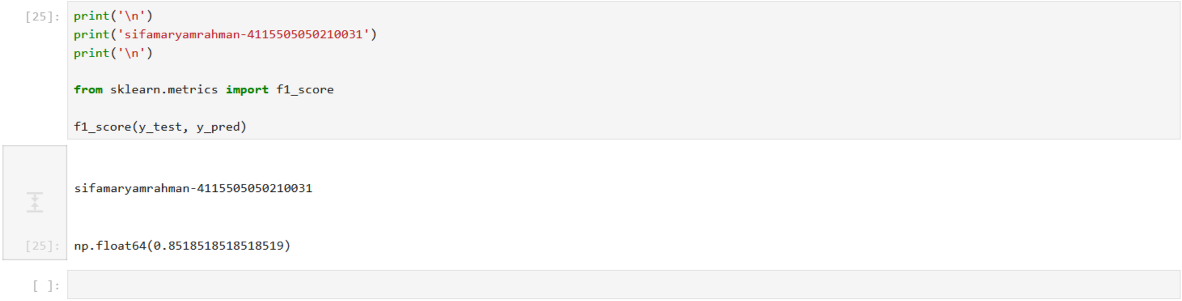
3.8.Pengenalan Accuracy Score



```
[21]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.metrics import accuracy_score

      accuracy_score(y_test, y_pred)


      sifamaryamrahman-4115505050210031


[21]: 0.96554199569274 95
[ ]:
```

3.9.Pengenalan Precision dan Recall

3.10. Pengenalan F1 Score | F1 Measure

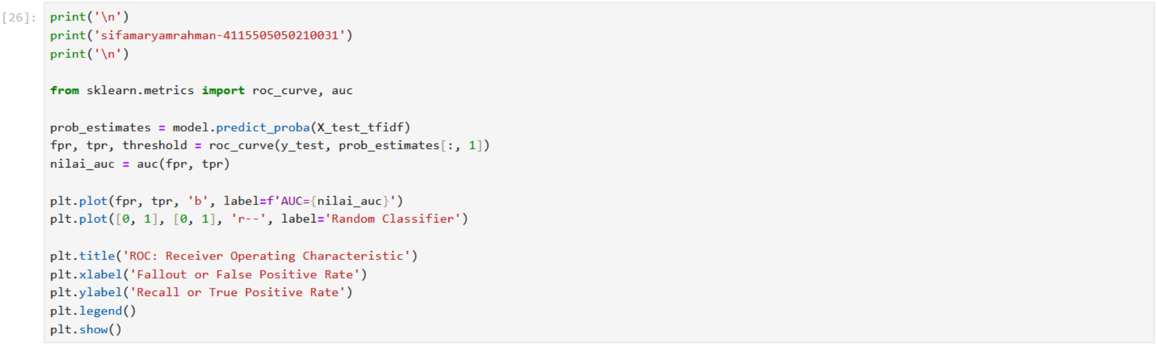


```
[25]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.metrics import f1_score

      f1_score(y_test, y_pred)
```

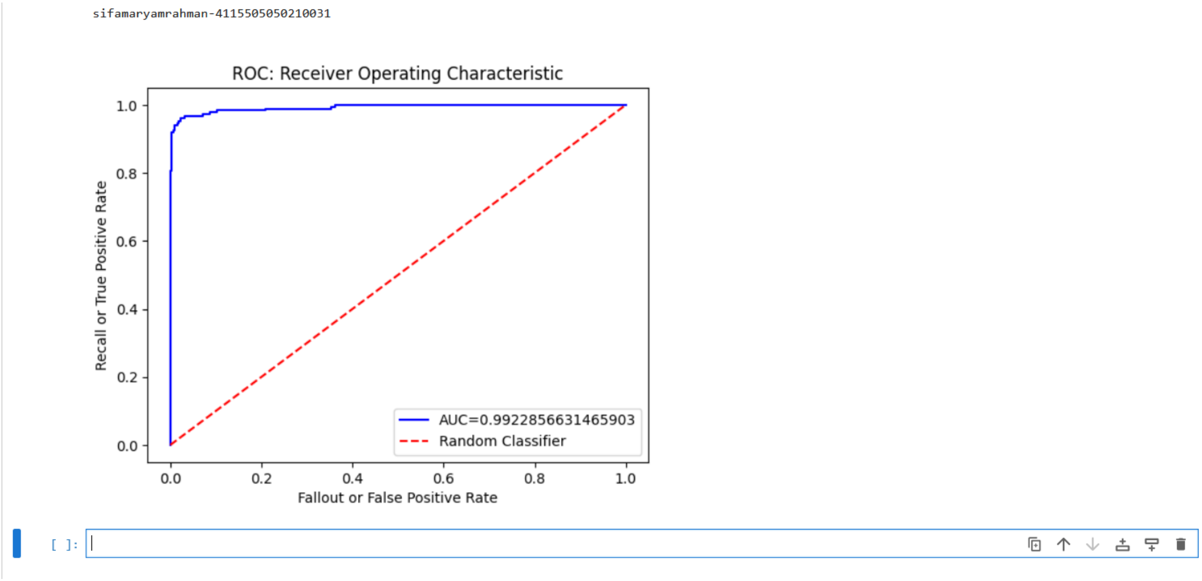
```


sifamaryamrahman-4115505050210031



[25]: np.float64(0.8518518518518519)
```

```
[ ]:
```

3.11. Pengenalan ROC | Receiver Operating Characteristic



```
[26]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.metrics import roc_curve, auc

      prob_estimates = model.predict_proba(X_test_tfidf)
      fpr, tpr, threshold = roc_curve(y_test, prob_estimates[:, 1])
      nilai_auc = auc(fpr, tpr)

      plt.plot(fpr, tpr, 'b', label=f'AUC={nilai_auc}')
      plt.plot([0, 1], [0, 1], 'r--', label='Random Classifier')

      plt.title('ROC: Receiver Operating Characteristic')
      plt.xlabel('Fallout or False Positive Rate')
      plt.ylabel('Recall or True Positive Rate')
      plt.legend()
      plt.show()
```



```


sifamaryamrahman-4115505050210031


```

ROC: Receiver Operating Characteristic
1.0
0.8
0.6
0.4
0.2
0.0
Recall or True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Fallout or False Positive Rate
AUC=0.9922856631465903
Random Classifier

```
[ ]:
```