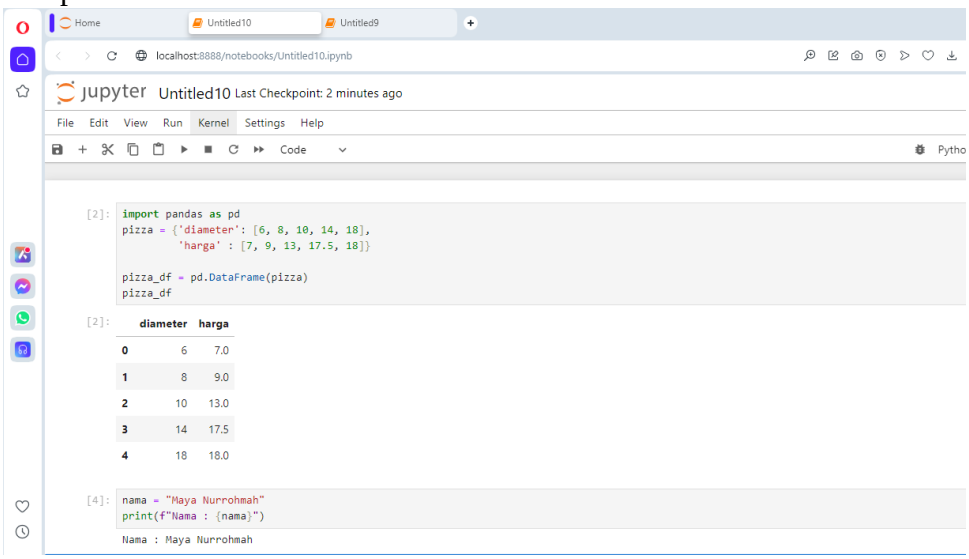
TUGAS PERTEMUAN 2

NAMA : MAYA NURROHMAH

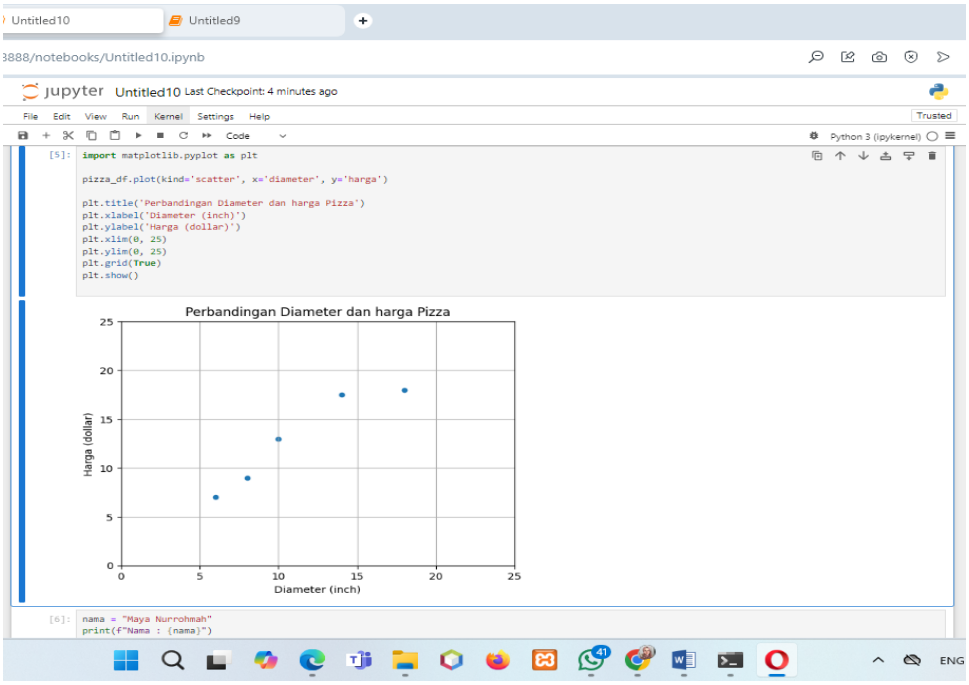
NPM : 41155050210019

KELAS : INF_A1

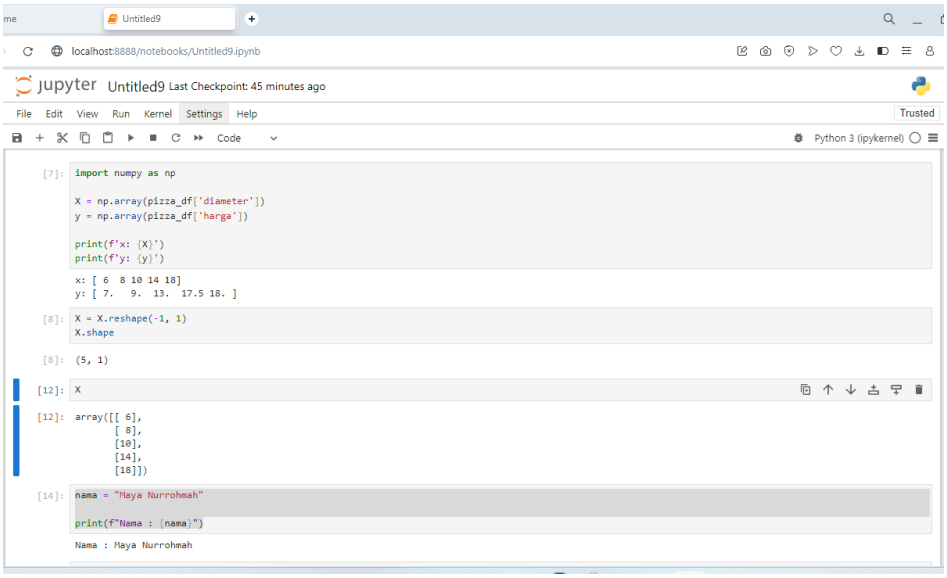
1. Lakukan praktek dari https://youtu.be/lcjq7-2zMSA?si=f4jWJR6lY8y0BZKl dan buat screen shot hasil run dengan nama anda pada hasil run tersebut. Praktek tersebut yaitu:
   1) Sample dataset



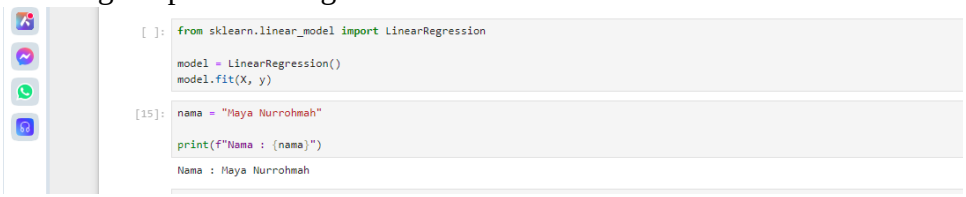
   2) Visualisasi dataset



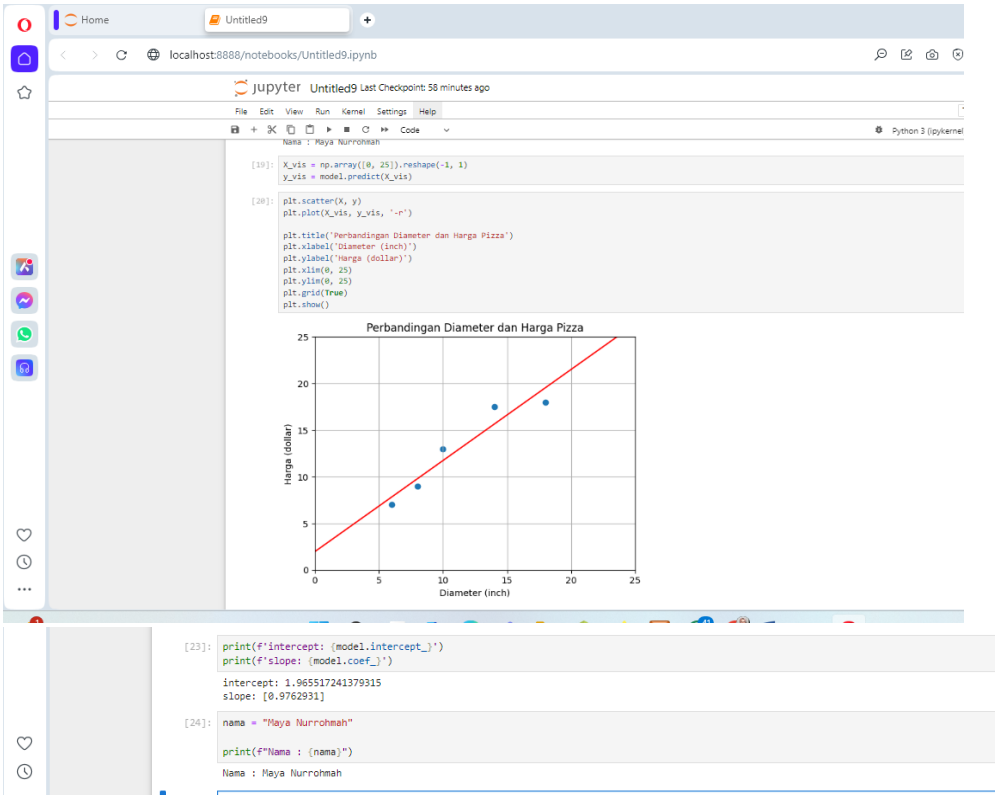
3) Transformasi dataset



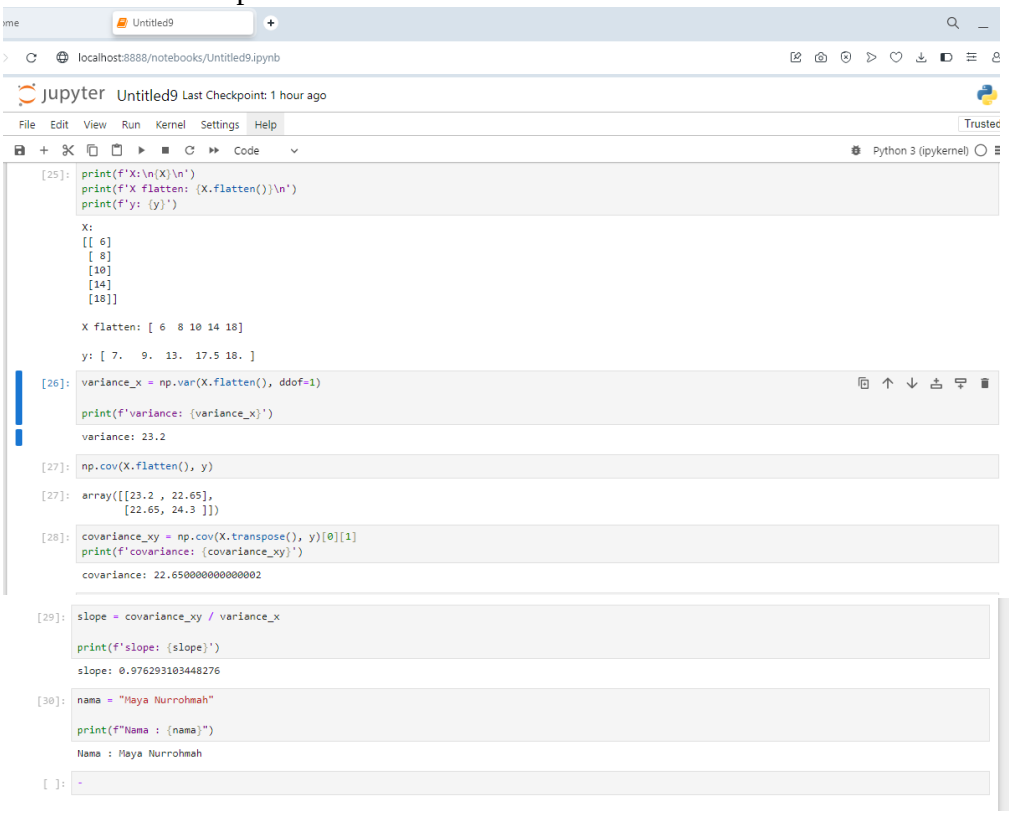
4) Training simple linear regression model



5) Visualisasi Simple Linear Regression Model | Penjelasan persamaan garis linear

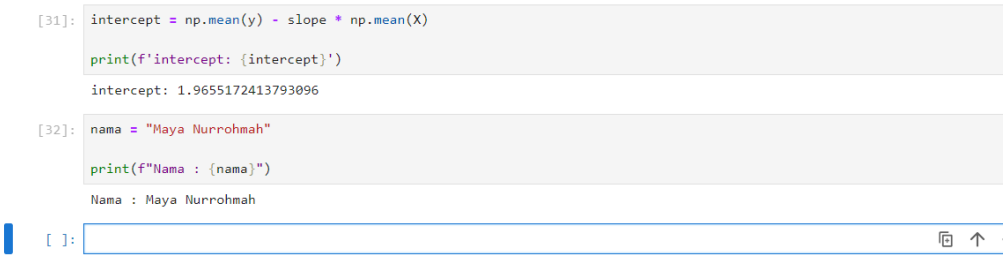


6) Kalkulasi nilai slope



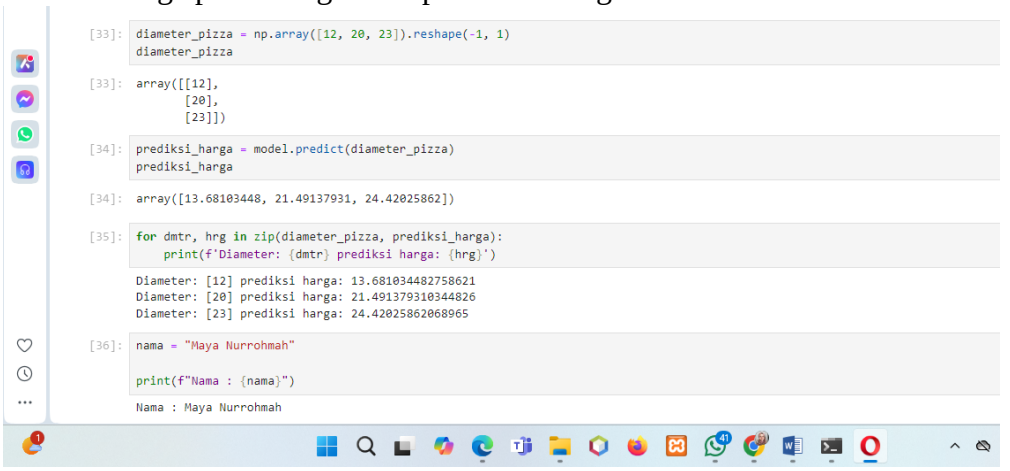
```
[25]: print(f'X:\n{X}\n')
      print(f'X flatten: {X.flatten()}\n')
      print(f'y: {y}')

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

      X flatten: [ 6  8 10 14 18]

      y: [ 7.   9.  13.  17.5 18. ]

[26]: variance_x = np.var(X.flatten(), ddof=1)

      print(f'variance: {variance_x}')

      variance: 23.2

[27]: np.cov(X.flatten(), y)

[27]: array([[23.2 , 22.65],
             [22.65, 24.3 ]])

[28]: covariance_xy = np.cov(X.transpose(), y)[0][1]
      print(f'covariance: {covariance_xy}')

      covariance: 22.650000000000002

[29]: slope = covariance_xy / variance_x

      print(f'slope: {slope}')

      slope: 0.976293103448276

[30]: nama = "Maya Nurrohmah"

      print(f"Nama : {nama}")

      Nama : Maya Nurrohmah

[ ]:
```

7) Kalkulasi nilai intercept



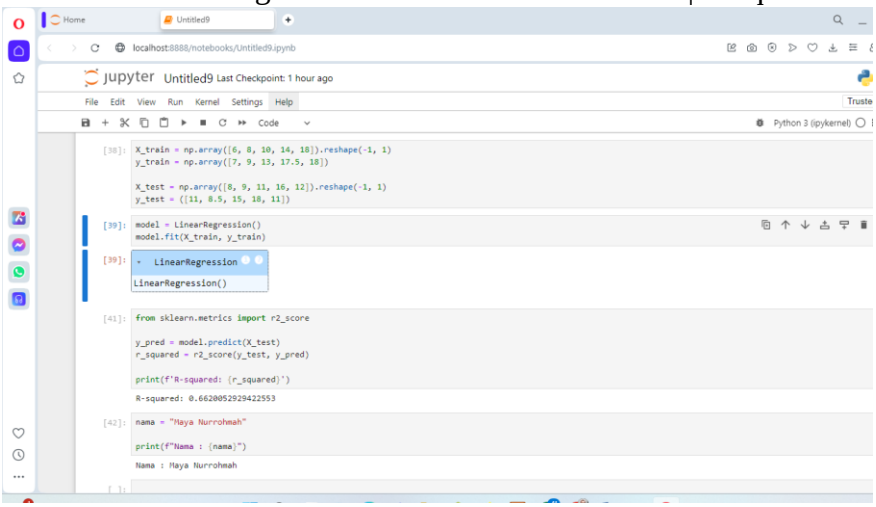
```
[31]: intercept = np.mean(y) - slope * np.mean(X)

      print(f'intercept: {intercept}')

      intercept: 1.9655172413793096

[32]: nama = "Maya Nurrohmah"

      print(f"Nama : {nama}")

      Nama : Maya Nurrohmah

[ ]:
```

8) Prediksi harga pizza dengan Simple Linear Regression Model

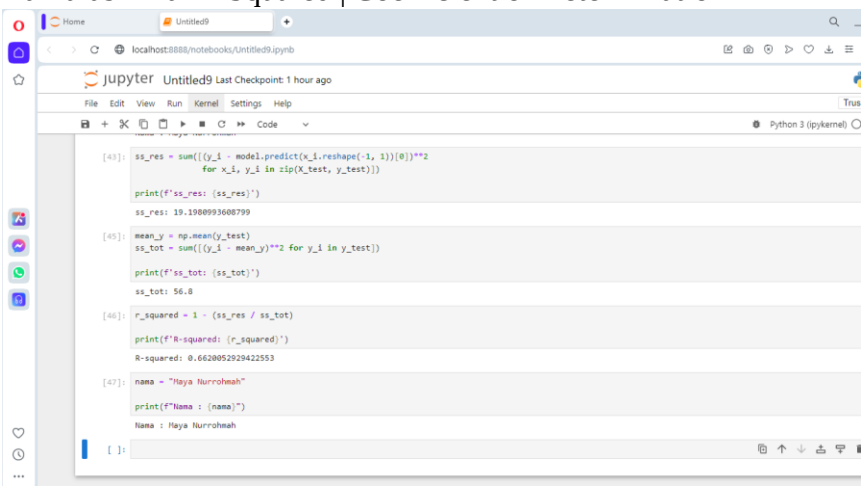


```
[33]: diameter_pizza = np.array([12, 20, 23]).reshape(-1, 1)
      diameter_pizza

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

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

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

[35]: 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

[36]: nama = "Maya Nurrohmah"

      print(f"Nama : {nama}")

      Nama : Maya Nurrohmah
```

9) Evaluasi model dengan Coefficient of Determination | R Squared

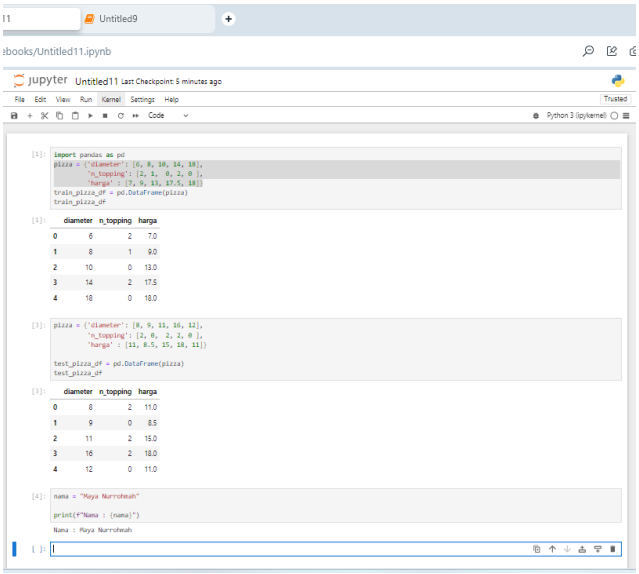


10) Kalkulasi nilai R Squared | Coefficient of Determination

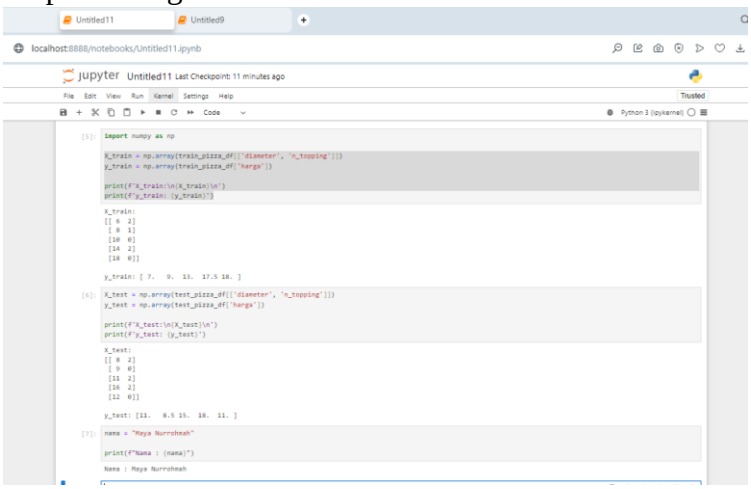


2. Lakukan praktek dari https://youtu.be/nWJUJenAyB8?si=BQDzWwrMnr8jtzpV dan buat screen shot hasil run dengan nama anda pada hasil run tersebut. Praktek tersebut yaitu:

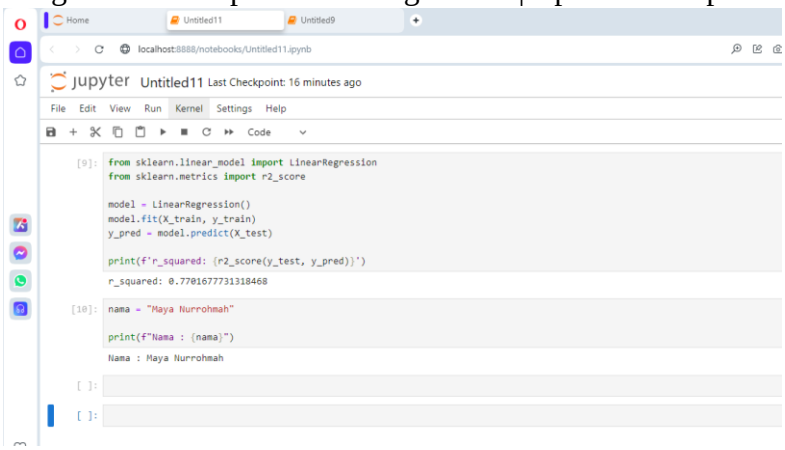
1) Persiapan sample dataset



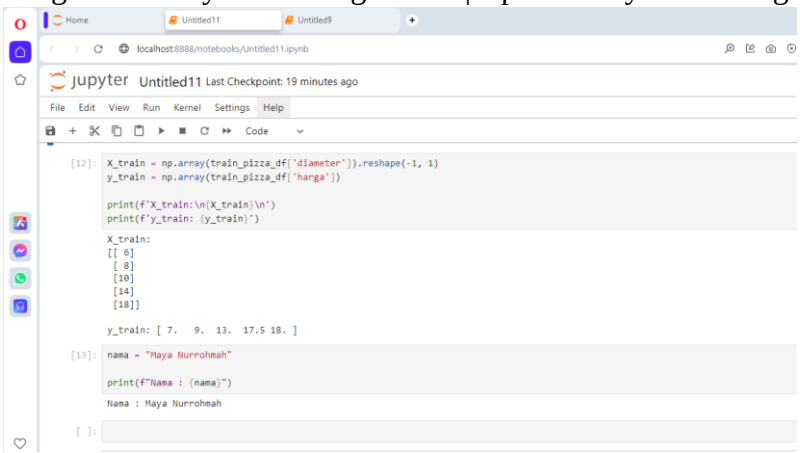
2) Preprocessing dataset



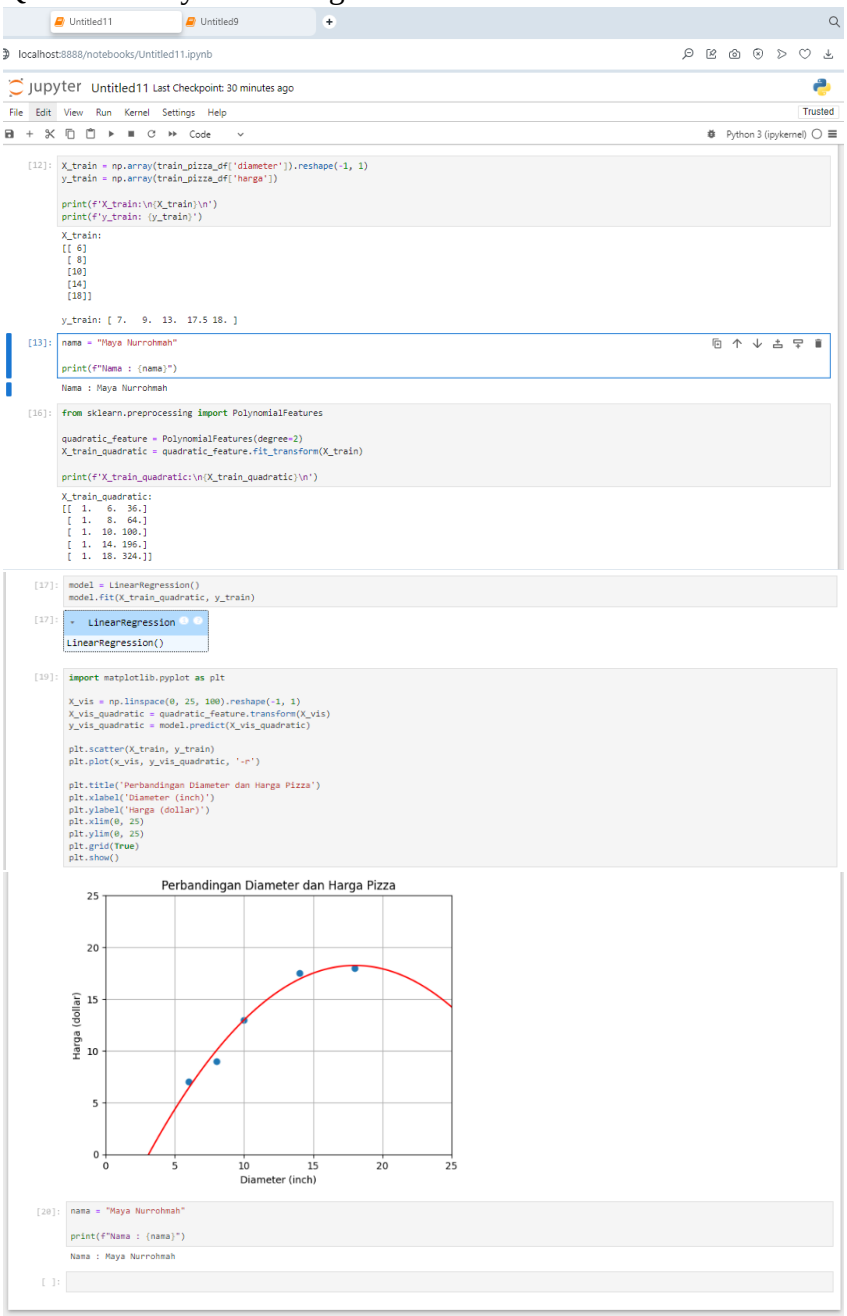
3) Pengenalan Multiple Linear Regression | Apa itu Multiple Linear Regression?



4) Pengenalan Polynomial Regression | Apa itu Polynomial Regression?



5) Quadratic Polynomial Regression



```
[12]: X_train = np.array(train_pizza_df['diameter']).reshape(-1, 1)
      y_train = np.array(train_pizza_df['harga'])

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

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

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

[13]: nama = "Maya Nurrohmah"

      print(f"Nama : {nama}")

Nama : Maya Nurrohmah

[16]: 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')

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

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

[17]: LinearRegression()

[19]: 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()
```

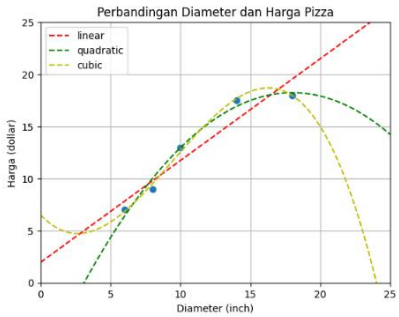


```
[20]: nama = "Maya Nurrohmah"

      print(f"Nama : {nama}")

Nama : Maya Nurrohmah

[ ]:
```

6) Linear Regression vs Quadratic Polynomial Regression vs Cubic Polynomial Regression

```
[27]: # 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_quadratic = model.predict(X_vis_quadratic)
      plt.plot(X_vis, y_vis_quadratic, '--g', label='quadratic')

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

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

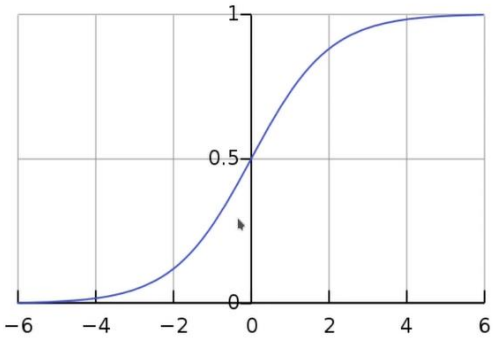
```
plt.xlim(0, 25)
plt.ylim(0, 25)
plt.grid(True)
plt.legend()
plt.show()
```



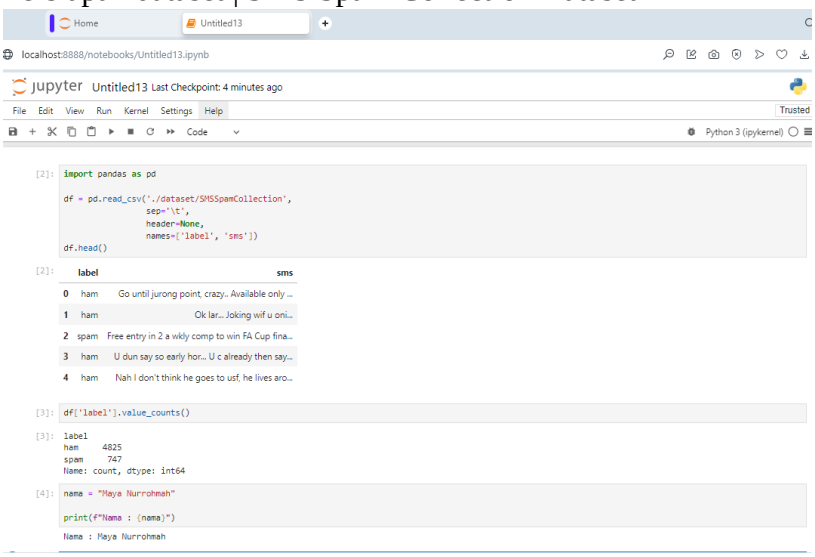

```
[28]: nama = "Maya Nurrohmah"

print(f"Nama : {nama}")
```

```
Nama : Maya Nurrohmah
```

3. Lakukan praktek dari https://youtu.be/oe7DW4rSH1o?si=H-PZJ9rs9-Kab-Ln dan buat screen shot hasil run dengan nama anda pada hasil run tersebut. Praktek tersebut yaitu:
    1) Formula dasar pembentuk Logistic Regression | Fungsi Sigmoid
        - Simple Linear Regression
          Formula dari Simple Linear Regression
          $y= \alpha+\beta X$
          $g(X)=\alpha+\beta X$
        - Multiple Linear Regression
          Formula dari Multiple Linear Regression
          $y=\alpha+\beta_1 x_1 + \beta_2 x_2+\cdots+\beta_n x_n$
          $Xg(X)=\alpha+\beta X$
        - Logistic Regression
          Formula dari Logistic Regression
          $g(X)=\text{sigmoid}(\alpha+\beta X)$
          $\text{sigmoid}(x) = \frac{1}{1+exp(-x)}$



2) Persiapan dataset | SMS Spam Collection Dataset



```
[2]: import pandas as pd

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

[2]:

	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...

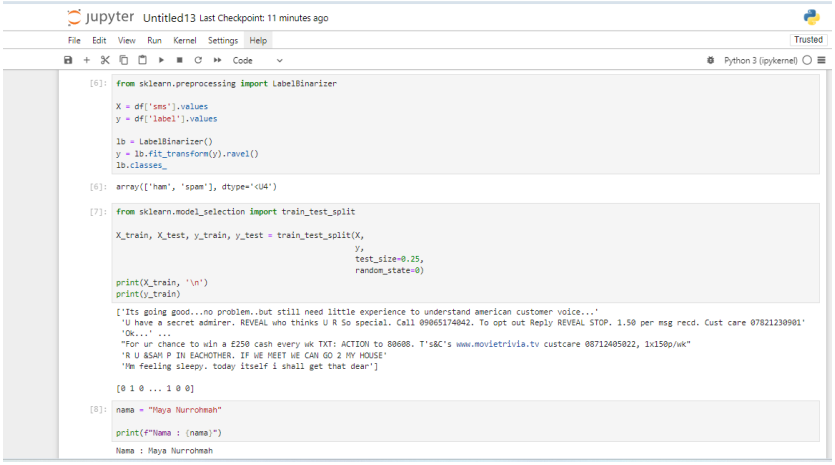
```
[3]: df['label'].value_counts()
```

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

```
[4]: nama = "Maya Nurrohmah"

     print(f"Nama : {nama}")
```

```
Nama : Maya Nurrohmah
```

3) Pembagian training dan testing set



```
[6]: from sklearn.preprocessing import LabelBinarizer

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

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

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

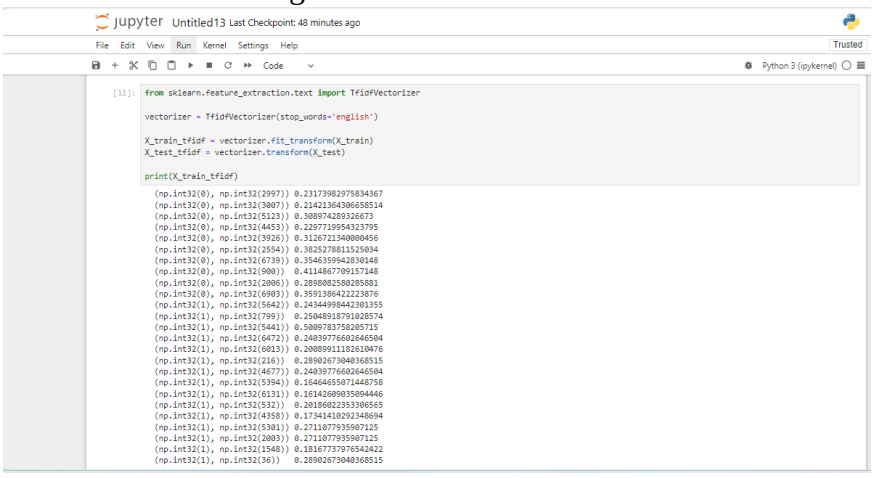
```
[7]: from sklearn.model_selection import train_test_split

     X_train, X_test, y_train, y_test = train_test_split(X,
                                                         y,
                                                         test_size=0.25,
                                                         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"
 'A U &SAH P IN EACHOTHER. IF WE MEET WE CAN GO 2 MY HOUSE'
 'Wm feeling sleepy. today itself i shall get that dear']

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

```
[8]: nama = "Maya Nurrohmah"

     print(f"Nama : {nama}")
```

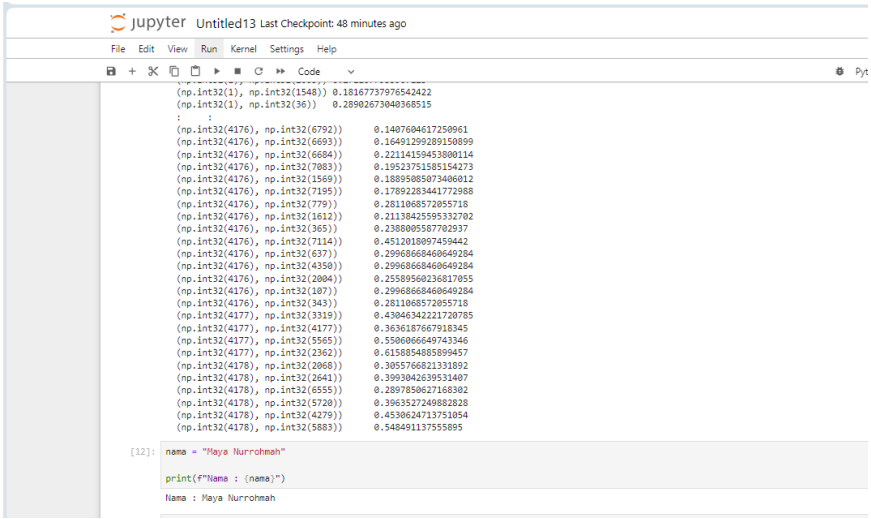
```
Nama : Maya Nurrohmah
```

4) Feature extraction dengan TF-IDF

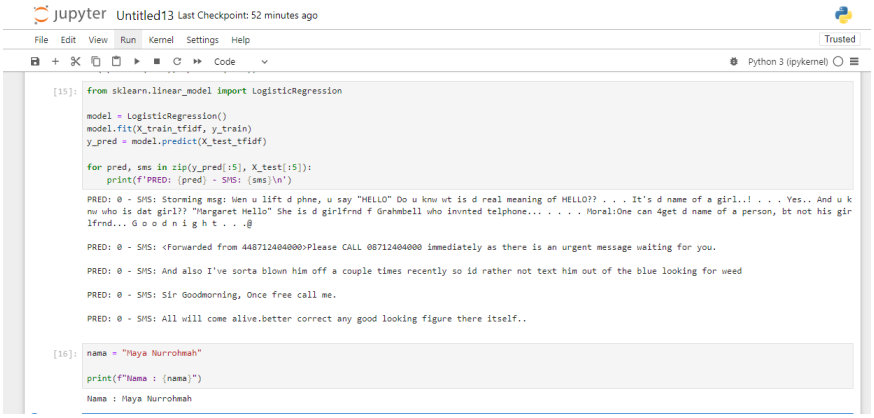


```
[11]: from sklearn.feature_extraction.text import TfidfVectorizer

      vectorizer = TfidfVectorizer(stop_words='english')

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

      print(X_train_tfidf)
```

```
  (np.int32(0), np.int32(2997)) 0.23173982975834367
  (np.int32(0), np.int32(3007)) 0.21421364306658514
  (np.int32(0), np.int32(5123)) 0.308974289326673
  (np.int32(0), np.int32(4453)) 0.2297719954323795
  (np.int32(0), np.int32(3926)) 0.3126721340000456
  (np.int32(0), np.int32(2554)) 0.3825278811525034
  (np.int32(0), np.int32(6739)) 0.3546359942830148
  (np.int32(0), np.int32(900))  0.4114867709157148
  (np.int32(0), np.int32(2006)) 0.2898082580285881
  (np.int32(0), np.int32(6903)) 0.3591386422223876
  (np.int32(1), np.int32(5642)) 0.24344998442301355
  (np.int32(1), np.int32(799))  0.25048918791028574
  (np.int32(1), np.int32(5441)) 0.5009783758205715
  (np.int32(1), np.int32(6472)) 0.24039776602646504
  (np.int32(1), np.int32(6013)) 0.20089911182610476
  (np.int32(1), np.int32(216))  0.28902673040368515
  (np.int32(1), np.int32(4677)) 0.24039776602646504
  (np.int32(1), np.int32(5394)) 0.16464665071448758
  (np.int32(1), np.int32(6131)) 0.16142609035094446
  (np.int32(1), np.int32(532))  0.20186022353306565
  (np.int32(1), np.int32(4358)) 0.17341410292348694
  (np.int32(1), np.int32(5301)) 0.2711077935907125
  (np.int32(1), np.int32(2003)) 0.2711077935907125
  (np.int32(1), np.int32(1548)) 0.18167737976542422
  (np.int32(1), np.int32(36))   0.28902673040368515
```



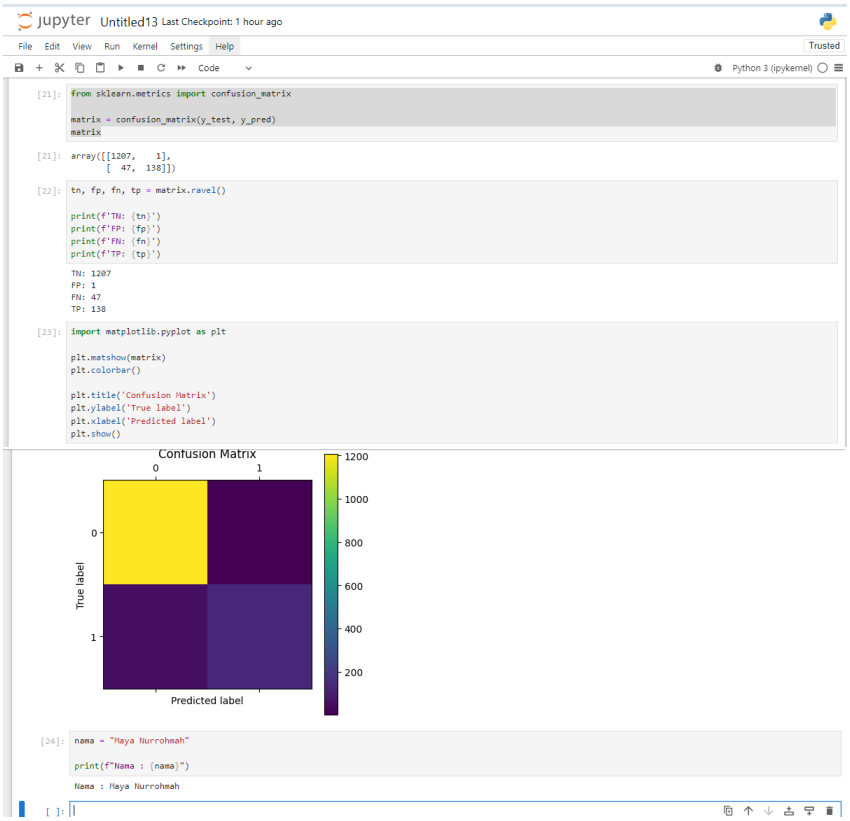
5) Binary Classification dengan Logistic Regression



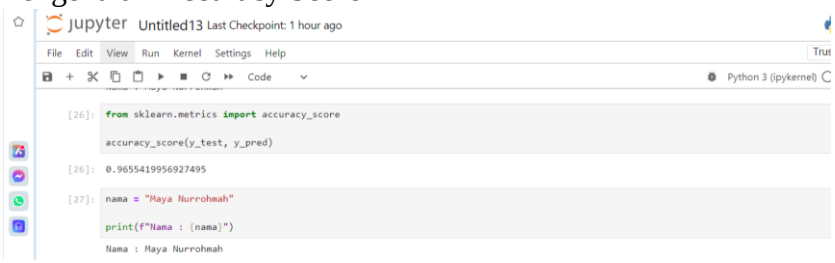
6) Evaluation Metrics pada Binary Classification Task
   - Confusion Matrix
   - Accuracy
   - Precission & Recall
   - F1 Score
   - ROC

   Terminologi Dasar
   - True Positive (TP)
   - True Negative (TN)
   - False Positive (FP)
   - False Negative (FN)

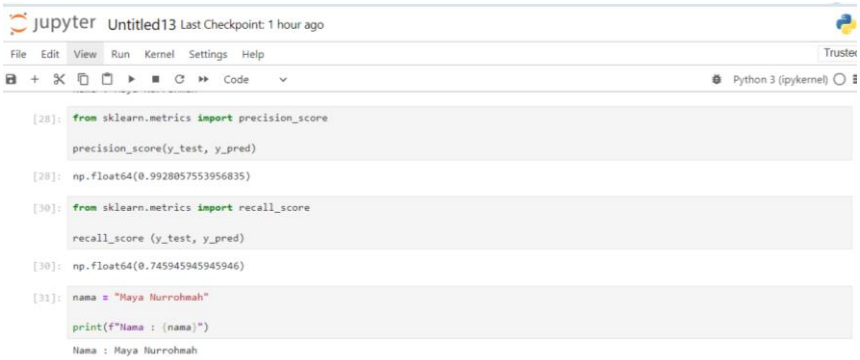
7) Pengenalan Confusion Matrix



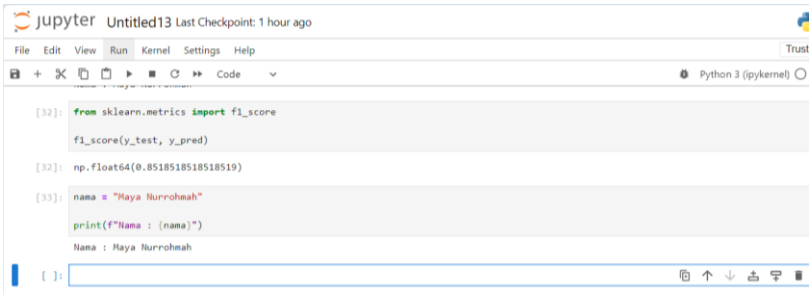
8) Pengenalan Accuracy Score



9) Pengenalan Precision dan Recall



10) Pengenalan F1 Score | F1 Measure



11) Pengenalan ROC | Receiver Operating Characteristic