

TUGAS PERTEMUAN 2

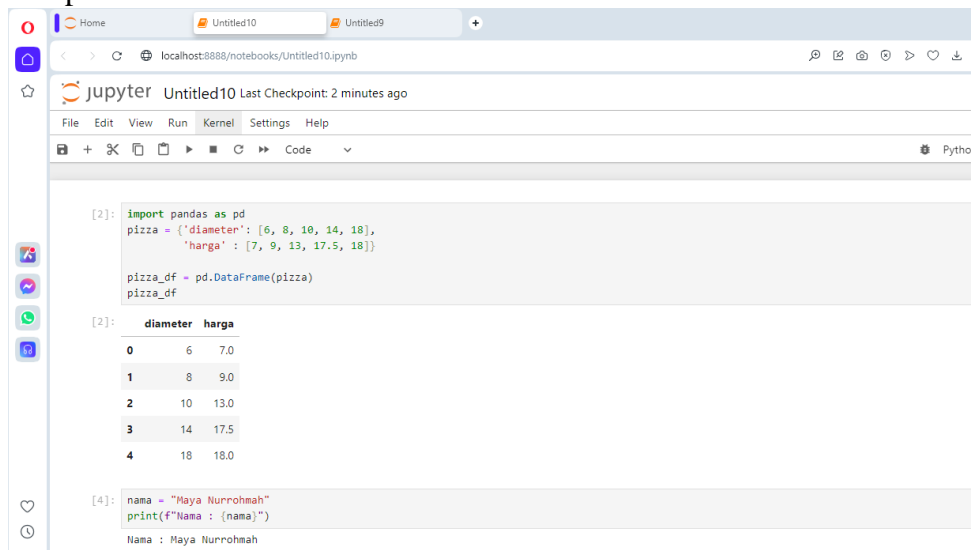
NAMA : MAYA NURROHMAH

NPM : 41155050210019

KELAS : INF_A1

1. Lakukan praktek dari <https://youtu.be/lcj7-2zMSA?si=f4jWJR6lY8y0BZKl> dan buat screen shot hasil run dengan nama anda pada hasil run tersebut. Praktek tersebut yaitu:

1) Sample dataset



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
[2]: import pandas as pd
pizza = {'diameter': [6, 8, 10, 14, 18],
        'harga': [7, 9, 13, 17.5, 18]}

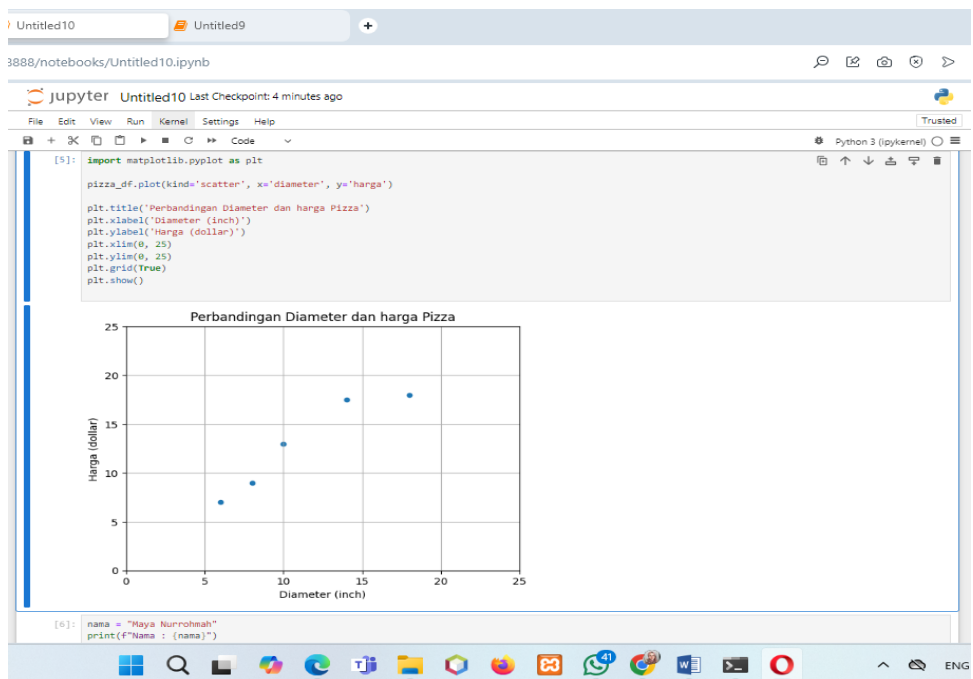
pizza_df = pd.DataFrame(pizza)
pizza_df
```

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

```
[4]: nama = "Maya Nurrohmah"
print(f>Nama : {nama}")

Nama : Maya Nurrohmah
```

2) Visualisasi dataset



3) Transformasi dataset

```
me Untitled9 +
localhost:8888/notebooks/Untitled9.ipynb
jupyter Untitled9 Last Checkpoint: 45 minutes ago
File Edit View Run Kernel Settings Help Trusted
Python 3 (ipykernel)
[7]: import numpy as np
X = np.array(pizza_df['diameter'])
y = np.array(pizza_df['harga'])
print(f'x: {X}')
print(f'y: {y}')

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

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

[8]: (5, 1)

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

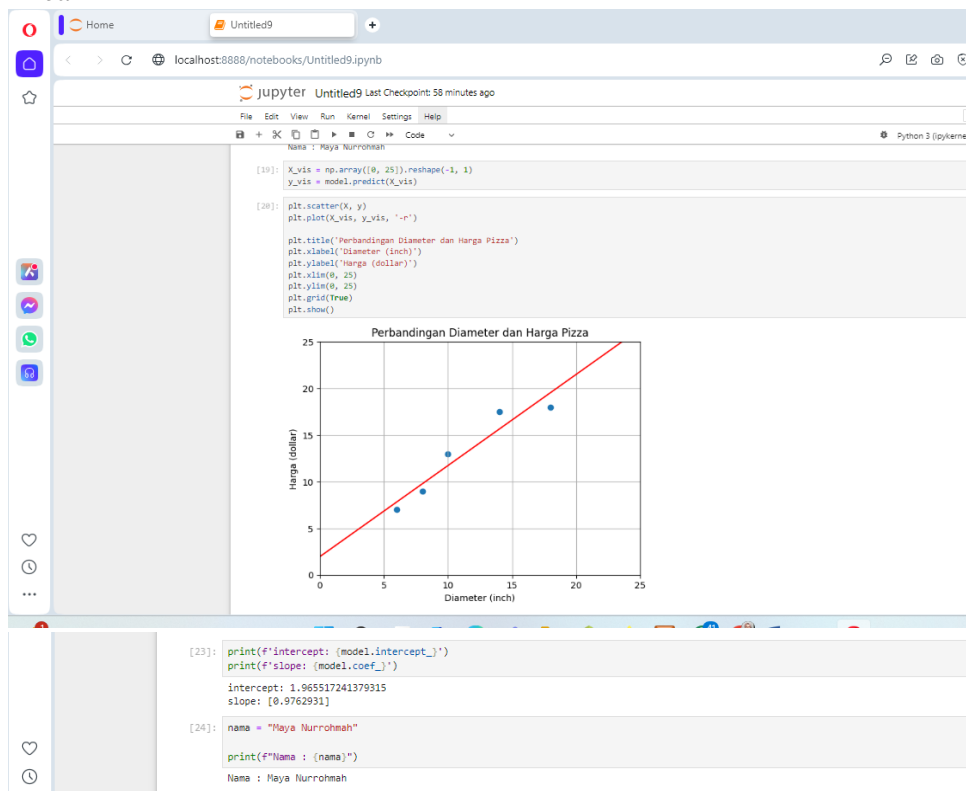
[14]: nama = "Maya Nurrohmah"
print(f>Nama : {nama}")
Nama : Maya Nurrohmah
```

4) Training simple linear regression model

```
[ ]: from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X, y)

[15]: nama = "Maya Nurrohmah"
print(f>Nama : {nama}")
Nama : Maya Nurrohmah
```

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



6) Kalkulasi nilai slope

```
me Untitled9 +
localhost:8888/notebooks/Untitled9.ipynb
jupyter Untitled9 Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help Trusted
Python 3 (ipykernel)

[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])

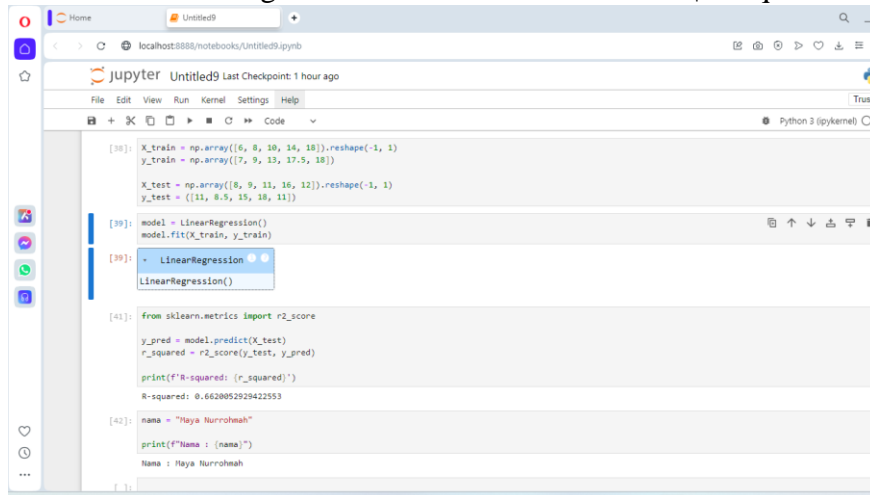
[35]: for dmtr, hrg in zip(diameter_pizza, prediksi_harga):
      print(f'Diameter: {dmtr} prediksi harga: {hrg}')

      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



```
[38]: 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 = [[11, 8.5, 15, 18, 11]]

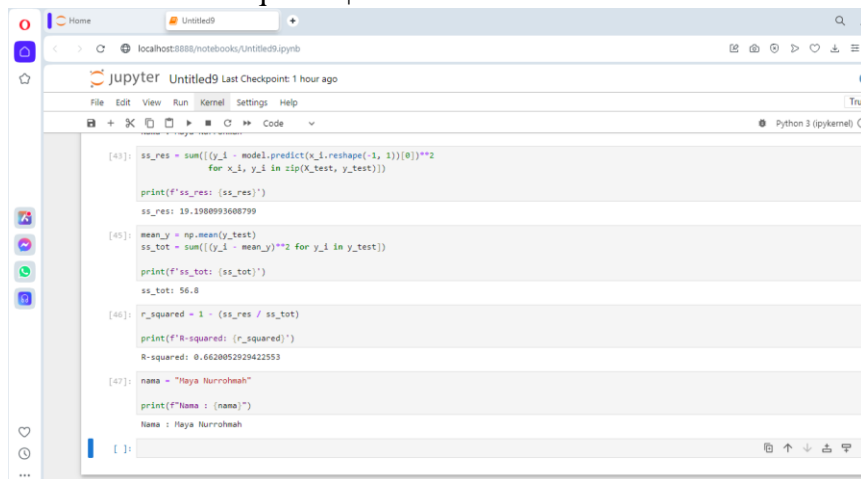
[39]: model = LinearRegression()
      model.fit(X_train, y_train)

[39]: LinearRegression
      LinearRegression()

[41]: 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: 0.6620052920422553

[42]: nama = "Haya Nurrohmah"
      print(f'Nama : {nama}')
      Nama : Haya Nurrohmah
```

10) Kalkulasi nilai R Squared | Coefficient of Determination



```
[43]: 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}')
      ss_res: 19.1908993608799

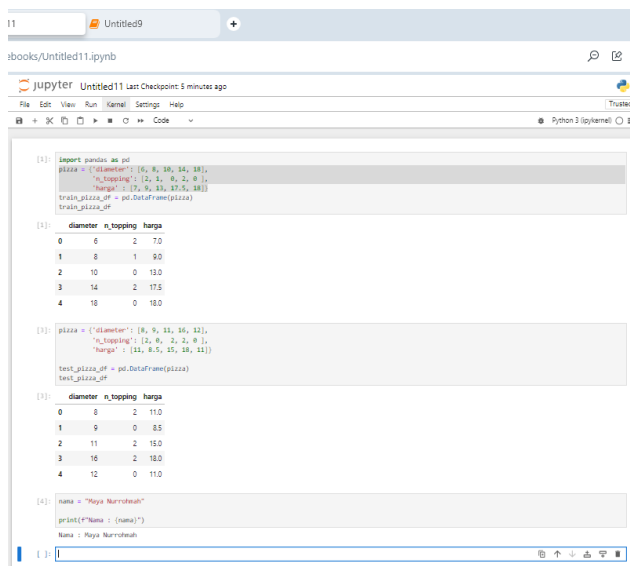
[45]: mean_y = np.mean(y_test)
      ss_tot = sum([(y_i - mean_y)**2 for y_i in y_test])
      print(f'ss_tot: {ss_tot}')
      ss_tot: 56.8

[46]: r_squared = 1 - (ss_res / ss_tot)
      print(f'R-squared: {r_squared}')
      R-squared: 0.6620052920422553

[47]: nama = "Haya Nurrohmah"
      print(f'Nama : {nama}')
      Nama : Haya Nurrohmah
```

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



```
[1]: 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

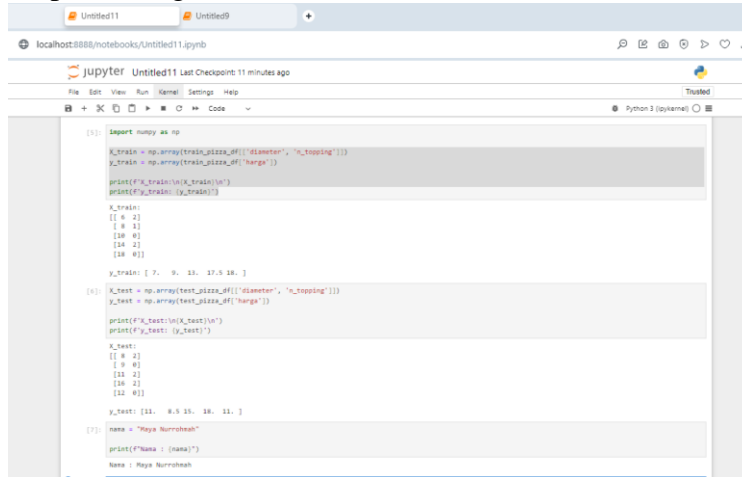
[1]: diameter  n_topping  harga
0         6         2     7.0
1         8         1     9.0
2        10         0    13.0
3        14         2    17.5
4        18         0    18.0

[2]: pizza = {'diameter': [8, 9, 11, 16, 12],
              'n_topping': [2, 0, 2, 2, 0],
              'harga': [11, 8.5, 15, 18, 11]}
      test_pizza_df = pd.DataFrame(pizza)
      test_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

[4]: nama = "Haya Nurrohmah"
      print(f'Nama : {nama}')
      Nama : Haya Nurrohmah
```

2) Preprocessing dataset



A Jupyter Notebook interface with two tabs: 'Untitled11' and 'Untitled9'. The 'Untitled11' tab is active, showing a Python script for data preprocessing. The script imports numpy and uses it to load and preprocess data from 'train_pizza_df' and 'test_pizza_df'. It prints out the dimensions and values of the resulting X_train, y_train, X_test, and y_test arrays.

```
[3]: 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:\n{y_train}\n')

X_train:
[[ 4  2]
 [ 8  3]
 [10  4]
 [14  2]
 [18  0]]

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

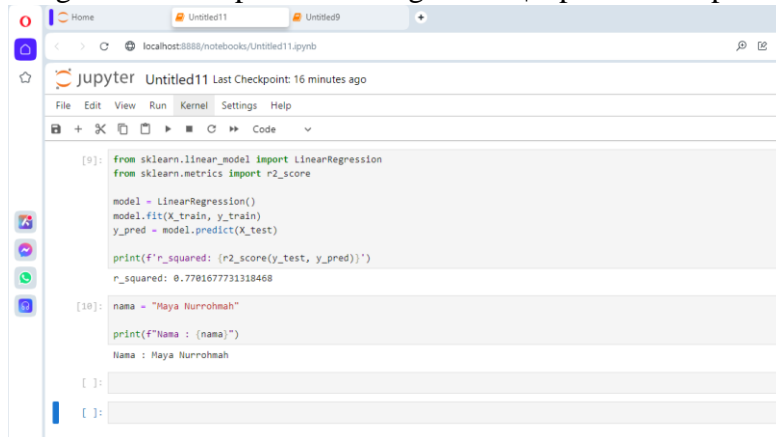
[4]: X_test = np.array(test_pizza_df[['diameter', 'n_topping']])
y_test = np.array(test_pizza_df['harga'])
print(f'X_test:\n{X_test}\n')
print(f'y_test:\n{y_test}\n')

X_test:
[[ 4  2]
 [ 9  4]
 [13  2]
 [16  2]
 [22  0]]

y_test: [13.  8.5 15. 18. 11. ]

[7]: nama = "Maya Nurrohmah"
print(f'Nama : {nama}')
Nama : Maya Nurrohmah
```

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



A Jupyter Notebook interface with two tabs: 'Untitled11' and 'Untitled9'. The 'Untitled11' tab is active, showing a Python script for training and using a Multiple Linear Regression model from sklearn. The script imports LinearRegression and r2_score, fits the model on training data, and prints the r-squared value. It also includes a print statement for the user's name.

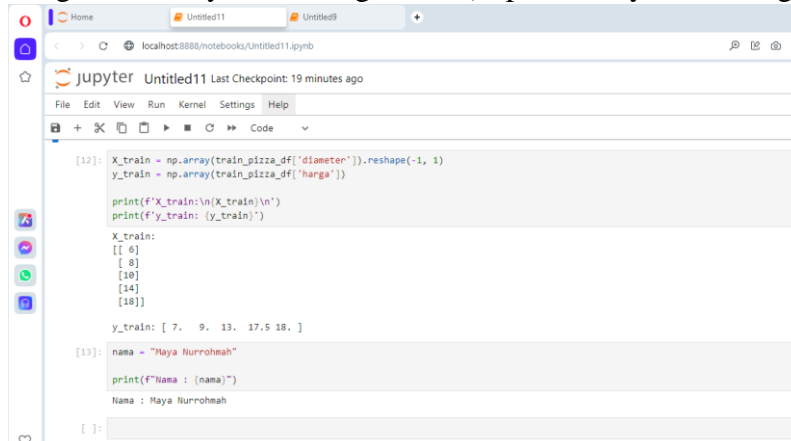
```
[9]: from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

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

print(f'r_squared: {r2_score(y_test, y_pred)}')
r_squared: 0.7781677731318468

[10]: nama = "Maya Nurrohmah"
print(f'Nama : {nama}')
Nama : Maya Nurrohmah
```

4) Pengenalan Polynomial Regression | Apa itu Polynomial Regression?



A Jupyter Notebook interface with two tabs: 'Untitled11' and 'Untitled9'. The 'Untitled11' tab is active, showing a Python script for training and using a Polynomial Regression model. The script uses np.array to load training data, reshapes it, and prints its dimensions. It also includes a print statement for the user's name.

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

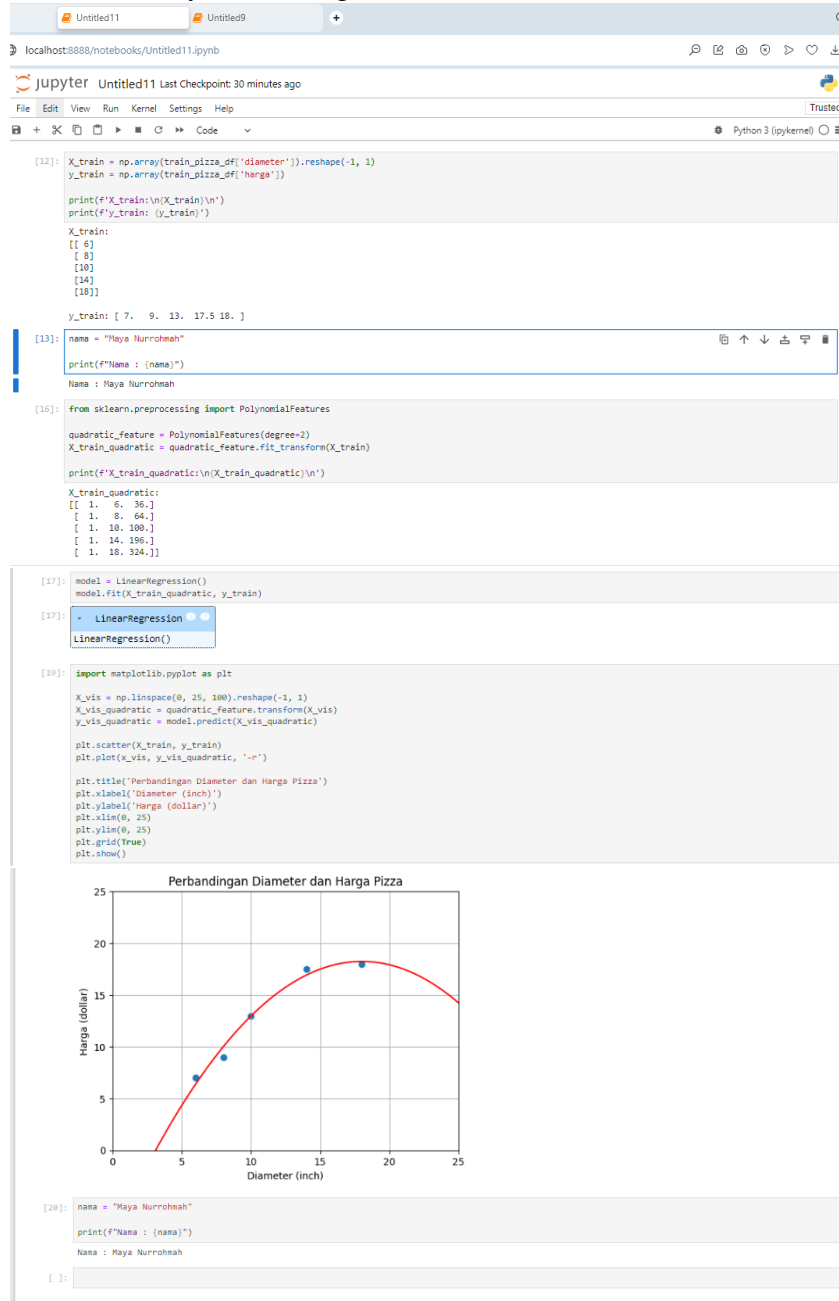
print(f'X_train:\n{X_train}\n')
print(f'y_train:\n{y_train}\n')

X_train:
[[ 4]
 [ 8]
 [10]
 [14]
 [18]]

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

[13]: nama = "Maya Nurrohmah"
print(f'Nama : {nama}')
Nama : Maya Nurrohmah
```

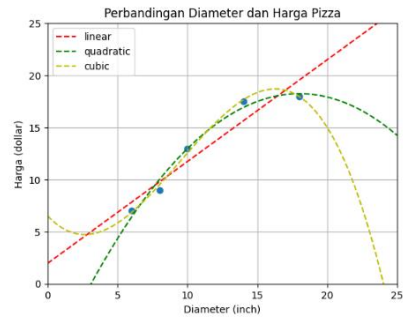
5) Quadratic Polynomial Regression



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



```
plt.ylabel('harga (dolar)')
plt.xlim(0, 25)
plt.ylim(0, 25)
plt.grid(True)
plt.legend()
plt.show()
```

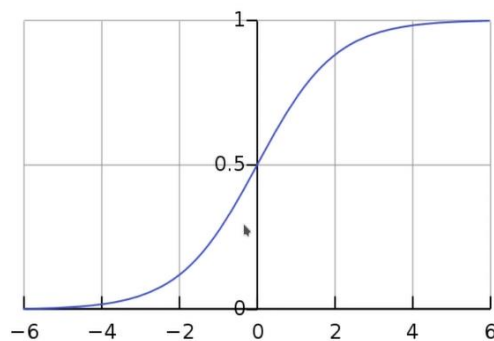


```
[28]: nama = "Maya Nurrohaah"
print(f'Nama : {nama}')
Nama : Maya Nurrohaah
```

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
 $y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$
 $Xg(X) = \alpha + \beta X$
- 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

```
Home Untitled13
localhost:8888/notebooks/Untitled13.ipynb
jupyter Untitled13 Last Checkpoint: 4 minutes ago
File Edit View Run Kernel Settings Help Python 3 (ipykernel) Trusted

[2]: import pandas as pd

df = pd.read_csv("../dataset/SMS Spam Collection",
                 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}")

Name : Maya Nurrohmah
```

3) Pembagian training dan testing set

```
jupyter Untitled13 Last Checkpoint: 11 minutes ago
File Edit View Run Kernel Settings Help Python 3 (ipykernel) Trusted

[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=object)

[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 0822300901']
'Ok...'
'For ur chance to win a £250 cash every wk TXT: ACTION to 80608. T's&C's www.movietrivia.tv custcare 08712409022, 1x150p/wk'
'R U 854H P IN EACHOTHER. IF HE MEET ME CAN GO 2 MY HOUSE'
'I'm feeling sleepy. today itself i shall get that dear']

[8]: nama = "Maya Nurrohmah"
print(f>Nama : {nama}")

Name : Maya Nurrohmah
```

4) Feature extraction dengan TF-IDF

```
jupyter Untitled13 Last Checkpoint: 48 minutes ago
File Edit View Run Kernel Settings Help Python 3 (ipykernel) Trusted

[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)) 0.308574289326673
(np.int32(0), np.int32(4453)) 0.1297719964323795
(np.int32(0), np.int32(3926)) 0.3126721340000456
(np.int32(0), np.int32(2554)) 0.3825278811525034
(np.int32(0), np.int32(6738)) 0.3546355942830148
(np.int32(0), np.int32(900)) 0.4114867709157148
(np.int32(0), np.int32(2006)) 0.2898082580285881
(np.int32(0), np.int32(6903)) 0.359138442223876
(np.int32(1), np.int32(5642)) 0.24344998442301355
(np.int32(1), np.int32(799)) 0.25048918791028574
(np.int32(1), np.int32(5442)) 0.5005783758205715
(np.int32(1), np.int32(6472)) 0.24039776602646594
(np.int32(1), np.int32(6013)) 0.20089911182610476
(np.int32(1), np.int32(216)) 0.28902673048308515
(np.int32(1), np.int32(4677)) 0.24039776602646594
(np.int32(1), np.int32(5394)) 0.16464655071448758
(np.int32(1), np.int32(6131)) 0.16142698935984466
(np.int32(1), np.int32(5121)) 0.30186802351390565
(np.int32(1), np.int32(4358)) 0.1734148292348694
(np.int32(1), np.int32(5301)) 0.2711077535907125
(np.int32(1), np.int32(2003)) 0.2711077535907125
(np.int32(1), np.int32(1548)) 0.18167737976542422
(np.int32(1), np.int32(36)) 0.28902673048308515
```



```

(np.int32(1), np.int32(1548)) 0.18167737976542422
(np.int32(1), np.int32(36)) 0.28902673040368515
:
:
(np.int32(4176), np.int32(6792)) 0.140760461250961
(np.int32(4176), np.int32(6693)) 0.16491299289150899
(np.int32(4176), np.int32(6684)) 0.22114159453000114
(np.int32(4176), np.int32(7083)) 0.19523751585154273
(np.int32(4176), np.int32(1509)) 0.18095085073406012
(np.int32(4176), np.int32(7195)) 0.17892283441772988
(np.int32(4176), np.int32(779)) 0.2811068572055718
(np.int32(4176), np.int32(1612)) 0.21138425595332702
(np.int32(4176), np.int32(365)) 0.2380005587702937
(np.int32(4176), np.int32(7114)) 0.4512018097459442
(np.int32(4176), np.int32(637)) 0.2996866840649284
(np.int32(4176), np.int32(4350)) 0.2996866840649284
(np.int32(4176), np.int32(2004)) 0.25589560236817055
(np.int32(4176), np.int32(107)) 0.2996866840649284
(np.int32(4176), np.int32(343)) 0.2811068572055718
(np.int32(4177), np.int32(3319)) 0.43046342221720785
(np.int32(4177), np.int32(4177)) 0.3636187667918345
(np.int32(4177), np.int32(5565)) 0.5580606649743346
(np.int32(4177), np.int32(2362)) 0.615885485899457
(np.int32(4178), np.int32(2068)) 0.3055766821331892
(np.int32(4178), np.int32(2641)) 0.3993042639531407
(np.int32(4178), np.int32(6555)) 0.2897850627168302
(np.int32(4178), np.int32(5720)) 0.3963527249882828
(np.int32(4178), np.int32(4279)) 0.4530624713751054
(np.int32(4178), np.int32(5883)) 0.548491137555895

[12]: nama = "Maya Nurrohmah"
print(f"Nama : {nama}")
Nama : Maya Nurrohmah

```

5) Binary Classification dengan Logistic Regression

```

[15]: 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")

PRED: 0 - SMS: Storming msg: Men u lift d phne, u say "HELLO" Do u knw ut is d real meaning of HELLO?? . . . It's d name of a girl! . . . Yes.. And u k
nu who is dat girl? "Margret Hello" She is d girlfrnd f Brahmabell who invnted telphone... . . . Horal: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..

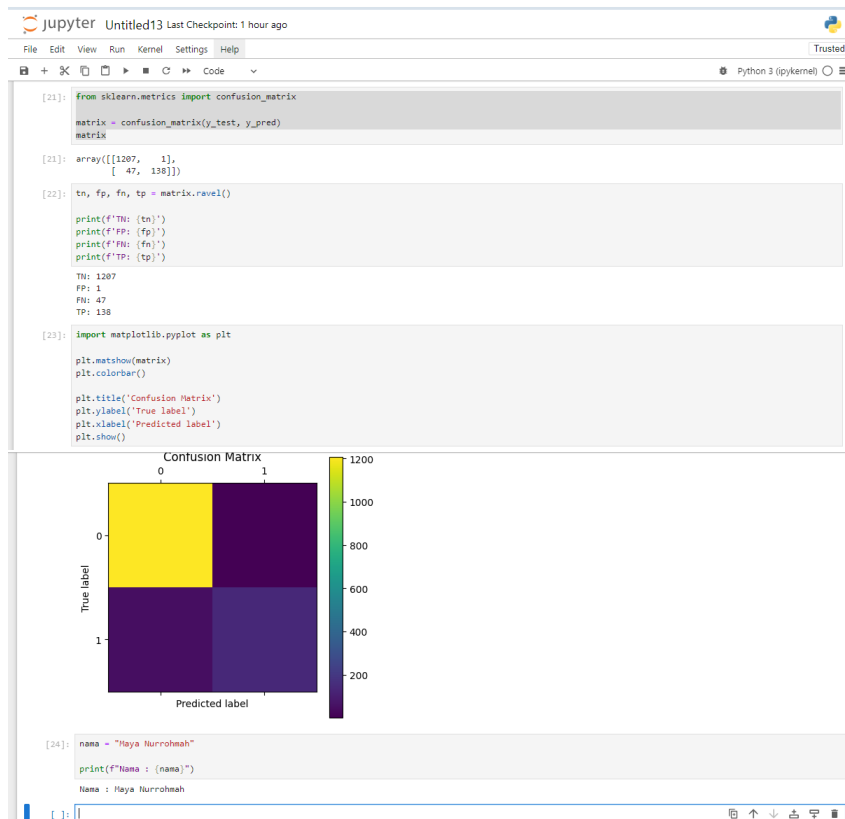
[16]: nama = "Maya Nurrohmah"
print(f"Nama : {nama}")
Nama : Maya Nurrohmah

```

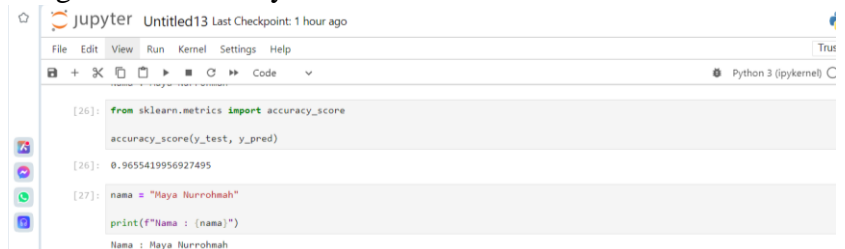
6) Evaluation Metrics pada Binary Classification Task

- Confusion Matrix
- Accuracy
- Precission & Recall
- F1 Score
- ROC
- Terminologi Dasar
- True Positive (TP)
- True Negative (TN)
- False Positive (FP)
- False Negative (FN)

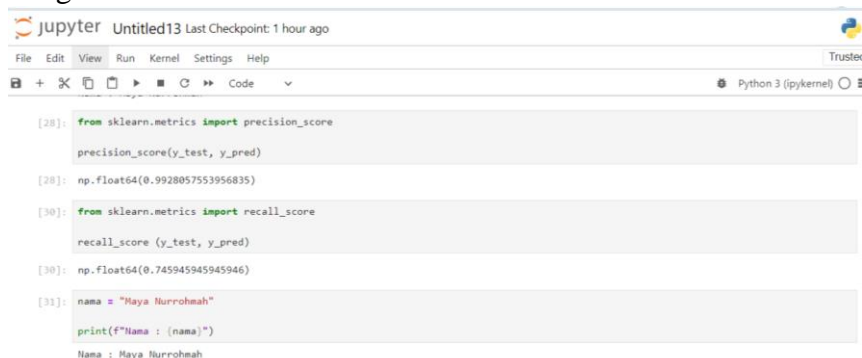
7) Pengenalan Confusion Matrix



8) Pengenalan Accuracy Score



9) Pengenalan Precision dan Recall



10) Pengenalan F1 Score | F1 Measure

```
jupyter Untitled13 Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Python 3 (pykernel)

[32]: from sklearn.metrics import f1_score
      f1_score(y_test, y_pred)

[32]: np.float64(0.8518518518518519)

[33]: nama = "Haya Nurrohmah"
      print(f"Nama : {nama}")

Nama : Haya Nurrohmah

[ ]:
```

11) Pengenalan ROC | Receiver Operating Characteristic

