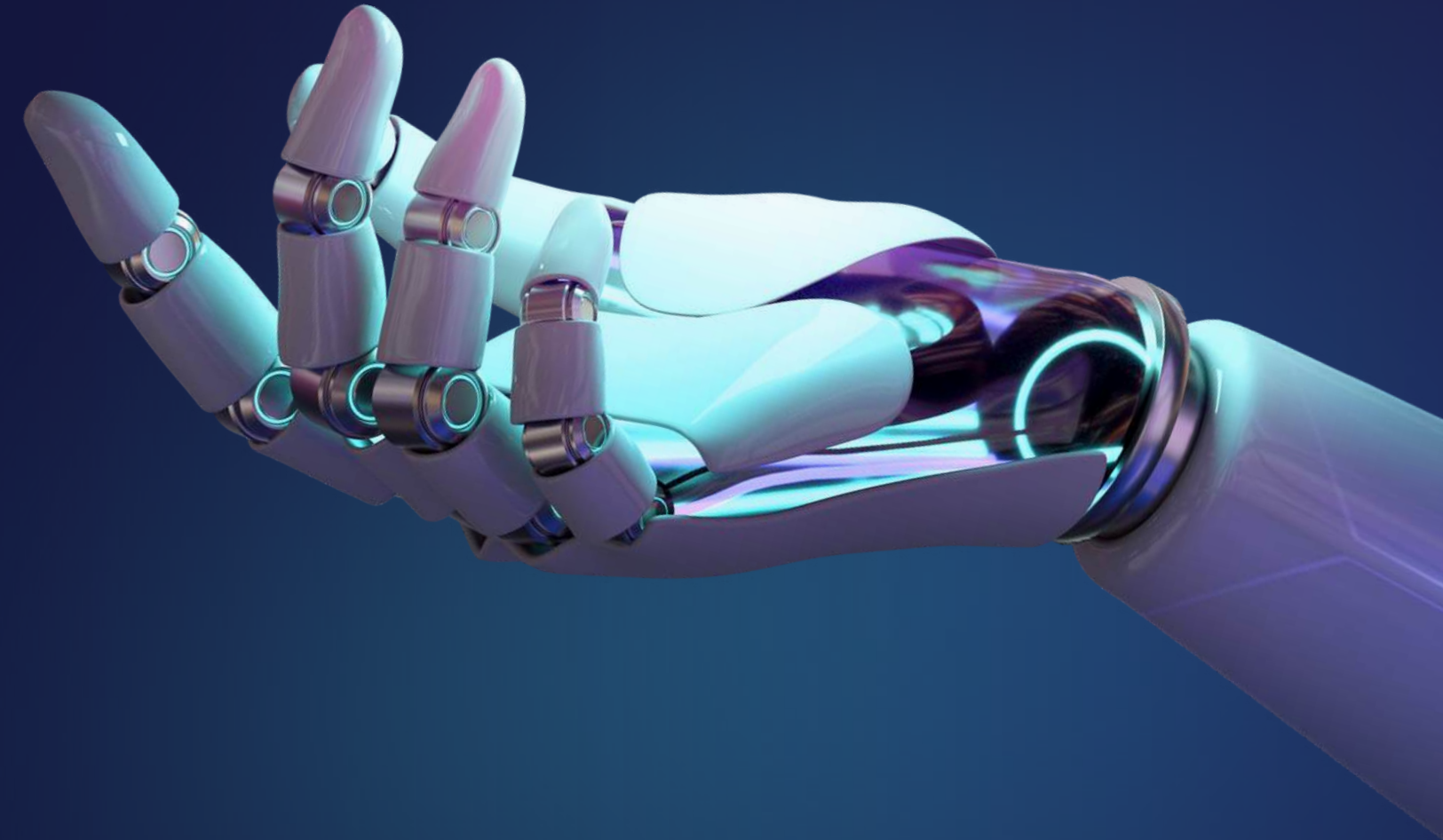


# PRESENTATION

## MACHINE LEARNING

Rizky Ramadhan

Read More



# MACHINE LEARNING

Machine Learning adalah cara komputer mempelajari data untuk membuat prediski atau keputusan yang berguna untuk manusia. ini berguna dalam berbagai aplikasi, penegnanan wajah prediksi harga saham prediksi chum, prediksi harga



# LIBRARIES

Libraries ini untuk mengelola data pada saat code live berlangsung


```
[1] # Import libraries and resources
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
```

ini adalah code buat membaca data yang telah di upload oleh kita di google colabs dan juga untuk menyimpan variable data

```
# read the dataset using pandas  
data = pd.read_excel('student_scores.xlsx', engine='openpyxl') # Use pd.read_excel with the 'openpyxl' engine
```

CODE ini berguna untuk magambil data di baris 1 sampai dari colum data yang kita upload horurs yaitu x nya lalu Scoes y nah kita akan memprediksi scores dalam jarak waktu

```
# This display the top 7 rows of the data  
data.head(7)
```



	Hours (x)	Scores (y)
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88

code disini untuk menagalisis data  
explanatory

```
[ ] # Provides some informaion regarding the columns in the data
data.info()
```

```
↔ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
#   column      Non-Null Count  Dtype
---  -
0   Hours (x)    25 non-null         float64
1   Scores (y)   25 non-null         int64
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

code ini adalah describe untuk medapat informasi secara statistiknya

```
[ ] # this describe the basic stat behind the dataset used  
data.describe()
```



	Hours (x)	Scores (y)
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

code bertujuan memvisualisasikan data dengan menggunakan grafik scatter

[Read More](#)

```
[ ] # These Plots help explain the values and how they are scattered

plt.figure(figsize=(12,6))
sns.pairplot(data,x_vars=['Hours (x)'],y_vars=['Scores (y)'],size=7,kind='scatter')
plt.xlabel('Hours (x)')
plt.ylabel('Scores (y)')
plt.title('Scores Prediction')
plt.show()
```



# FEATURE ENGINEERING

Code ini untuk memeriksa data apakah ada duplicate atau tidak karena di machine learning tidak bisa menolak data yang bersifat duplicate

```
[ ] df = data.copy()

[ ] #Check Duplicated Data
    print("DataSebelum Pemeriksaan Duplikat:")
    print(df.shape)

↔ DataSebelum Pemeriksaan Duplikat:
   (25, 2)

[ ] duplicate_rows_before = df[df.duplicated()]
    duplicate_rows_before

↔
```

Hours (x)	Scores (y)
-----------	------------

```
[ ] df = df.drop_duplicates()

[ ] print("\nData Setelah Pemeriksaan Duplikat;")
    print(df.shape)

↔ Data Setelah Pemeriksaan Duplikat;
   (25, 2)
```

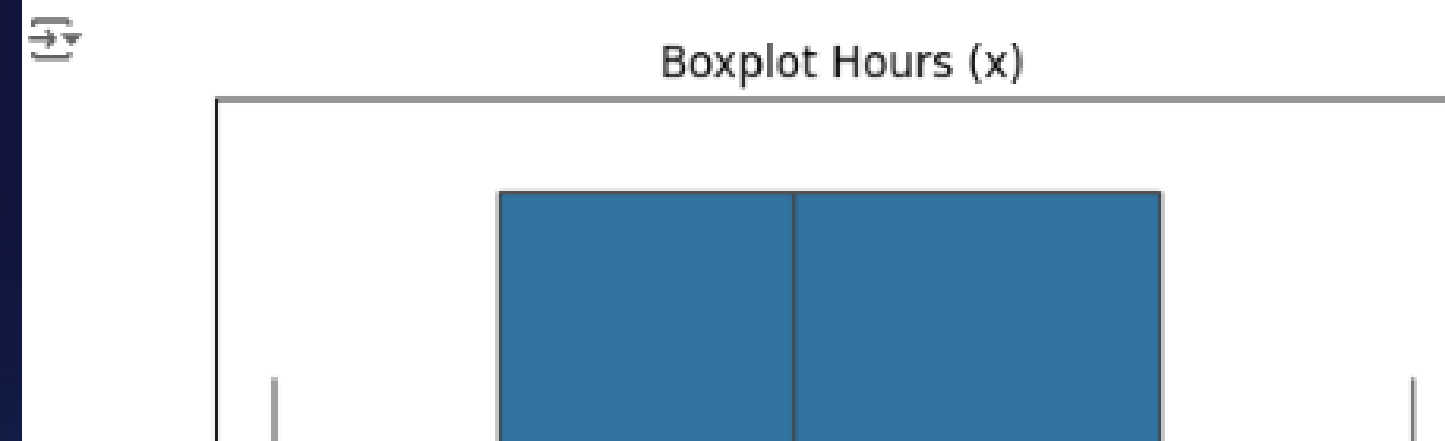
Code disini berfungsi untuk mengecek ada outlier atau tidak dengan menggunakan libraries seabron dan menggunakan boxplot

```
[ ] import seaborn as sns
import matplotlib.pyplot as plt

# Outlier Analysis
sns.boxplot(x="Hours (x)", data=df)

# Menambahkan label sumbu dan judul
plt.xlabel("Hours (x)")
plt.ylabel("Frequency") # Anda bisa menggunakan "Hours (x)" atau "Jam"
plt.title("Boxplot Hours (x)")

# Menampilkan plot
plt.show()
```



Code split ini berfungsi untuk membagi data

```
[ ] # Import machine learning data from scikit learn
    from sklearn.model_selection import train_test_split

# Split the data for train and test
# train : test = 75 ; 25 atau 80 : 20 atau 70:30 atau 85:15
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.75, random_state=42)
```

Cde ini berfungsi fitting data train untuk dan juga proses belajar di Linear Regression setelah belajar lalu code dibawah berfungsi untuk memprediksi

```
[ ] # Fitting teh model using linear Regression
lr_model = LinearRegression() # Mahasiswa A
lr_model.fit(x_train,y_train) # Belajar
```



LinearRegression ⓘ ⓘ  
LinearRegression()

```
[ ] y_pred_train = lr_model.predict(x_train)
y_pred_train
```



```
array([52.02223727, 39.39391742, 17.05150538, 33.56546211, 46.19378196,
       88.93578761, 85.05015073, 36.47968977, 28.70841601, 20.93714226,
       49.10800962, 61.73632947, 78.25028619, 55.90787415, 77.27887697,
       13.1658685 , 74.36464932, 91.85001526])
```

Cde ini berfungsi mengevaluasi secara kualitatif dan memvisualisasikan dengan grafik

```
[ ] # Plotting the actual and predicted values

c = [i for i in range (1,len(y_test)+1,1)]
plt.plot(c,y_test,color='r',linestyle='-',label='Actual Data')
plt.plot(c,y_pred,color='b',linestyle='dashed',label='Prediction')
plt.xlabel('Scores (x)')
plt.ylabel('index')
plt.title('Prediction with Linear Regression')
plt.legend()
plt.show()
```

Cde ini berfungsi untuk mengambil nilai coefisienya

```
[ ] # Intecept and coeff the line  
    print('Intercept of the Linear Regression model:',lr_model.intercept_)  
    print('coefficient of the line Linear Regression:', lr_model.coef_)
```

```
⇒ Intercept of the Linear Regression model: 2.4803670915057623  
   coefficient of the line Linear Regression: [9.71409219]
```

Then is said to from a line with result in Linear Regression

$$y = 2.4803 + 9.7140 x$$

Code ini berfungsi sama seperti linear regression untuk proses belajar mahasiswa tapi ini menggunakan decisio tree

```
[ ] from sklearn.tree import DecisionTreeRegressor # mahasiswa
```

```
[ ] dt_model = DecisionTreeRegressor()  
dt_model.fit(x_train, y_train)
```



DecisionTreeRegressor

DecisionTreeRegressor()

```
[ ] # Predicting the Scores (y) for the Test values  
y_pred_dt = dt_model.predict(x_test) # soal uts maupun uas
```