



Ghibran Muhammad - 08

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI**

LAPORAN TUGAS MINGGU 3

1. Gambar di bawah ini merupakan pemanggilan library di python

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

[1] ✓ 4.1s

2. library dari panda untuk membaca file “50_startups.csv” lalu ada kode program “data_awal.head()” digunakan untuk menampilkan 5 file teratas

```
data_awal = pd.read_csv('50_Startups.csv')
data_awal.head()
```

✓ 0.1s

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

3. dibawah ini merupakan proses dari OneHotEncoder

```
transformer_list = [
    ('encoded', OneHotEncoder(dtype='int'), ['State']),
    ('skip', 'passthrough', ["R&D Spend", "Administration", "Marketing Spend", "Profit"])
]
```

✓ 0.8s

- Setelah dilakukan proses OneHotEncoder maka dilakukan proses column transformer untuk mengubah isi dari column state

```
coltran = ColumnTransformer(transformer_list)
tran = coltran.fit_transform(data_awal)
```

✓ 0.9s

- menggabungkan isian dari column state yang sudah dipecahmenadi satu dengan file "50_startups.csv"

```
data_baru = pd.DataFrame(tran, columns=coltran.get_feature_names())
data_baru.head()
```

✓ 0.1s

c:\Users\acer\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function get_feature_names is deprecated in 1.2. Please use get_feature_names_out instead.
warnings.warn(msg, category=FutureWarning)

	encoded_x0_California	encoded_x0_Florida	encoded_x0_New York	R&D Spend	Administration	Marketing Spend	Profit
0	0.0	0.0	1.0	165349.20	136897.80	471784.10	192261.83
1	1.0	0.0	0.0	162597.70	151377.59	443898.53	191792.06
2	0.0	1.0	0.0	153441.51	101145.55	407934.54	191050.39
3	0.0	0.0	1.0	144372.41	118671.85	383199.62	182901.99
4	0.0	1.0	0.0	142107.34	91391.77	366168.42	166187.94

- Proses selanjutnya proses Linear Regeresion, pada bagian tahap pertama dibuat variable X dan y, X digunakan untuk mengambil semua column dari table yang sudah di lakukan proses Column Transformer

```
x = data_baru.iloc[:, :-1].values
y = data_baru.iloc[:, 6].values
```

✓ 0.1s

```
y = y.reshape(len(y), 1)
y.shape
```

✓ 0.7s

(50, 1)

7. proses selanjutnya yaitu proses pemanggilan library LinearRegression, pada proses ini dibuat 4 variable dan dilakukan proses train_test_split yang isinya terdapat variable X, dan y

```
Xm_train, Xm_test, ym_train, ym_test = train_test_split(X, y, test_size=0.2, random_state=50)
✓ 0.8s

mlr = LinearRegression()
mlr.fit(Xm_train, ym_train)

ym_pred = mlr.predict(Xm_test)
✓ 0.2s
```

8. selanjutnya yaitu penggabungan dari Linear Regression

```
gabung = np.concatenate((ym_test, ym_pred), axis=1)
gabung
✓ 0.8s

array([[ 96712.8      ,  84457.16115252],
       [ 90708.19     ,  72543.44544734],
       [191792.06     , 186070.97934498],
       [ 81229.06     ,  67431.79651296],
       [152211.77     , 150237.4771915 ],
       [126992.93     , 113635.43439889],
       [ 96479.51     ,  88703.00224459],
       [134307.35     , 124781.17620402],
       [101004.64     , 101217.77951759],
       [108552.04     , 113863.68825244]])
```

9. Selanjutnya dilakukan prediksi menggunakan r2_score, nilai yang sempurna adalah nilai yang mendekati 1

```
r2_mlr = r2_score(ym_test, ym_pred)
print(r2_mlr)
✓ 0.5s
```





10.10.