

**Laporan**  
**Final Project Data Science Club**  
**DSC**



**Disusun Oleh :**

**Moh. Subhan Fajarulloh                      (192400031)**

**PROGRAM STUDI STATISTIKA**  
**FAKULTAS SCIENCE DAN TEKNOLOGI (FST)**  
**UNIVERSITAS PGRI ADI BUANA SURABAYA**  
**2022**

## A. Statistika Deskriptif

Statistik Deskriptif juga merupakan metode yang sangat sederhana. Metode ini hanya mendeskripsikan kondisi dari data yang sudah anda miliki Dan menyajikannya dalam bentuk tabel diagram grafik dan bentuk lainnya yang disajikan dalam uraian – uraian singkat dan juga terbatas. Disini saya menggunakan data sekunder mengambil dari bps

## B. Data Preprocessing

### 1. Import Library

- **Import numpy as np**  
operasi komputasi tipe data numerik seperti penjumlahan, pengurangan, perkalian, pangkat, dan operasi lainnya yang bisa diterapkan pada vektor atau matriks
- **Import pandas as pd**  
Digunakan untuk membuat tabel, mengubah dimensi data, mengecek data, dan lain sebagainya. Struktur data dasar pada Pandas dinamakan DataFrame, yang memudahkan kita untuk membaca sebuah file dengan banyak jenis format seperti file .txt, .csv, dan .tsvd
- **Import statsmodels**  
Untuk mengeksplorasi data, memperkirakan model statistik, dan melakukan tes statistik.
- **Import patsy**  
untuk menggambarkan model statistik (terutama model linier, atau model yang memiliki komponen linier) dan desain matriks
- **Import statsmodels.api as sm**
- **Import matplotlib.pyplot as plt**  
Memvisualisasikan data

### 2. Import Model

```
from sklearn.linear_model import LinearRegression
from sklearn.cross_validation import train_test_split
from sklearn import metrics
```

### 3. Load dataset

Dataset yang digunakan yaitu ekspor.csv

```
df=pd.read_csv('dari 3 variabel.csv')
df
```

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

```
In [2]: #Import library
import pandas as pd
import numpy as np
import statsmodels
import patsy
import statsmodels.api as sm
import matplotlib.pyplot as plt

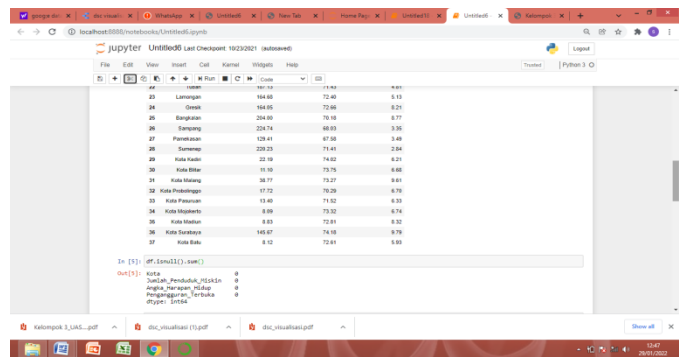
In [3]: from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics

In [8]: df=pd.read_csv("dari 3 variabel.csv")
df
```

The output of the last cell is a DataFrame with 8 rows and 4 columns:

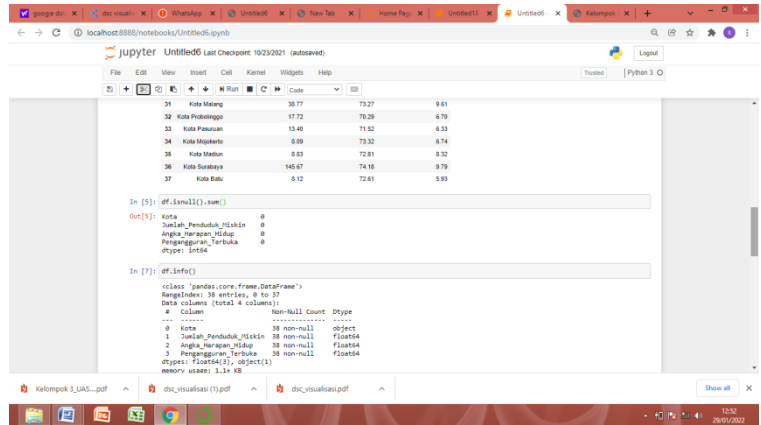
	Kota	Jumlah_Penduduk_Miskin	Angka_Harapan_Hidup	Pengangguran_Terbuka
0	Pacitan	80.82	71.94	2.28
1	Ponorogo	86.74	72.77	4.45
2	Trenggalek	81.06	73.75	4.11
3	Tulungagung	76.40	74.08	4.61
4	Blitar	106.55	73.52	3.82
5	Kediri	179.93	72.61	5.24
6	Malang	265.56	72.55	5.49
7	Lumajang	102.60	70.10	3.36

4. Mengatasi missing value pada data, untuk mengecek missing value pada data gunakan syntax `df.isnull().sum()`, `df` yaitu variabel yang menyimpan data Kelompok 3\_UAS Komstat\_Dataset (dari 3 variabel).csv, `isnull()` untuk mengecek apakah dataframe ada nilai null dan untuk `sum()` menjumlahkan dataframe yang tidak null



Output diatas menandakan tidak ada missing value pada ketiga variabel karena nilainya nol/tidak ada missing value, maka tidak diperlukan pengisian missing value

5. Melihat informasi pada data Kelompok 3, menggunakan fungsi `data.info()`. Untuk datanya menggunakan variabel `df` dan fungsi dari `info()` yaitu untuk melihat informasi pada data



Hasil ouput diatas tidak ada Dtype yang perlu diganti. Jika variabel tersebut Data typenya tidak cocok, maka dilakukan penggantian Dtype untuk contoh syntax `df['Kota']=df['Kota'].astype(Dtype)`, Dtype yaitu Dtype baru

6. Pilih Jumlah\_Penduduk\_Miskin dan Pengangguran\_Terbuka untuk variabel X dan Angka\_Harapan\_Hidup sebagai variabel Y

`feature_names=['Jumlah_Penduduk_Miskin','Pengangguran_Terbuka']`

`x=df[feature_names]`

`x`

`y=df.Angka_Harapan_Hidup`

7. Kemudian memisahkan X dan y ke dalam data latih (train) dan data pengujian (test):

`x_train, x_test, y_train, y_test = train_test_split(x,y,random_state=1)`

mengimpor library sklearn untuk memisahkan menjadi test set dan train set, kemudian dapat menggunakan fungsi “train\_test\_split” untuk mendefinisikan data X dan Y nya. Misalnya data X adalah semua kolom kecuali kolom paling ujung kanan dan data Y adalah kolom paling ujung kanan.

`x_train` : untuk menampung data X yang akan ditraining

X\_test : untuk menampung data X yang akan ditesing

Y\_train: untuk menampung data Y yang akan ditraining

Y\_test : untuk menampung data Y yang akan ditesing

X dan y adalah nama variabel yang digunakan saat mendefinisikan data X dan Y.

Atribut “random\_state” untuk menghindari pembagian train set dan test set jika mengulang proses running hasilnya akan berubah-ubah. Angka yang didefinisikan random\_state bisa angka berapa saja yang berupa integer.

## 8. Ketik linear Regression Model

**Linreg=LinearRegression()**, class LiniearRegression berasal dari library sklearn.linear\_model. **LiniearRegression()** adalah fungsi untuk mengimplementasikan algoritma linear regresi di python

## 9. Membuat model dengan data train

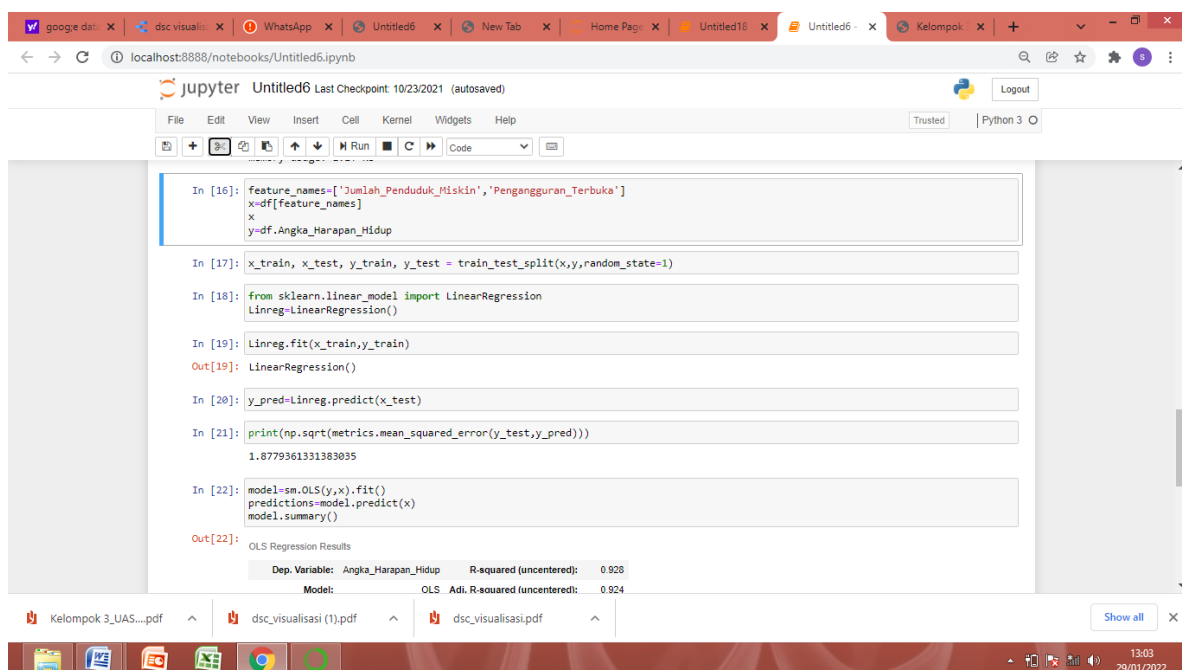
**Linreg.fit(x\_train,y\_train)**

## 10. Membuat prediksi pada data pengujian

**y\_pred=linreg.predict(x\_test)**

## 11. menghitung RMSE

**print(np.sqrt(metrics.mean\_squared\_error(y\_test,y\_pred)))**



```
In [16]: feature_names=['Jumlah_Penduduk_Miskin','Pengangguran_Terbuka']
x=df[feature_names]
y=df.Angka_Harapan_Hidup

In [17]: x_train, x_test, y_train, y_test = train_test_split(x,y,random_state=1)

In [18]: from sklearn.linear_model import LinearRegression
linreg=LinearRegression()

In [19]: Linreg.fit(x_train,y_train)
Out[19]: LinearRegression()

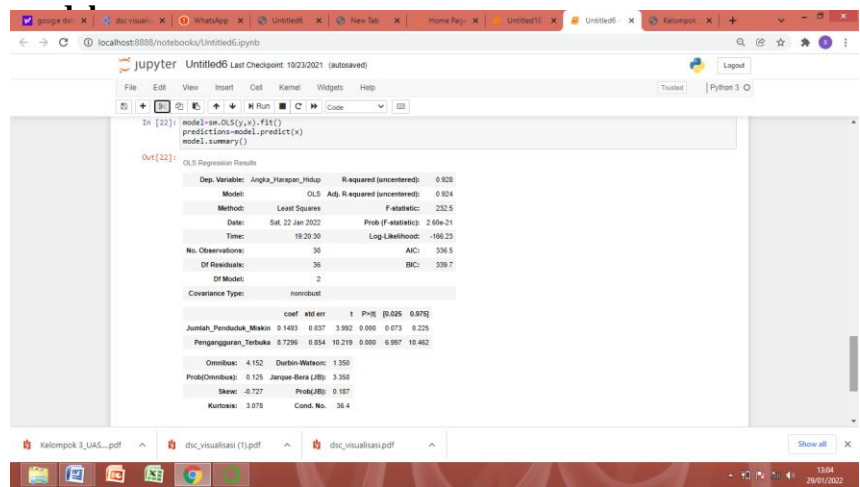
In [20]: y_pred=Linreg.predict(x_test)

In [21]: print(np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
1.8779361331383035

In [22]: model=sm.OLS(y,x).fit()
predictions=model.predict(x)
model.summary()
Out[22]:
```

OLS Regression Results	
Dep. Variable:	Angka_Harapan_Hidup
R-squared (uncentered):	0.928
Model:	OLS
Adj. R-squared (uncentered):	0.924

12. mencari model regresi
- model=sm.OLS(y,x).fit()**
- predictions=model.predict(x)**



C. Modelling

Regresi linier berganda sering diartikan sebagai salah satu alat analisis hubungan antar variabel dalam suatu penelitian. Analisis regresi linear berganda diharuskan melibatkan lebih dari satu variabel. Variabel tersebut terdiri dari variabel independen (bebas) dan variabel dependen (terikat). Jika Anda memiliki lebih dari dua variabel independen dalam satu data, maka regresi linear berganda yang digunakan untuk menganalisis. Dengan demikian, Anda dapat mengetahui kemungkinan seberapa besar variabel dependen yang ada dalam data.

Terdapat perbedaan antara regresi linear sederhana dan berganda. Jika variabel bebas berjumlah satu, maka itu disebut regresi linear sederhana. Sedangkan bila jumlah variabel lebih dari satu, entah dua atau lebih, akan dikatakan sebagai regresi linear berganda. Begitulah yang dikatakan teori regresi linear berganda.

Uji Analisis Regresi Linier Berganda

Selanjutnya, kita akan mencari dan menguji regresi linier berganda dari data yang sudah memenuhi uji klasik regresi linier berganda. Adapun rumus regresi linier berganda sebagai berikut:

**Y = α + β1 X2 + β2 X2 + βn Xn + e**

Y = Variabel terikat atau dependen

X = Variabel bebas atau independen

α = Konstanta.

β = Koefisien estimate.

Untuk meningkatkan pemahaman bersama, mari kita melihat soal regresi linier berganda dan jawabannya.

#### **D. Kesimpulan**

Statistika deskriptif merupakan metode yang mendeskriptifkan kondisi dari data yang sudah anda miliki Dan menyajikannya dalam bentuk tabel diagram grafik dan bentuk lainnya yang disajikan dalam uraian – uraian singkat dan juga terbatas. Data prosesing merupakan proses mengolah data. Dalam hal ini terdapat 12 langkah untuk mengerjakan pemrosesan data di Python menggunakan Regresi linear berganda. Regresi liner berganda sering diartikan sebagai salah satu alat analisis hubungan antar variabel dalam suatu penelitian