# UNIT 12

# pandas dan seaborn

## Tujuan:

### Mahasiswa dapat melakukan data visualisasi pada Python.

### Mahasiswa dapat melakukan manajemen serta memasukkana data dengan menggunakan Pandas.

### Mahasiswa dapat melakukan proses import library pada Python

### Mahasiswa mampu melakukan analisis data dengan menggunakan metode statistik dengan Seaborn

## Dasar Teori

Pandas adalah ­*open-source library* yang dibuat terutama untuk bekerja dengan data relasional atau berlabel baik secara mudah dan intuitif. Pandas menyediakan struktur data yang cepat, fleksibel, dan ekspresif, intuitif dan mudah digunakan. Library Pandas menyediakan fitur dan fungi yang dapat mempermudah pengguna dalam melakukan analisis data dan visualisasi menggunakan Python. Pandas dibangun dari *library* NumPy yang memiliki kinerja cepat dan produktivitas tinggi bagi pengguna..Pandas sangat cocok untuk digunakan dalam berbagai jenis data seperti:

### Data tabular heterogen, seperti dalam tabel SQL atau spreadsheet Excel

### Data time-series

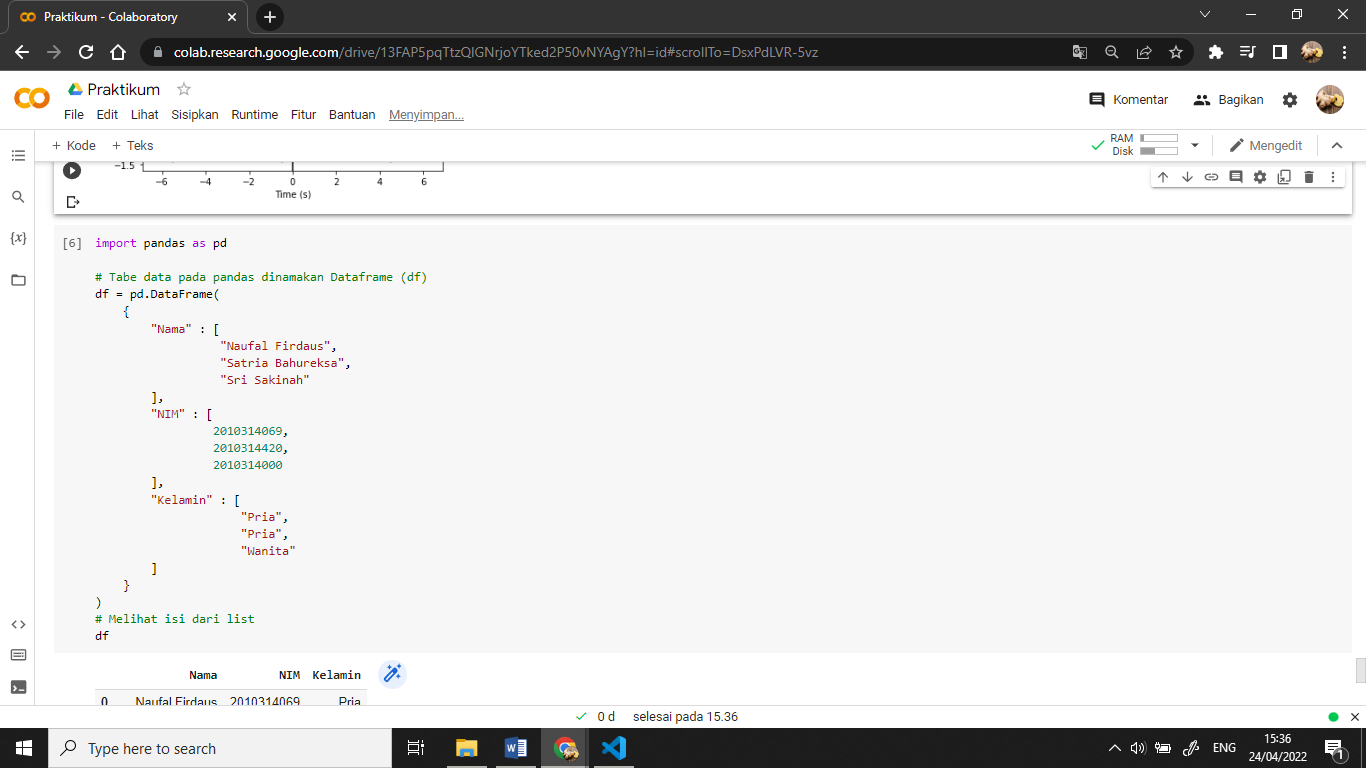
### Data matriks arbitrer dengan label baris dan kolom

### Bentuk lain dari kumpulan data observasional/statistik. Data tidak perlu diberi label sama sekali untuk ditempatkan ke dalam struktur data pandas

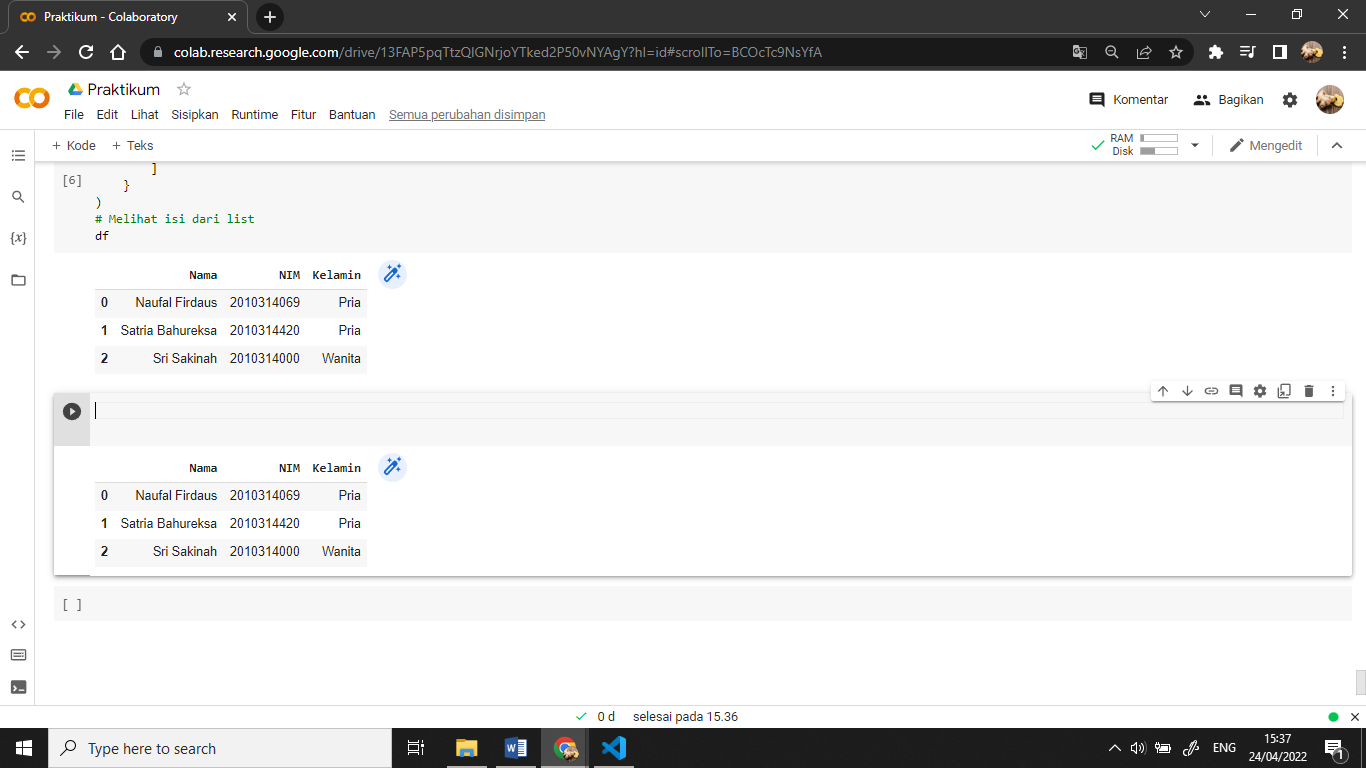
Seaborn adalah ­*open-source library* untuk membuat grafik statistik dengan Python. Seaborn dibangun di atas *library* matplotlib dan terintegrasi erat dengan struktur data pandas. Seaborn membantu dalam menganalisis dan memahami data Anda. Fungsi plotnya beroperasi pada *dataframe* dan *array* yang berisi seluruh kumpulan *datasets* dan secara internal melakukan pemetaan semantik dan agregasi statistik yang diperlukan untuk menghasilkan plot yang informatif.

## Membuat *Dataframe*

Kumpula data yang di representasikan dengan tabel pada Pandas dinamakan *Dataframe, Dataframe* terdiri atas kolom dan baris, kolom pada tabel dinamakan *Series* pada Pandas dan baris dinamakan *row.* Pada proses pembuatan *Dataframe,* data secara manual dimasukkan ke dalam tabel menggunakan tipe data *Dictionary* dalam membuat kolom dan *List* dalam membuat bagian baris dari kolom.

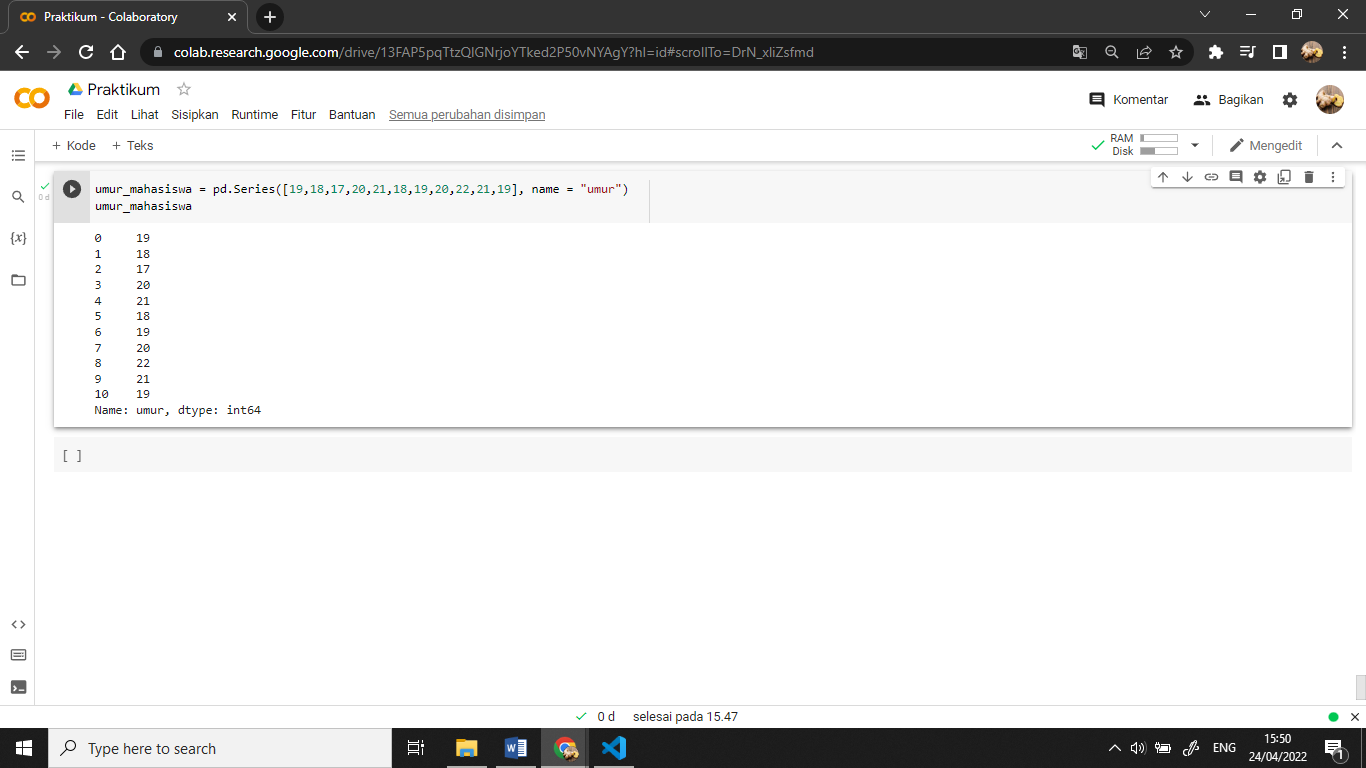


Program diatas merupakan program untuk membuat suatu dataframe dari identitas asistan laboratorium algoritma pemrograman UPN Veteran Jakarta, dengan bagian “Nama”, “NIM”, “Kelamin” merepresentasikan kolom dan bagian di dalam *List* merepresentasikan data dari kolom.



## Membuat *Series* atau kolom

*Dataframe* jenis *Series* merupakan *Dataframe* yang terdiri atas satu kolom, fungsi ini digunakan apabila ingin menciptakan suatu tabel data yang memuat satu informasi atau satu kolom.



## Mengetahui data statistik dari *Dataframe*

Mengetahui data statistik dari suatu data merupakan hal yang penting dalam penelitian, dengan menggunakan *library* Pandas, hal ini bisa dilakukan dengan cepat tanpa harus melakukan kalkulasi secara manual ataupun membuat program secara manual. Data statistik dari suatu kolom dapat diketahui dengan menggunakan dibawah:

min() : Mengetahui nilai terkecil dari data

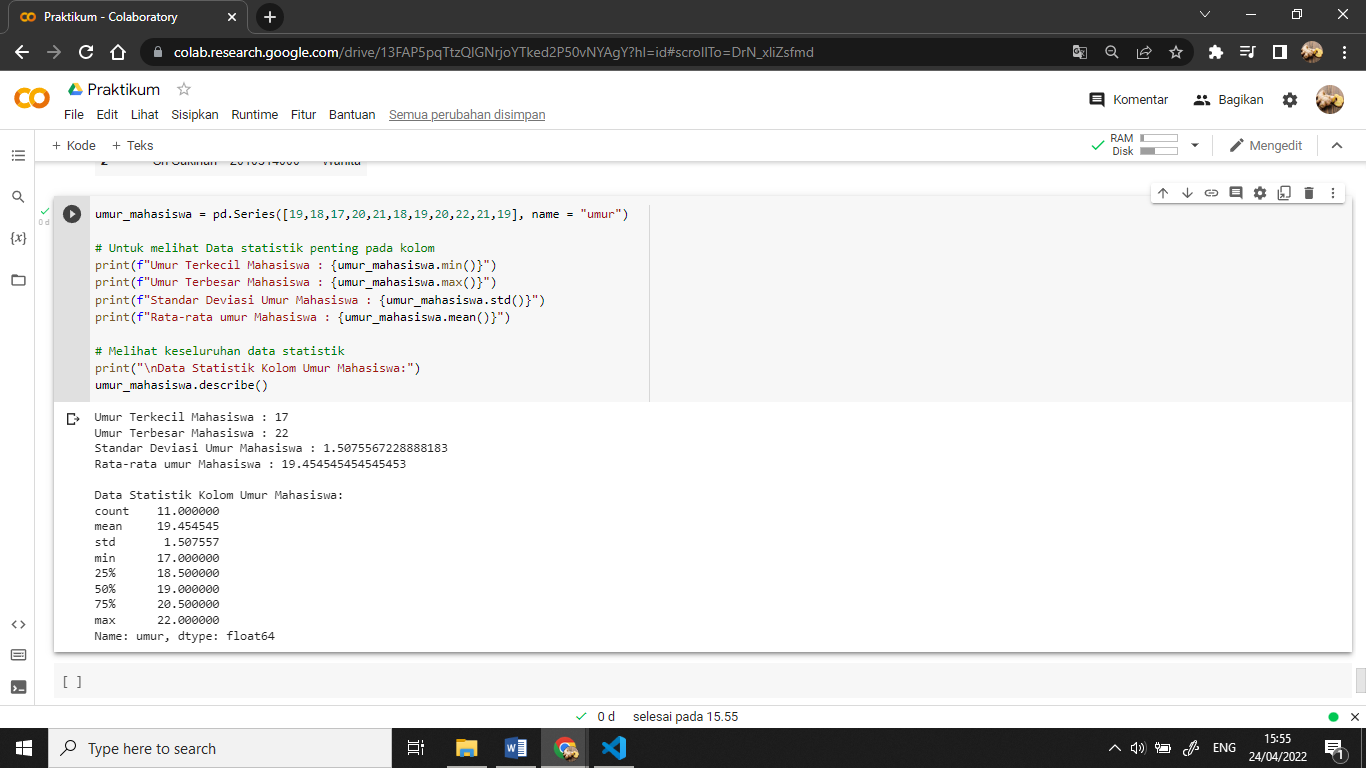
max() : Mengetahui nilai terbesar dari data

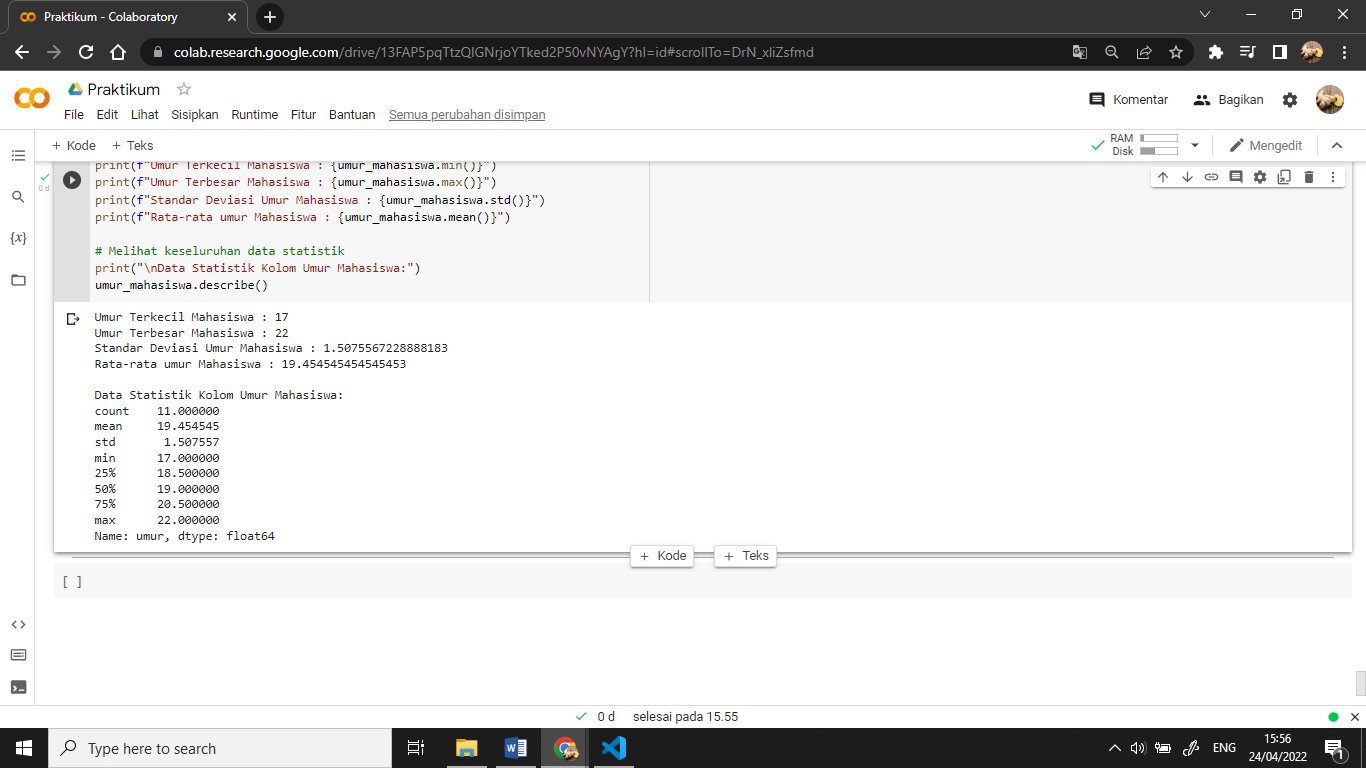
std() : Mengetahui standar deviasi dari data

25% : Mengetahui Q1 dari data

50% : Mengetahui Q2 dari data

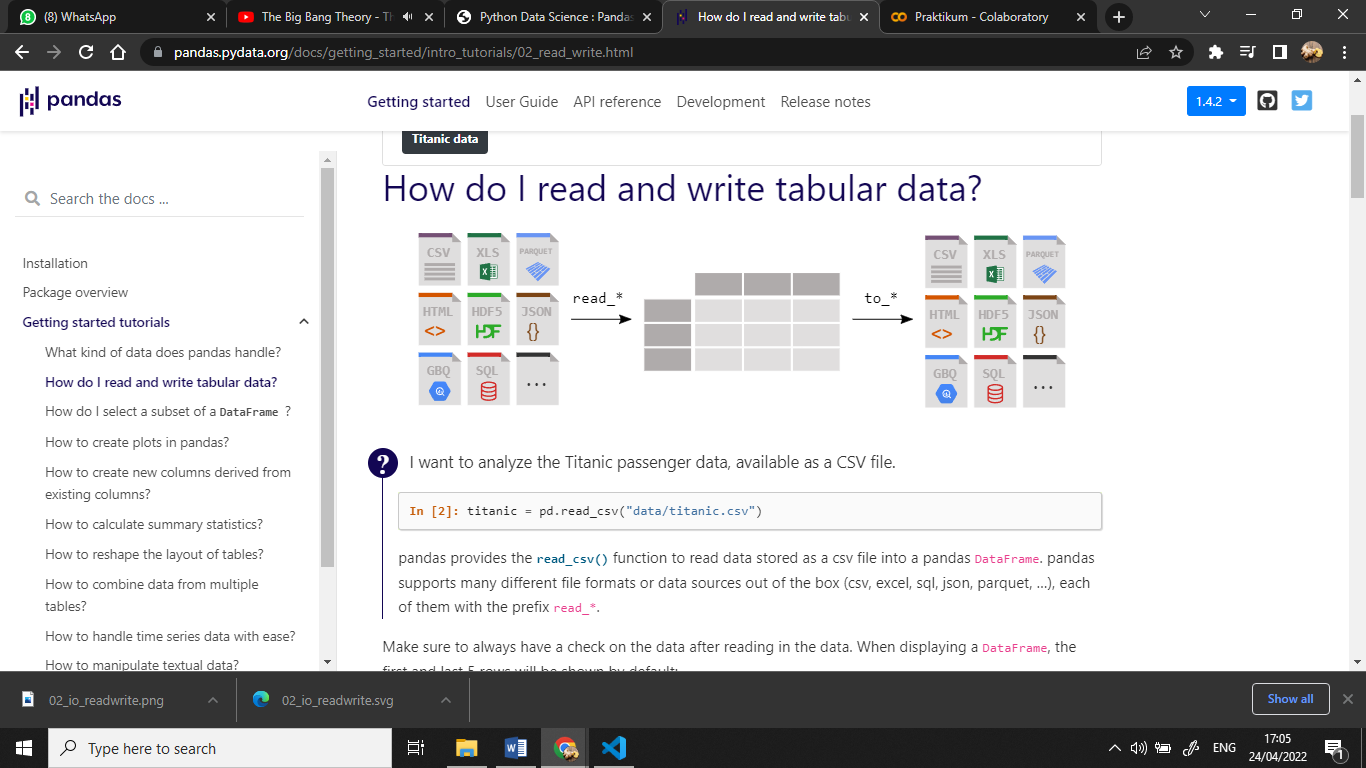
75% : Mengetahui Q3 dari data



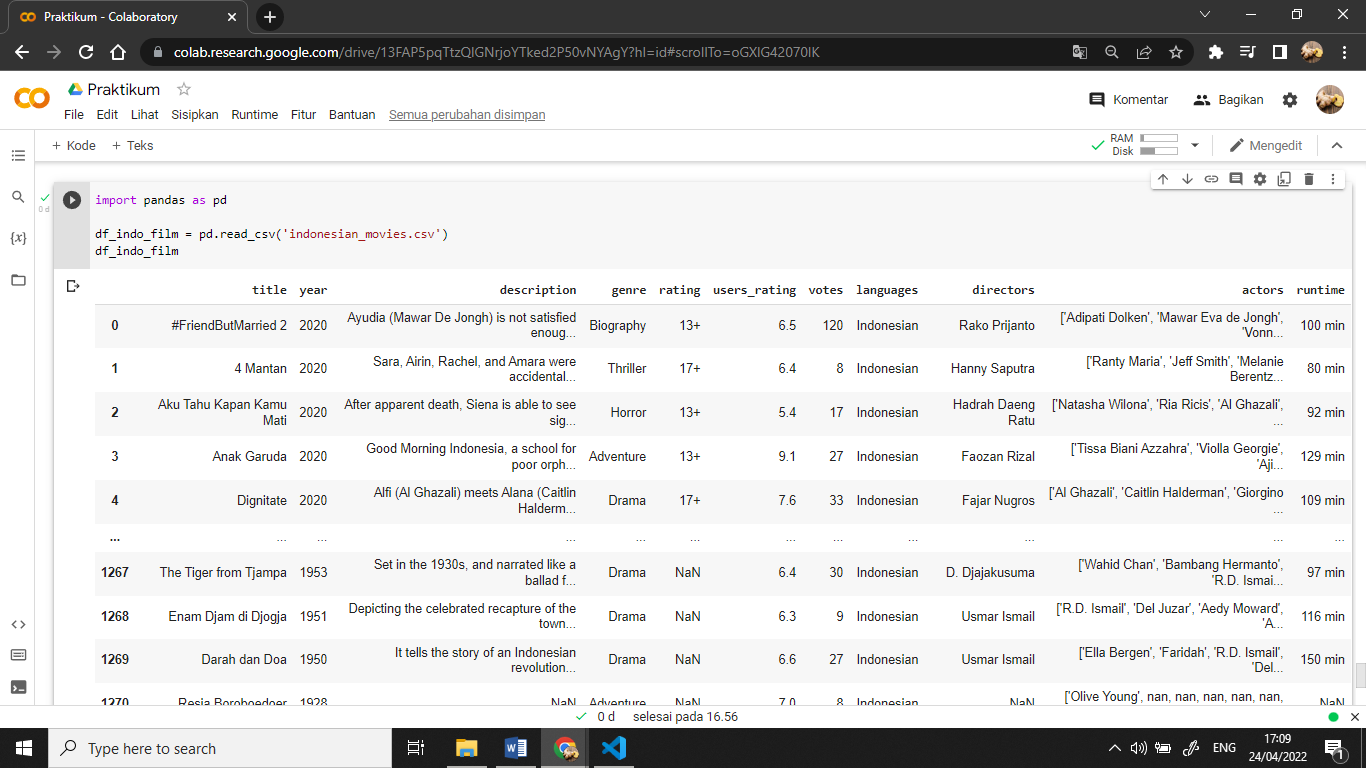


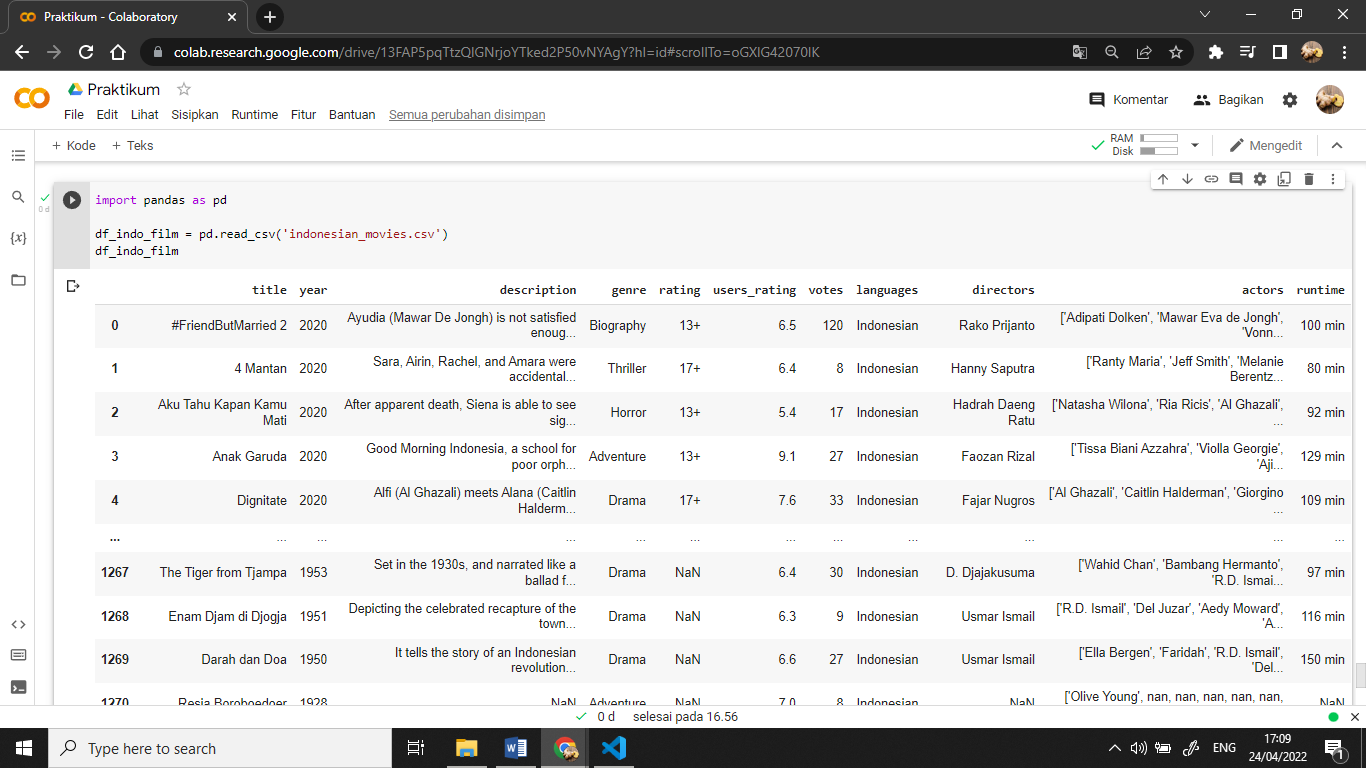
## Memasukkan Data dengan Panda

Pandas dapat membaca data dari banyak sumber salah satunya yakni csv, excel, json, SQL dan HTML dan mampu mengkonversi ke jenis data lainnya. Pandas akan membaca data ke dalam bentuk *Dataframe* dan dapat mengubah *Dataframe* ke dalam bentuk data yang berbeda.



Masukkan data csv yang disediakan asisten laboratorium ke dalam satu folder yang sama dengan program. Program akan mengidentifikasi *path* dari data csv dan Pandas akan membaca data yang ada di dalam file csv tersebut,

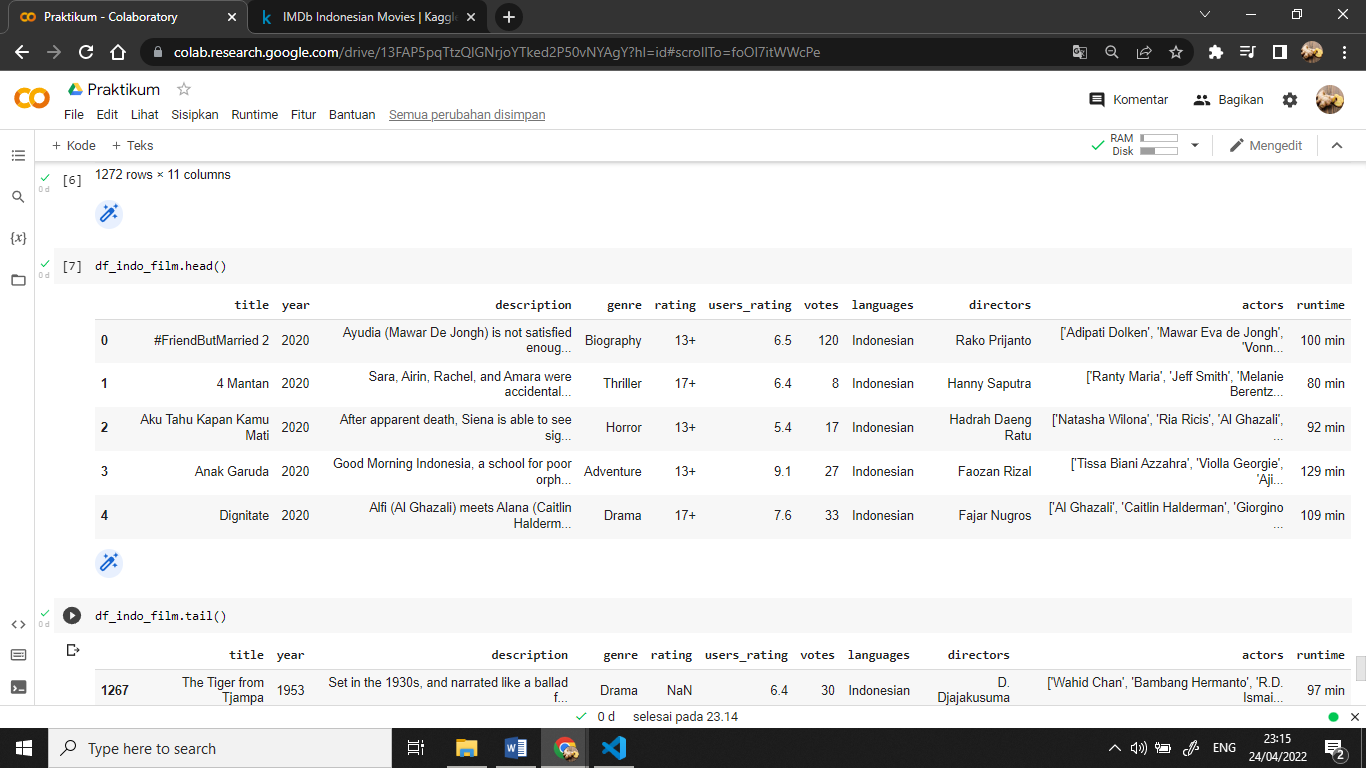




untuk dapat melihat isi dari *Dataframe* dapat dilakukan dengan memanggil nama variabel atau dengan menggunakan fungsi :

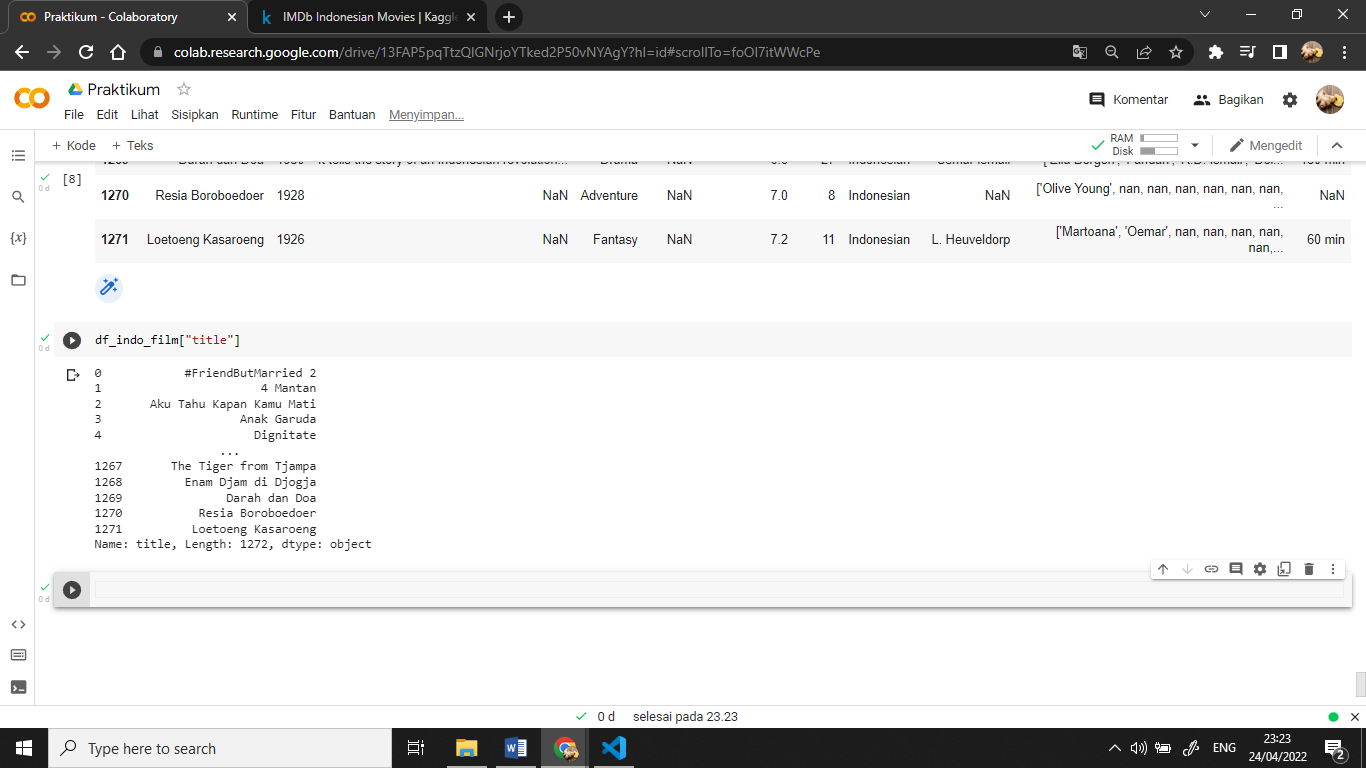
head() : Memanggil lima baris data dari awal

tail() : Memanggil lima baris data dari akhir

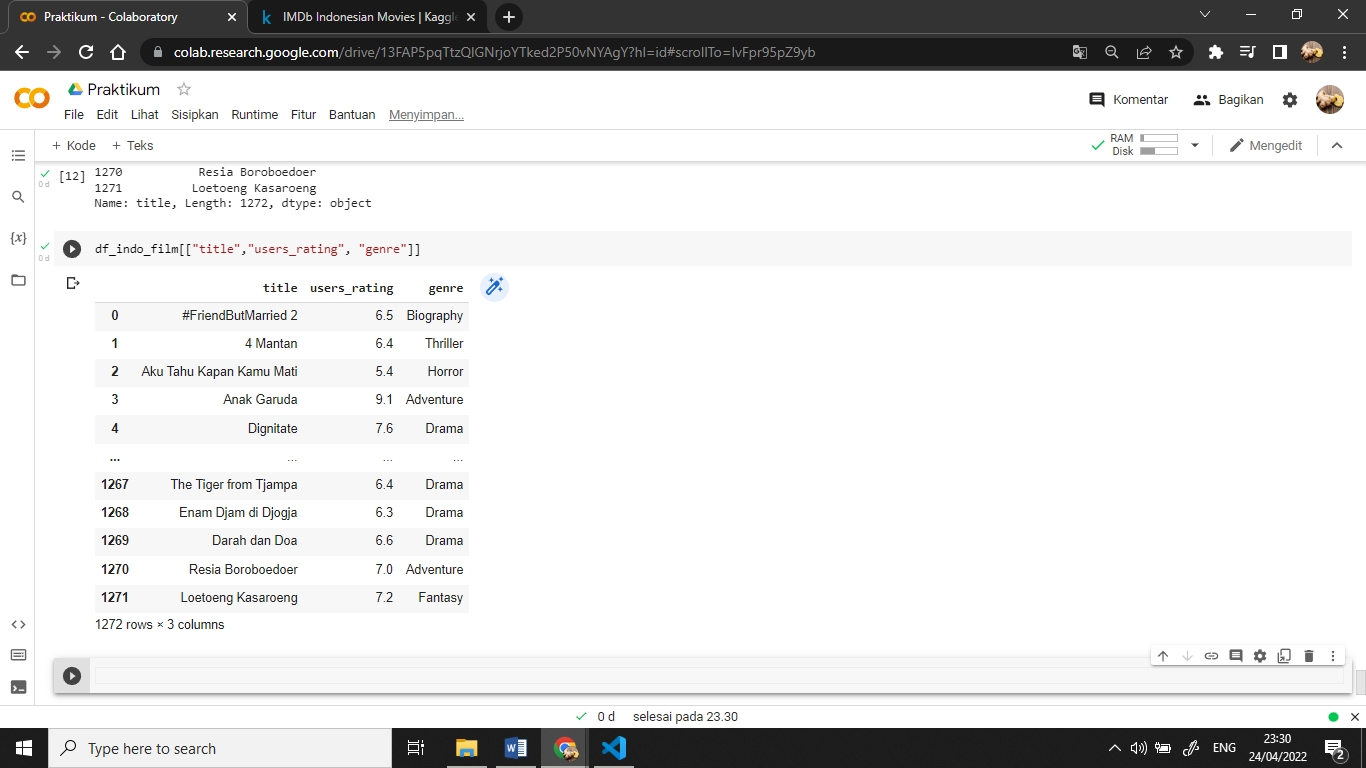


## Mengakses Kolom tertentu pada Data

Memilih data pada kolom tertentu yang ingin di observasi secara langu=sung dapat dicari dengan menggunakan metode seperti dibawah. Metode yang digunakan dibawah memiliki tampilan serupa dengan mengakses data dari tipe data *Dictionary* menggunakan *key values* nya

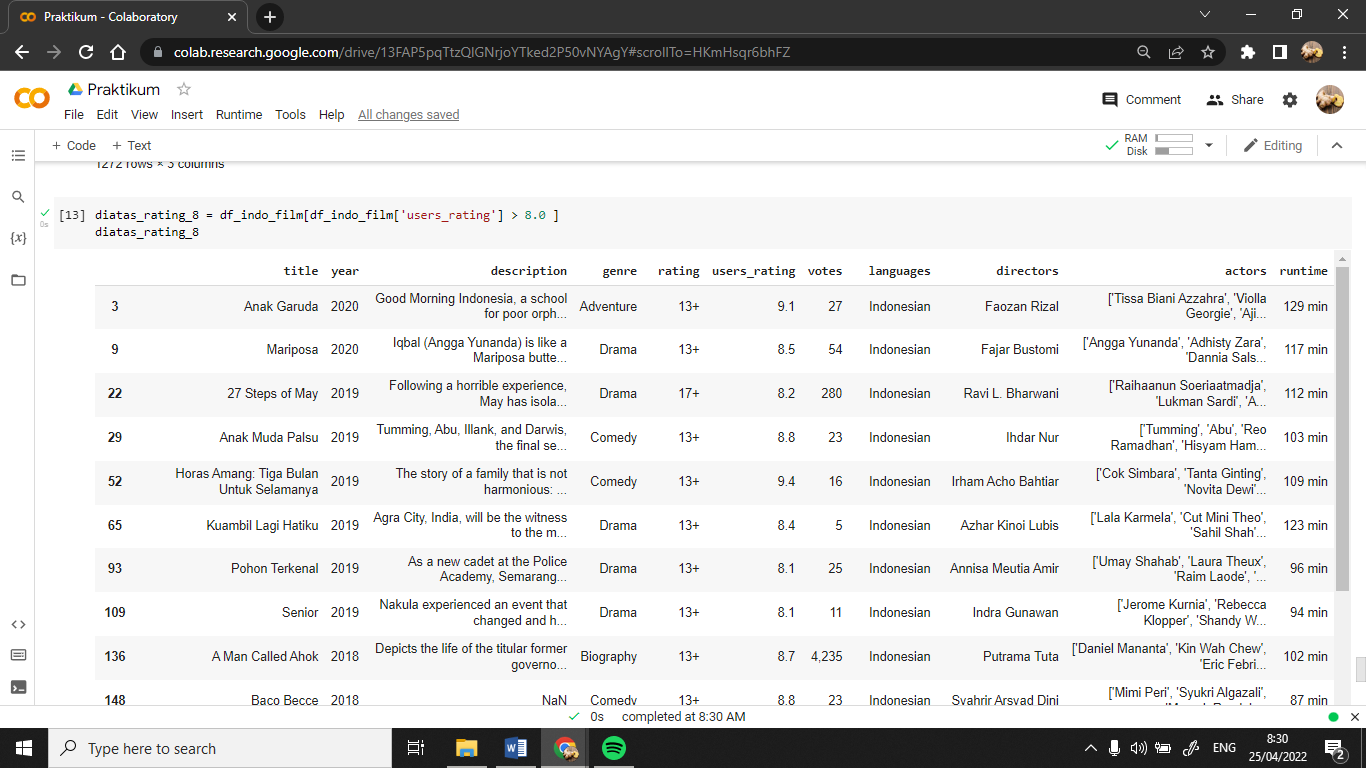


Apabila pengguna ingin mengobservasi lebih dari satu kolom maka dapat menggunakan fungsi berikut:

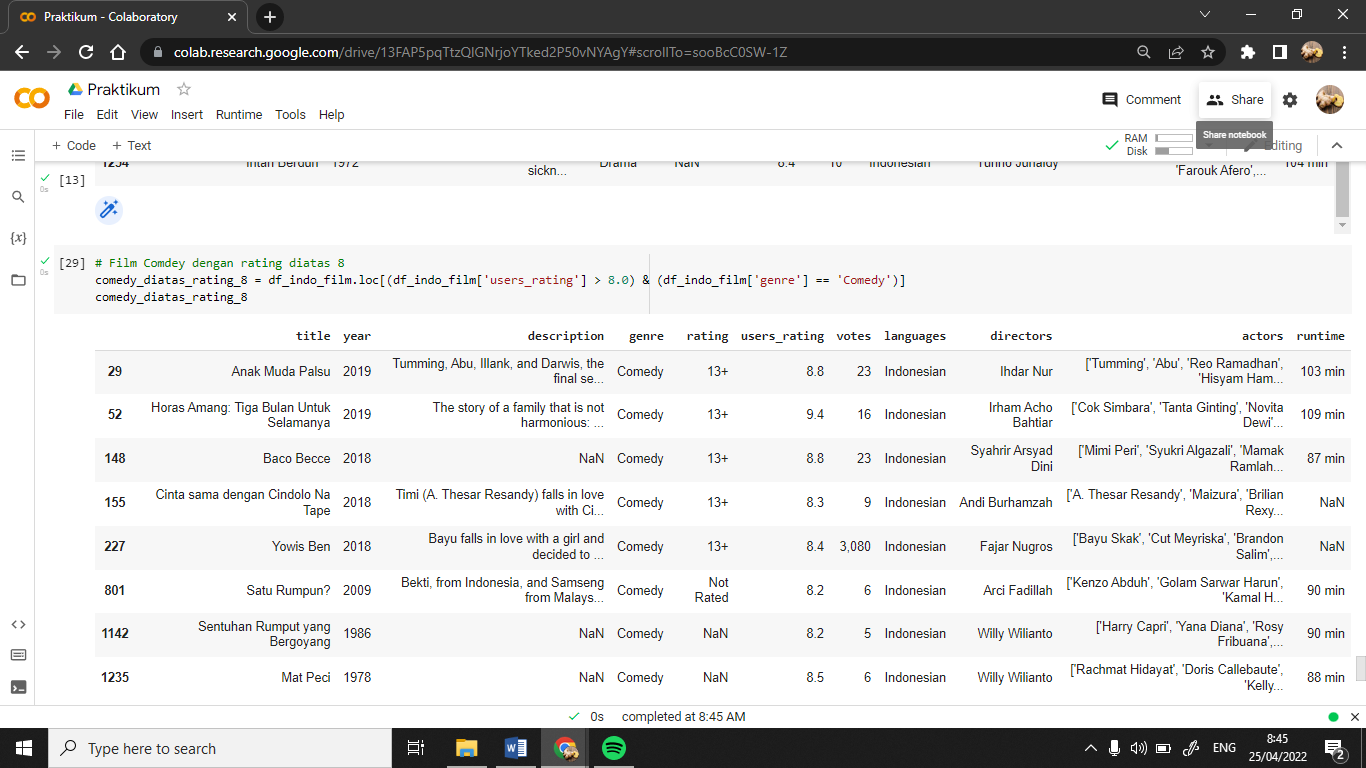


## Melakukan Filterisasi pada Data

Pandas menyediakan fitur untuk dapat melakukan filter pada data dengan memberikan *conditional expression* seperti ( >, ==, !=, <, <=, &, |, ..) yang akan memberikan nilai boolean *True* dan *False*, hanya data yang memiliki nilai *True* yang akan terpilih.

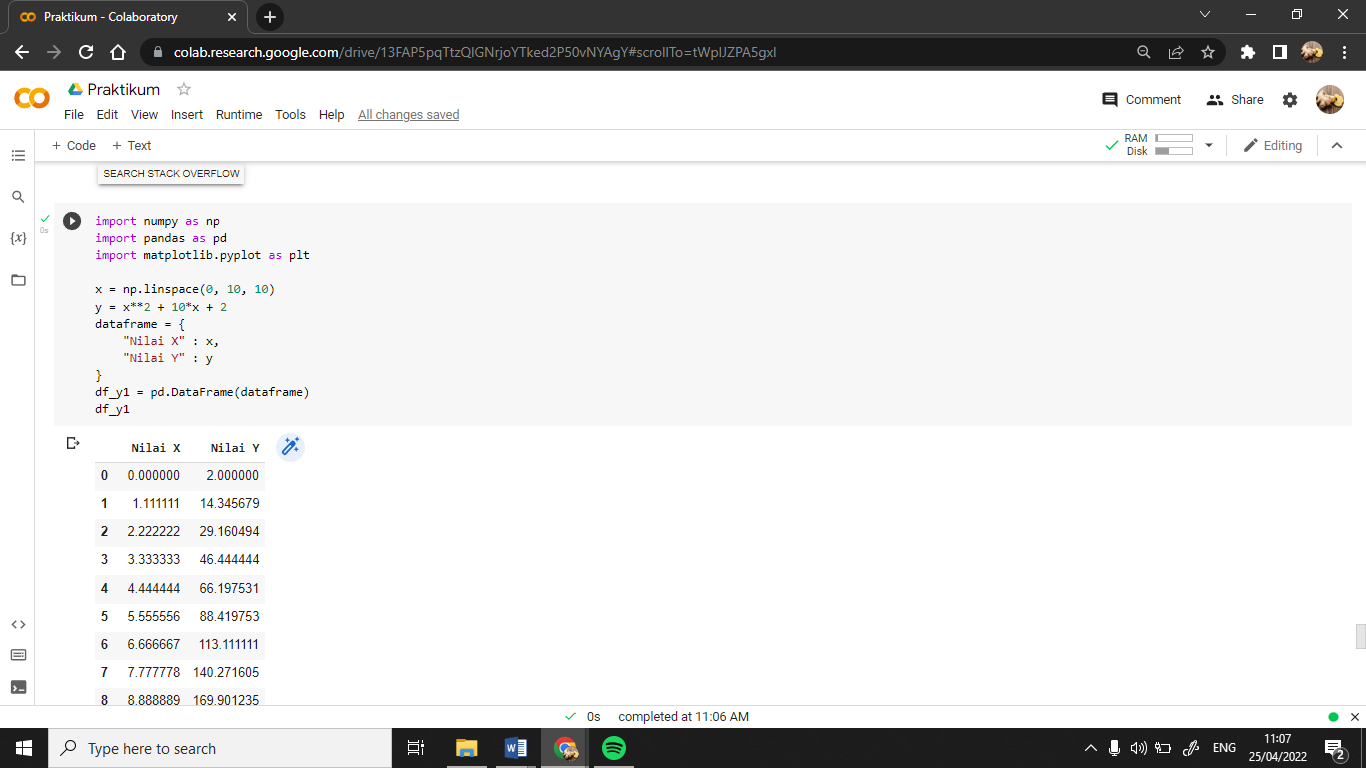
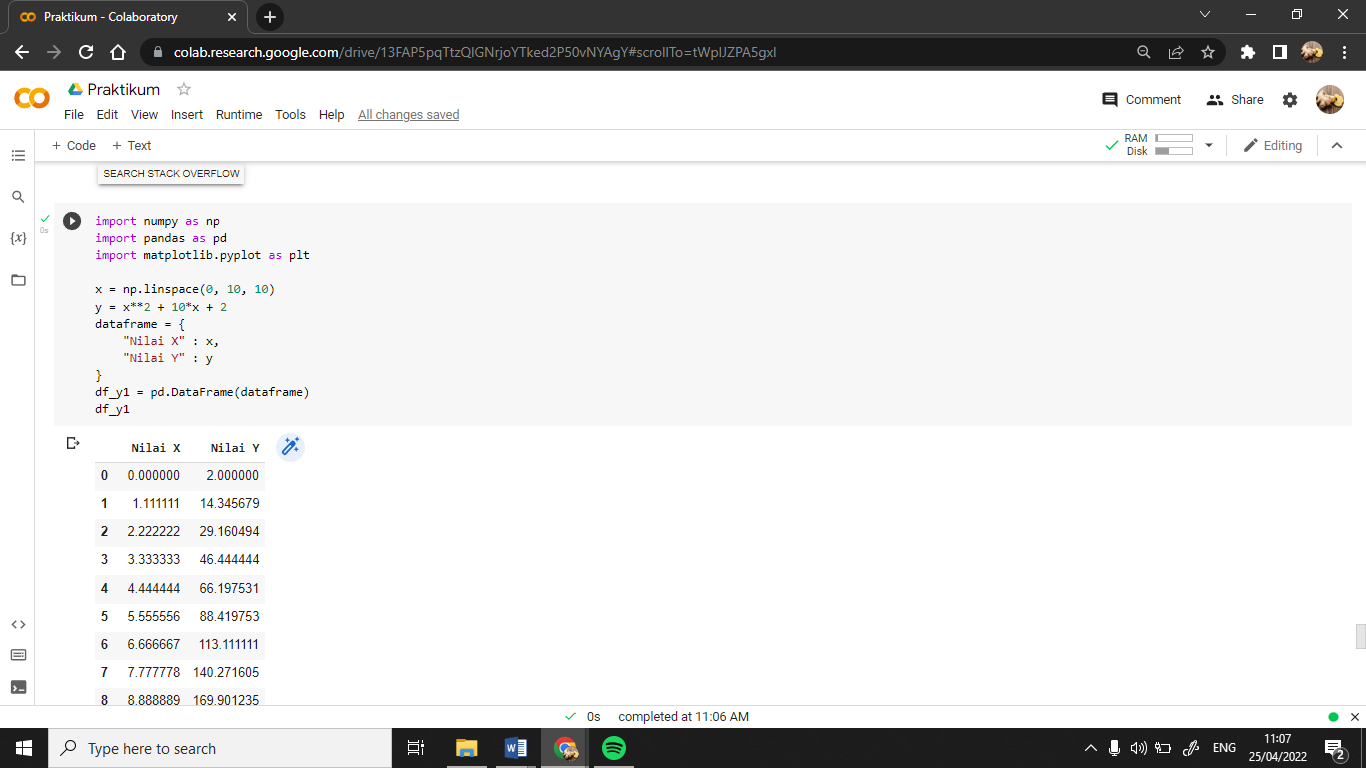


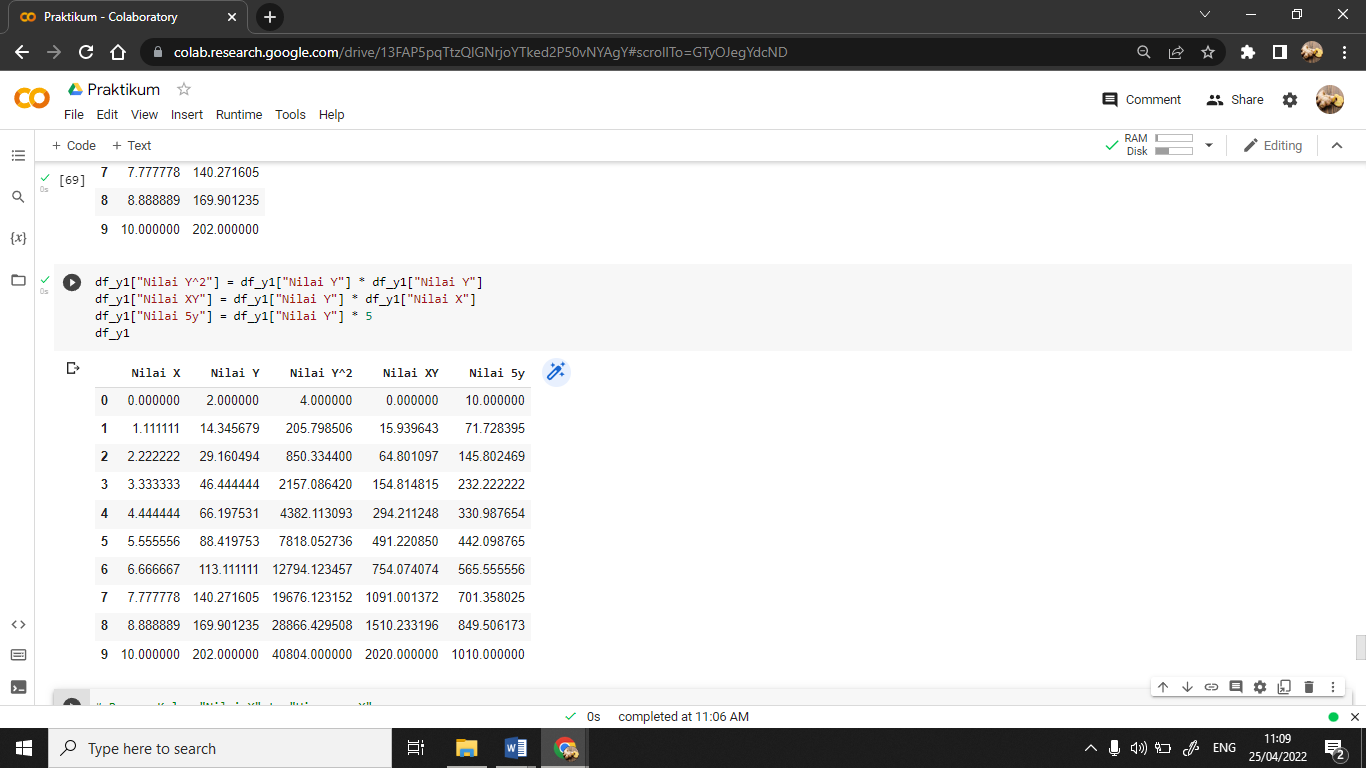
Apabila ingin melakukan filter dengan banyak kondisi maka dapat dilakukan dengan menambahkan *conditional expression* pada program.



## Membuat Dataframe dari Nilai tertentu

Membuat suatu *Dataframe* berdasarkan nilai fungsi yang telah dibuat menggunakan fungsi, yang di awali dengan memetakan nilai x pad ainterval tertentu dan memasukkan nilai fungsi y terhadap nilai x. Pemetaan fungsi ini menggunakan *library* Numpydan Matplotlib.



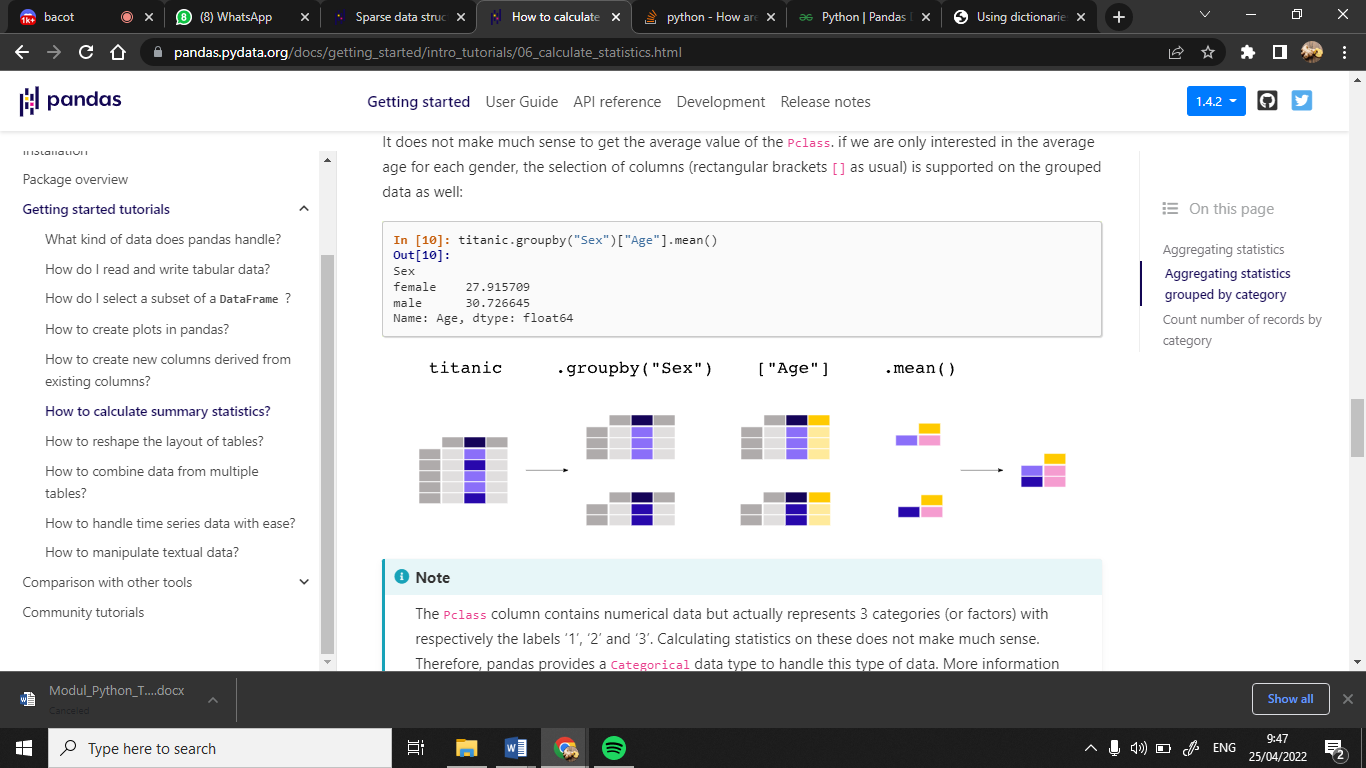
## Mengubah Nama dari Kolom

## Pandas menyediakan fitur untuk dapat mengubah nama dari kolom secara manual, dengan mendeklarasikan nama kolom awal dan nama kolom akhir dengan tipe data *dictionary.*

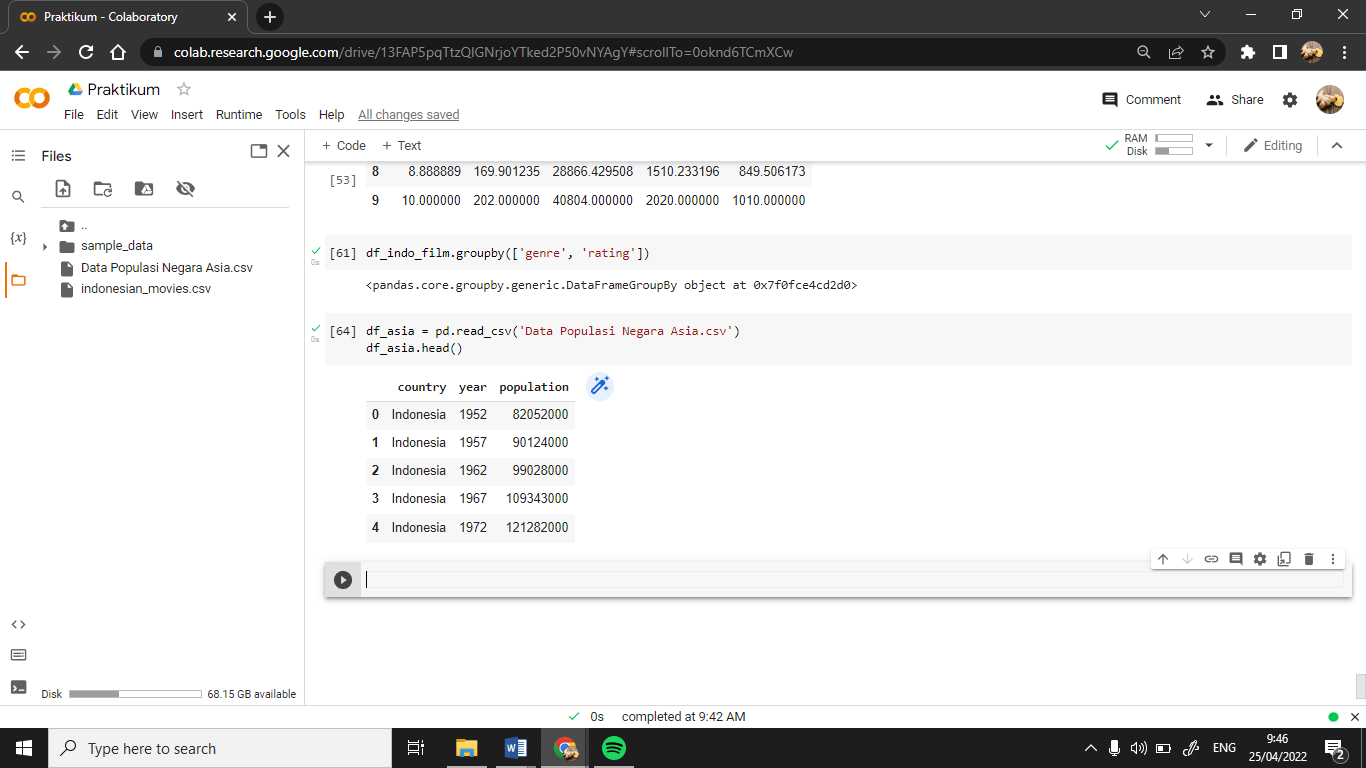
## 

## Mengelompokkan Data

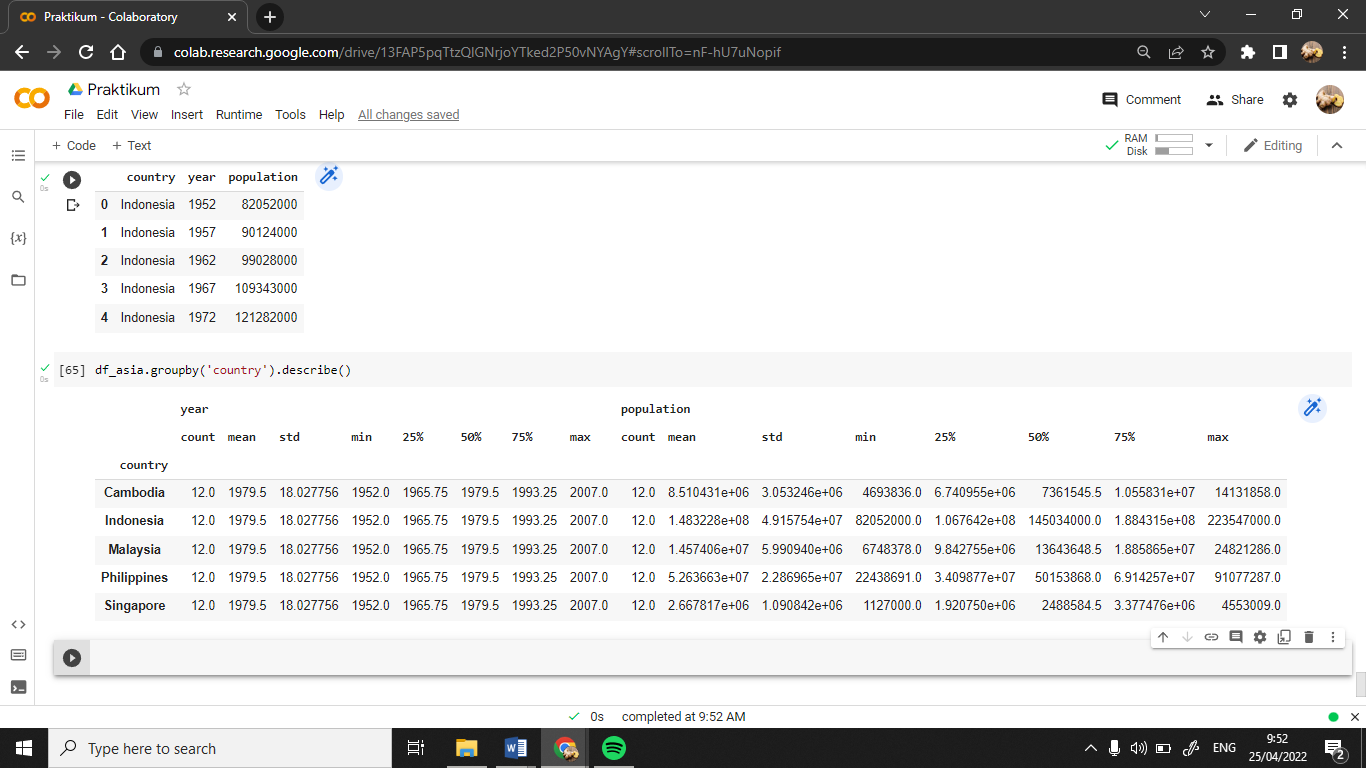
Melakukan pengelompokkan data pada kolom tertentu kedalam sub-bagian tertentu. Proses pengelompokkan diawali dengan mengubah memilih kolom tertentu dari *Dataframe,* kemudian dari kolom tersebut akan di kelompokkan kembali ke dalam kategori tertentu dalam bentuk objek, untuk dapat dilakukan proses analisis lebih lanjut seperti mencari nilai standar deviasi, mean, median, kuartil dan min.

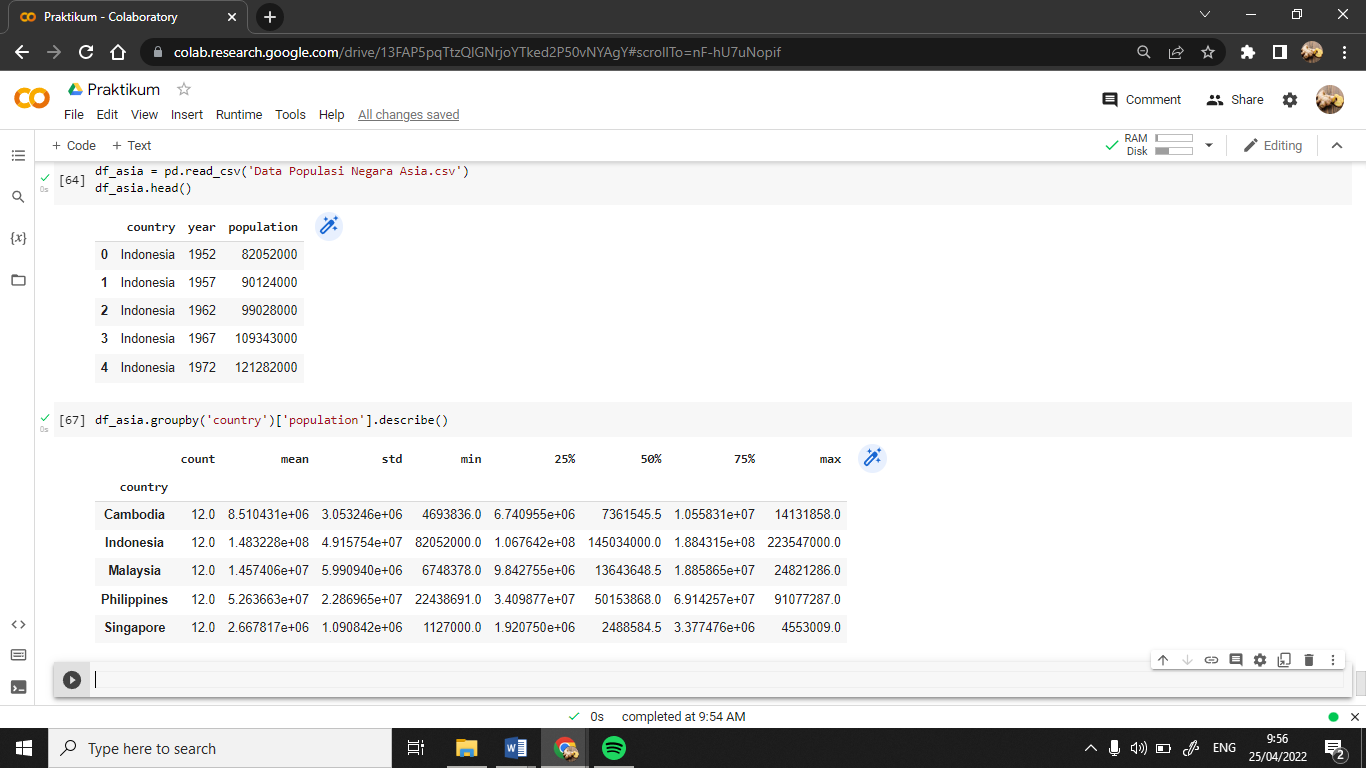


Data yang digunakan adalah data populasi negara asia dalam bentuk CSV, yang akan di konversi ke dalam bentuk *Dataframe* dan kemudian menggunakan fungsi head() untuk dapat melihat kelima data awalnya.



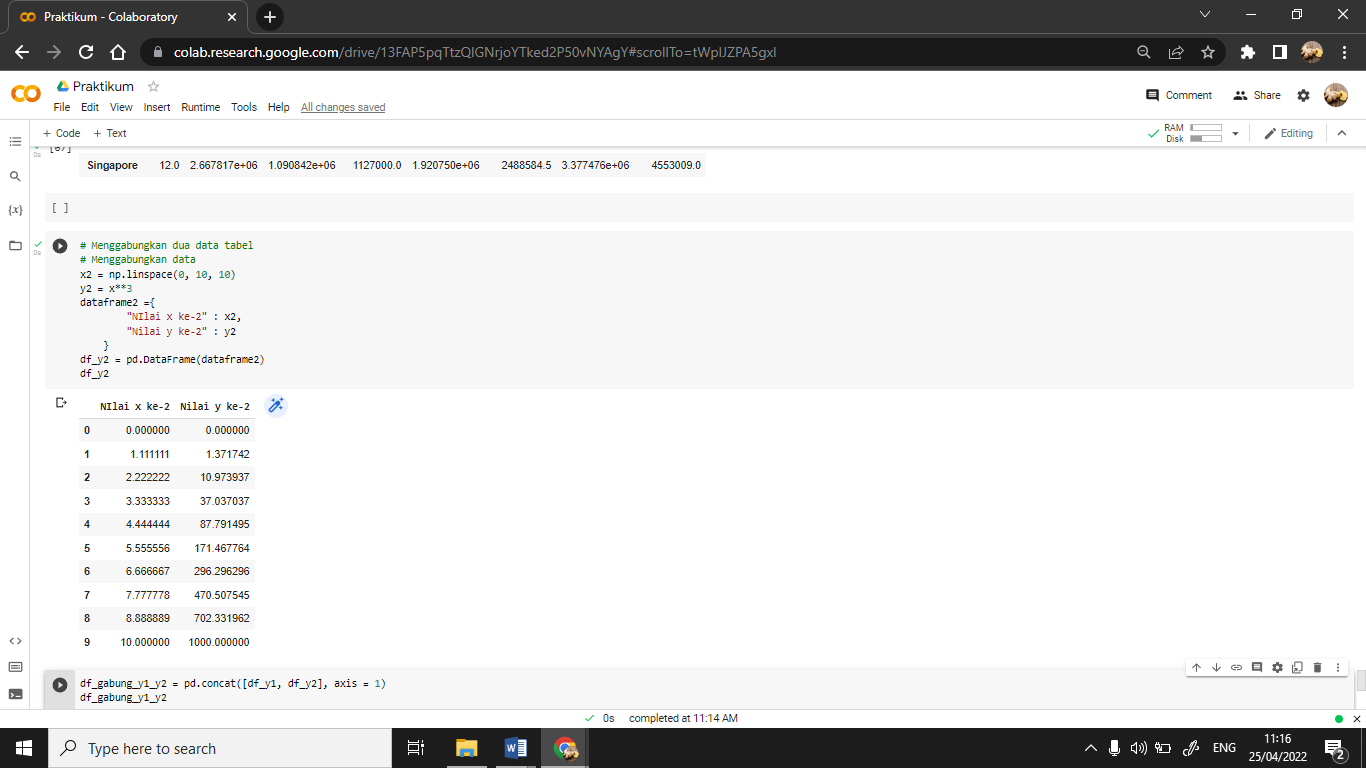
Data akan dikelompokkan berdasarkan negara nya untuk kemudian dicari komponen nilai statistik dari dua komponen kolom integer yakni *year* dan *population.* Apabila ingin mengamati data pada kolom tertentu dapat dengan merincikannya seperti gambar dibawah:



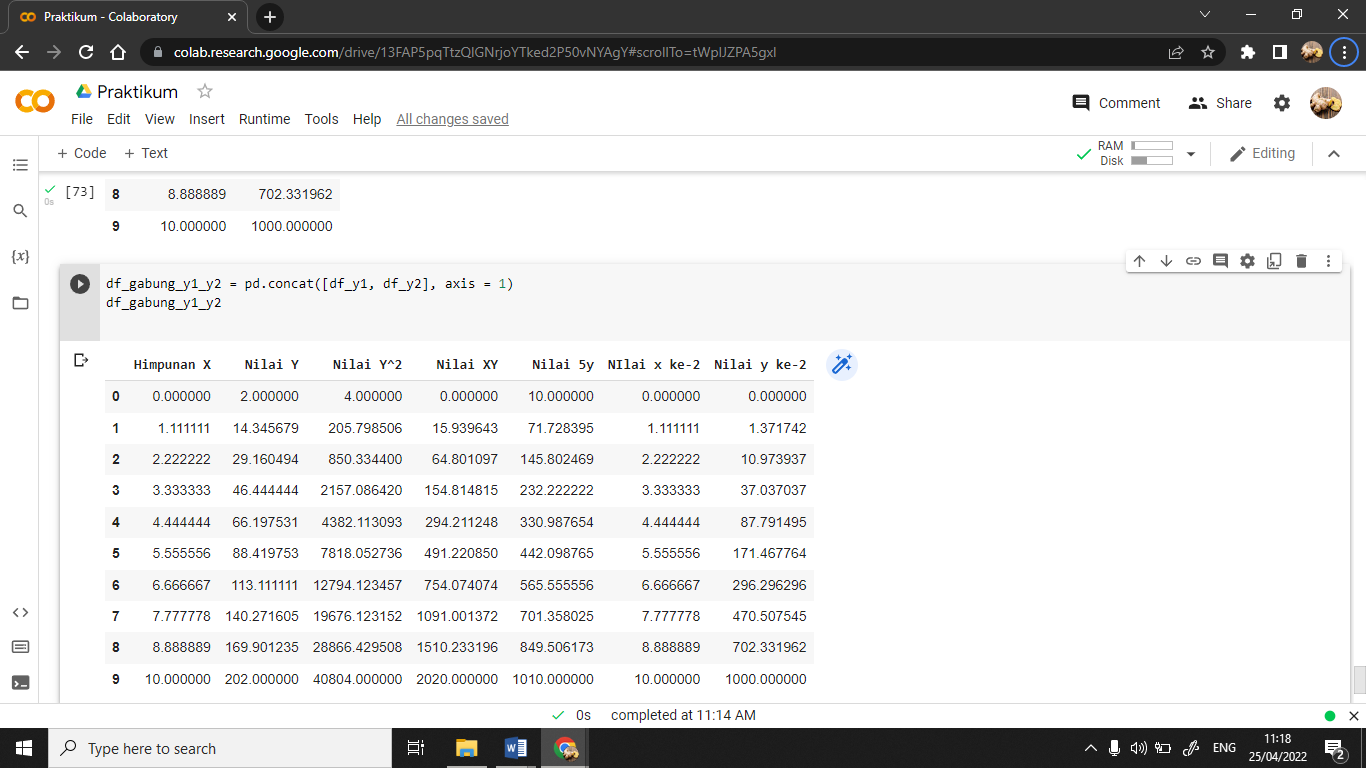


## Menggabungkan Kedua Dataframe

Menggabungkan kedua *Dataframe* yang terpisah dari fungsi *Dataframe* y1 dan fungsi dari *Dataframe* y2, fungsi y2 dapat di jabarkan sebagai fungsi berikut:

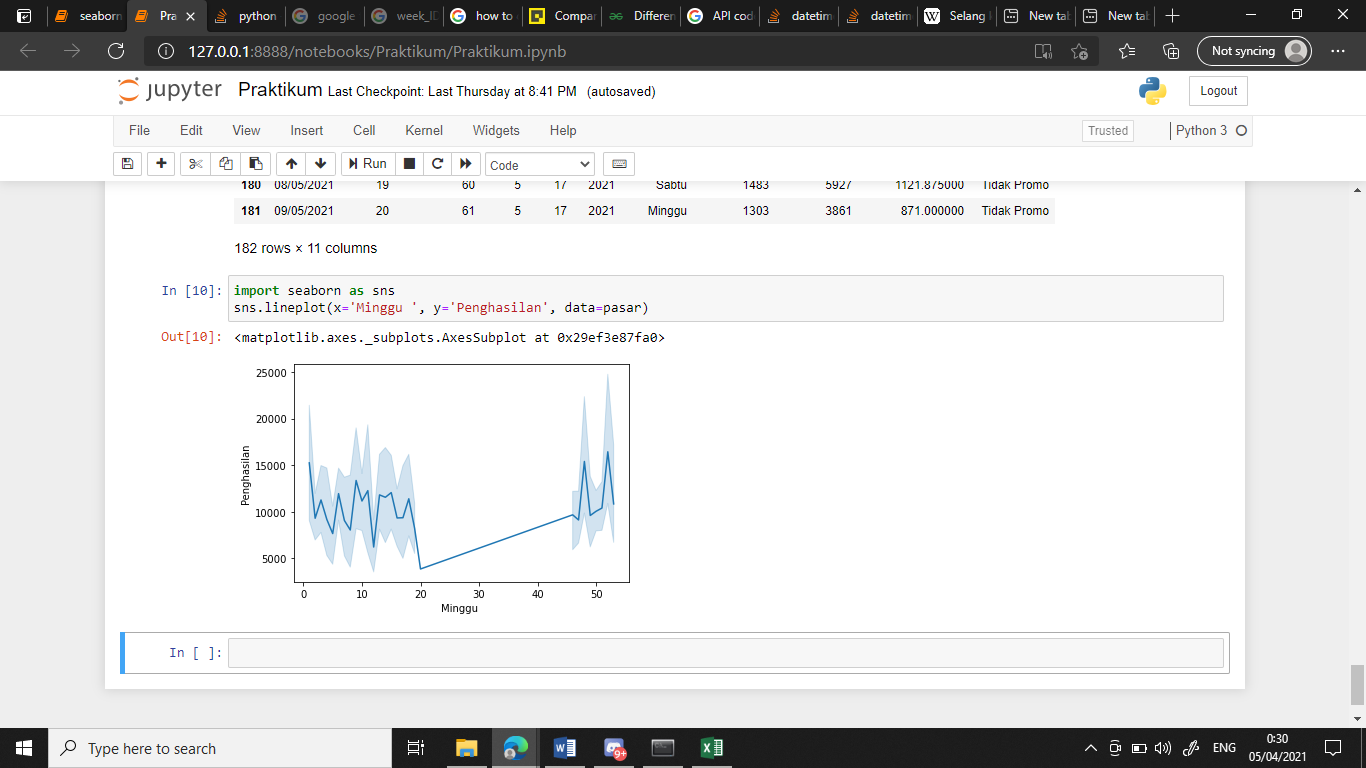


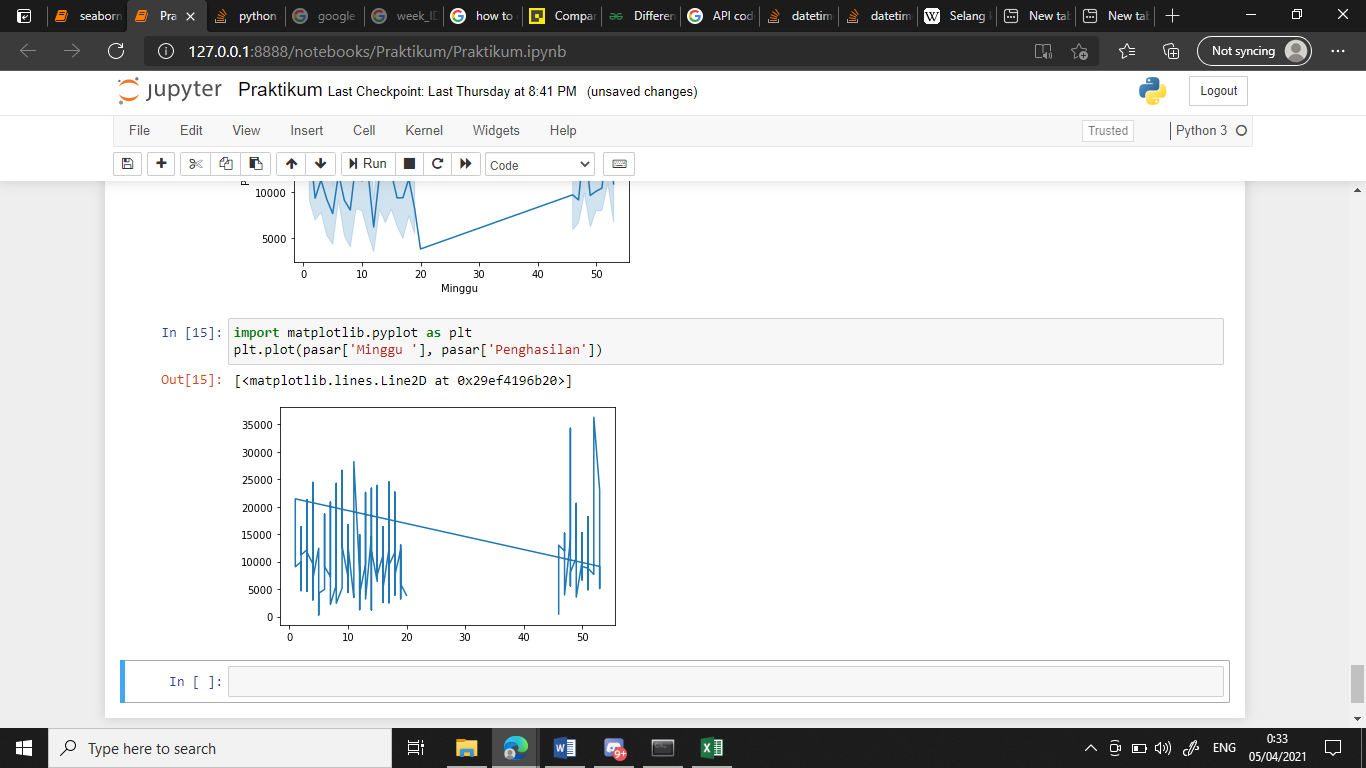
Setelah menggabungkan kedua *Dataframe* menggunakan fungsi pd.concat() dengan memasukkan kedua *Dataframe* ke dalam bentuk tipe data *List.*



## Plot Garis dengan Seaborn

Dalam menganilisis data dalam jumlah yang besar akan lebih akurat jika menggunakan seaborn ketimbang menggunakan matplotlib, karena terdapat error dari visualisasi yang dilakukan oleh matplotlib seperti ini:

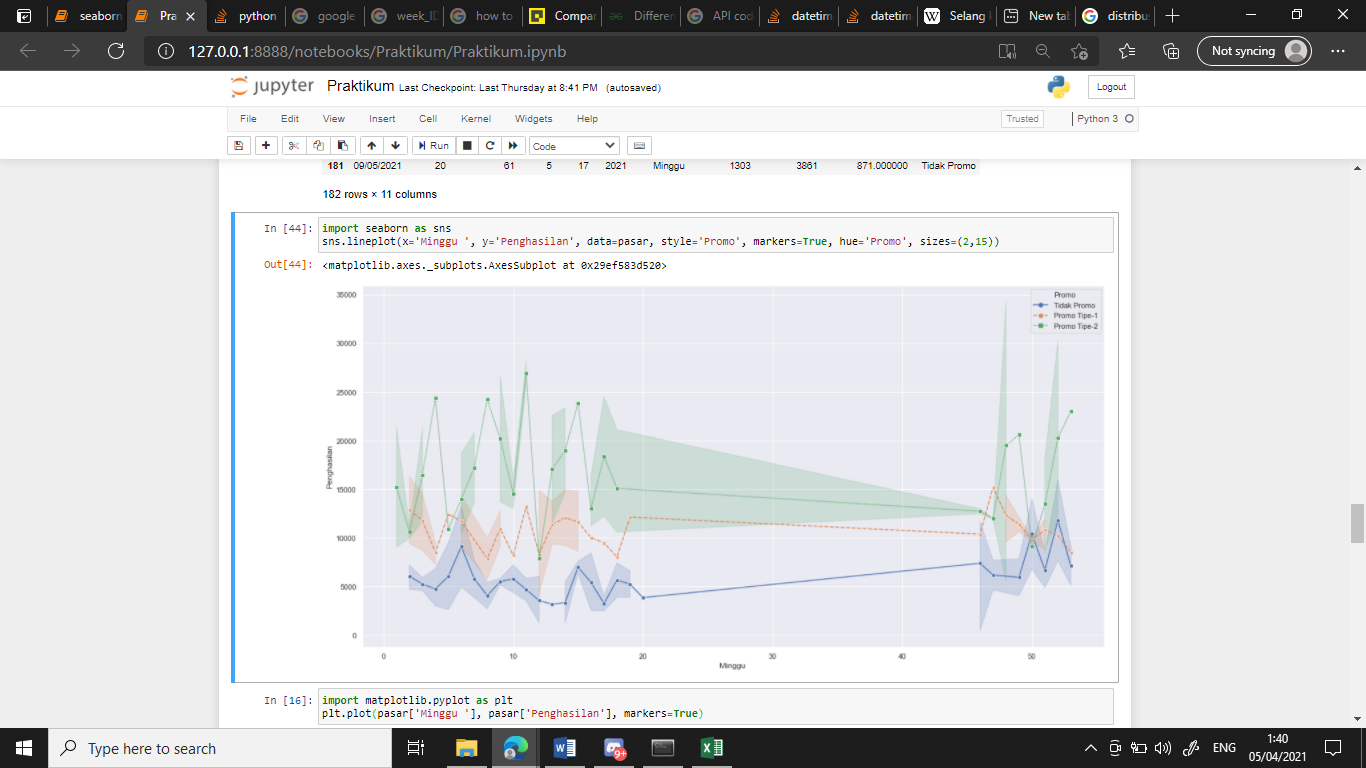




Terdapat error yang dihasilkan dari visualisasi menggunakan matplotlib seperti pada gambar, hal ini membuat seaborn lebih cocok untuk digunakan dalam analisis statistik, selain itu seaborn dilengkapi dengan garis samar-samar yang bertindak sebagai Interval Kepercayaan yang membuat analisis semakin akurat, untuk menggunakan fitur Seaborn diperlukan fungsi awal yakni:

Import seaborn as sns

Kemudian untuk membuat plot line dengan mendeklarasikan kolom mana yang ingin dijadikan axis-x dan variabel yang mana yang ingin dijadikan axis-y serta sumber data yang ingin digunakan .



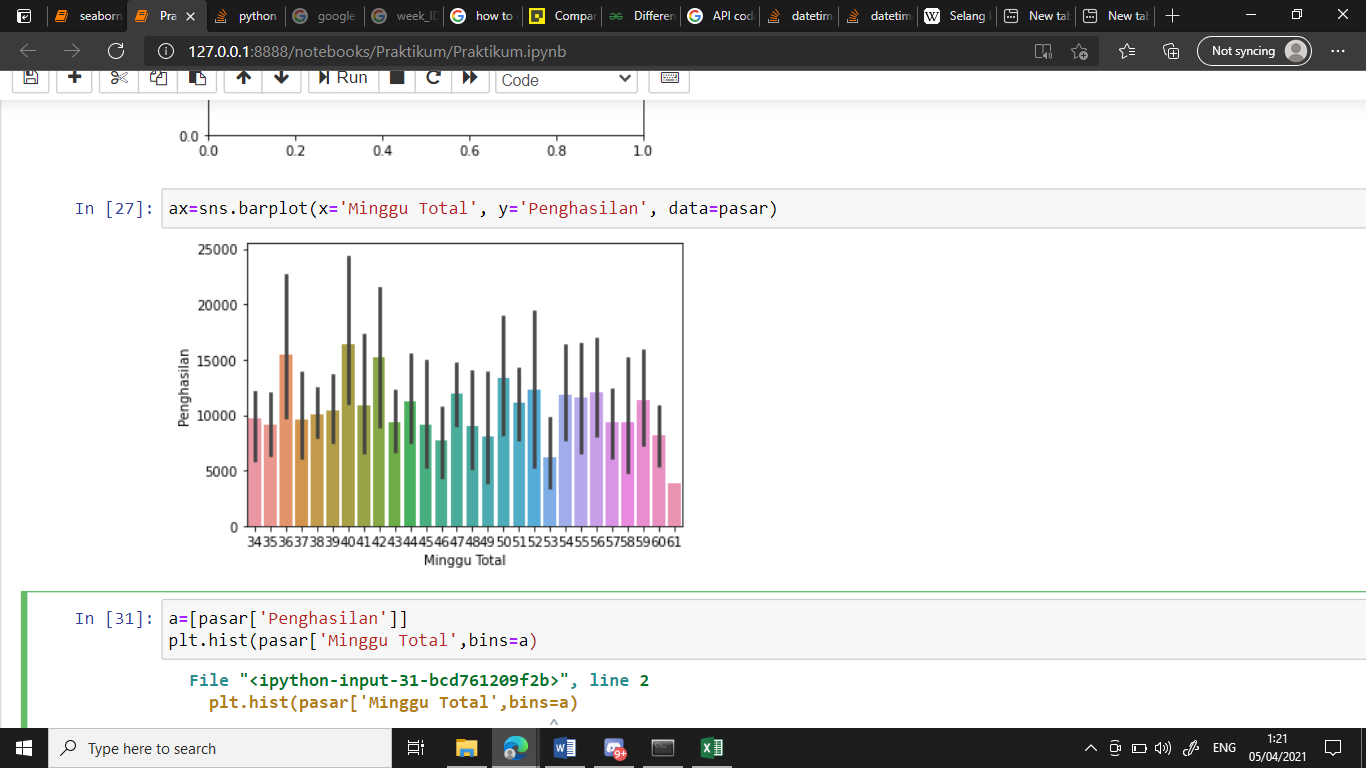
Jika ingin mengklasifikan plot menjadi tiga bagian dapat digunakan fungsi ‘hue’ dan untuk memberikan penanda menggunakan fungsi markers :

hue=’variabel’

markers=True

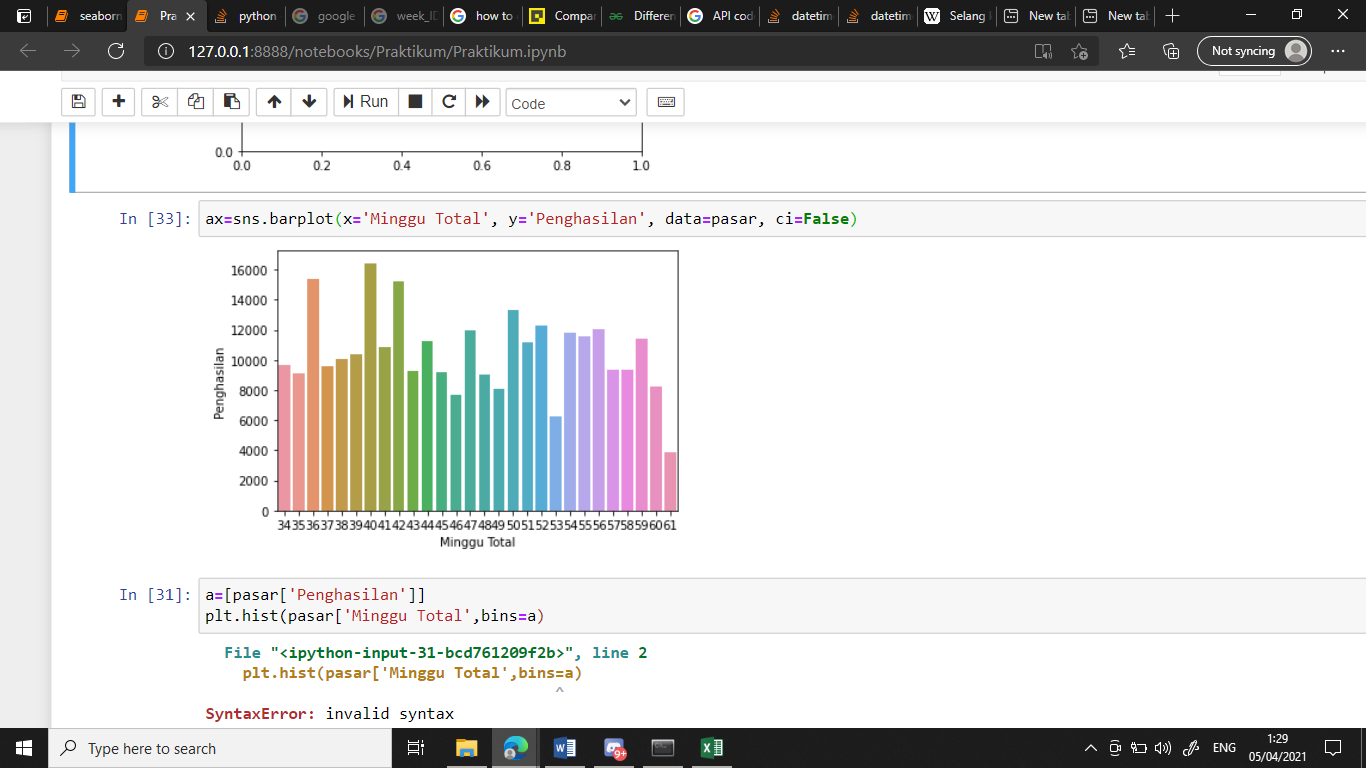
## Plot Histogram dengan Seaborn

Menggunakan data dari data eksternal untuk menghasilkan histogram, garis hitam di tengah menunjukkan nilai interval kepercayaan



Untuk menghilangkan nilai interval kepercayaan dapat menggunakan fungsi

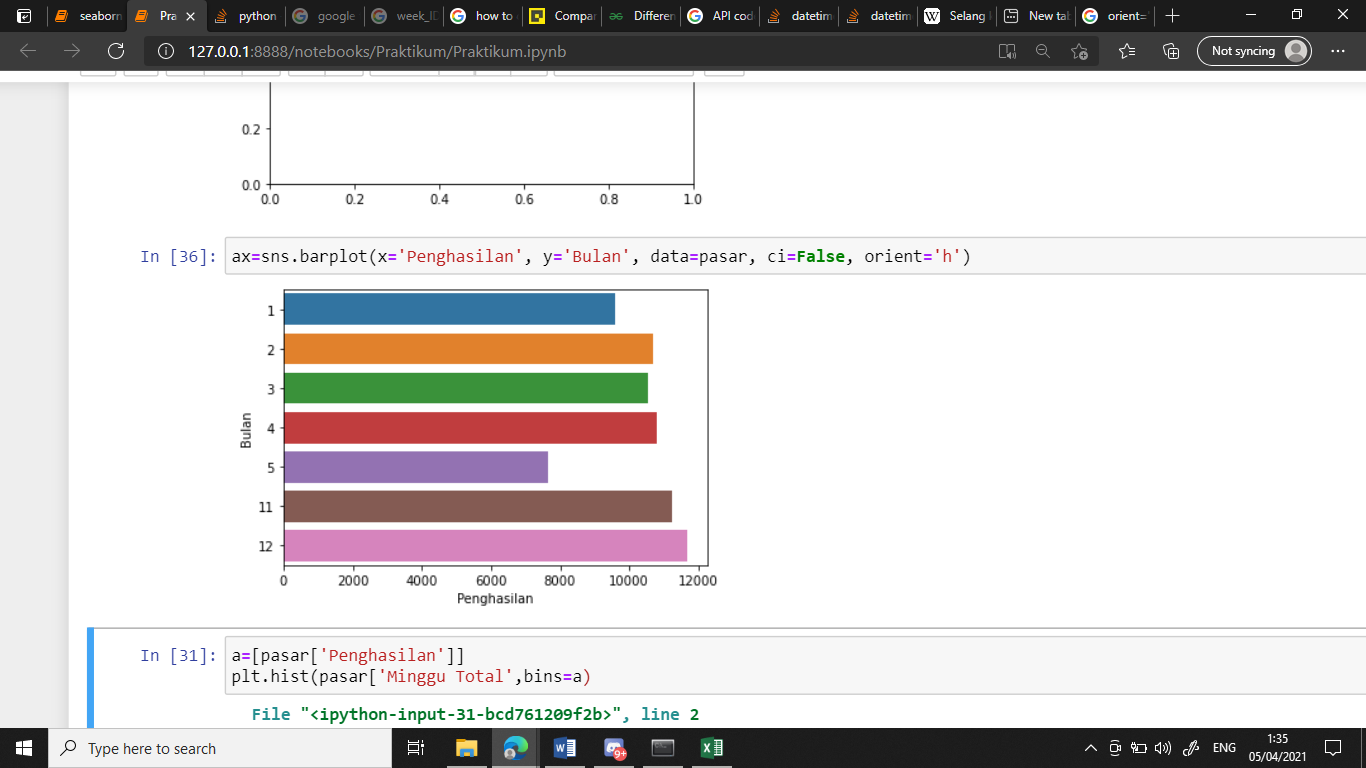
Ci=False



Untuk membuat data dalam bentuk horizontal, dapat mengganti

Orient=’h’

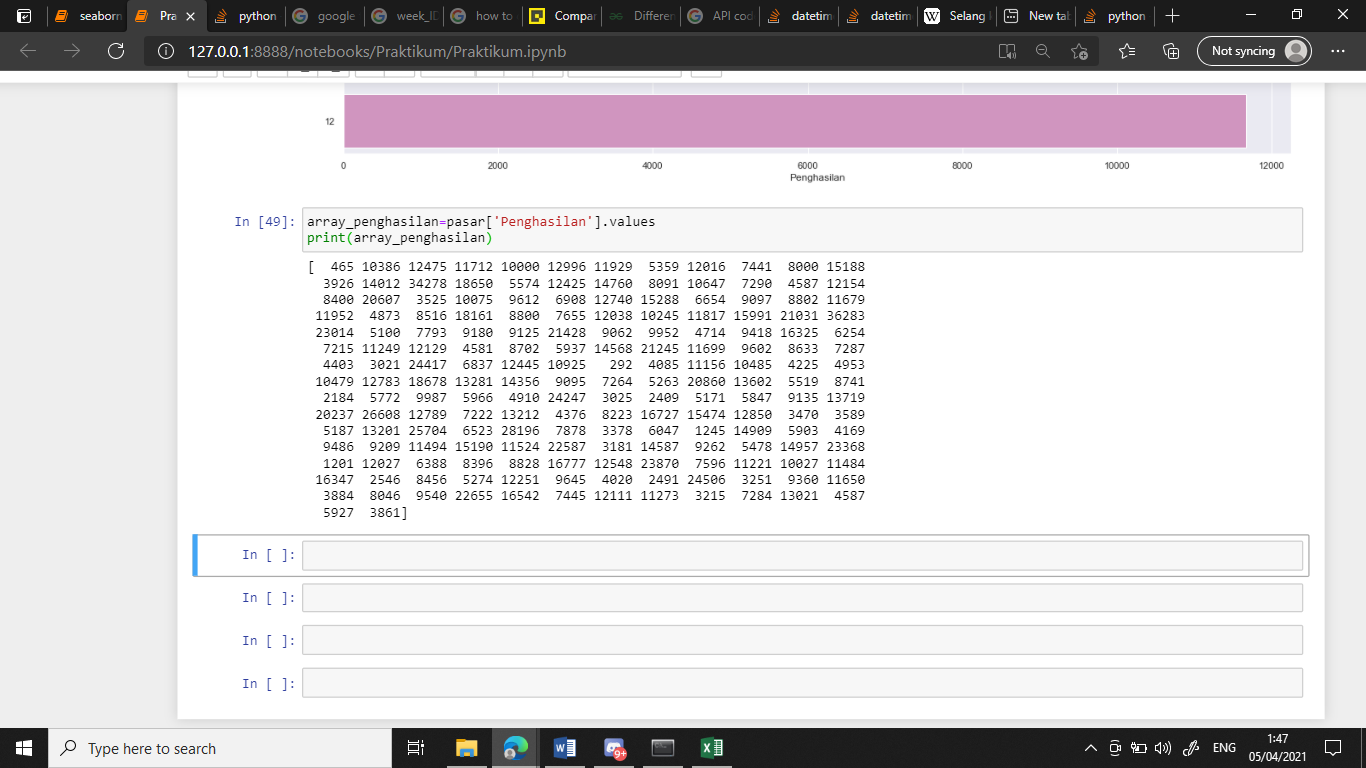
Dengan ‘h’ sebagai horizontal



## Plot Distribusi dengan Seaborn

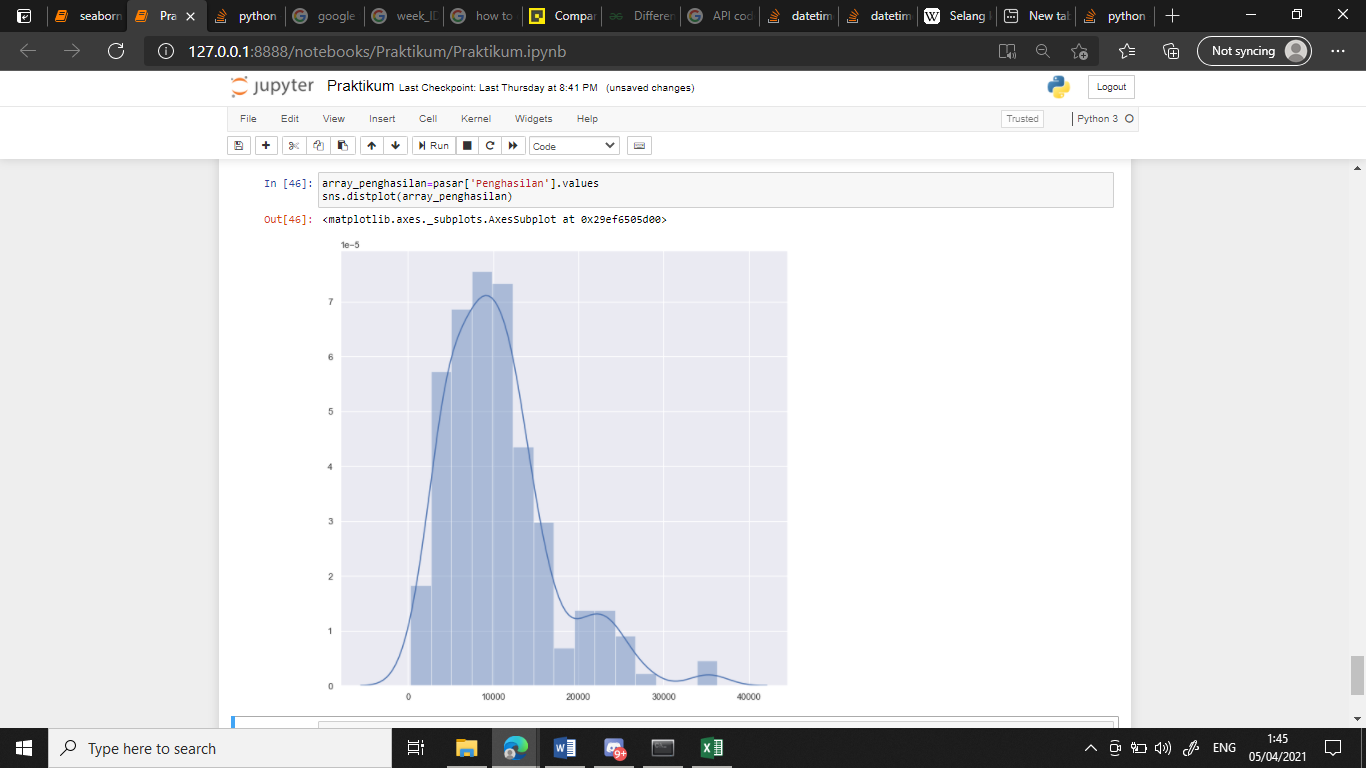
Menggunakan grafik distribusi untuk memplot grafik yang diinginkan yang dilengkapi dengan KDE ( Kernel Density Estimacy ) yang mampu memperkirakan tingkat probabilitas variabel acak pada grafik menggunakan fungsi values untuk mendapatkan nilai di dalam kolom yang diinginkan menjadi suatu array

pasar[‘Penghasilan’].values



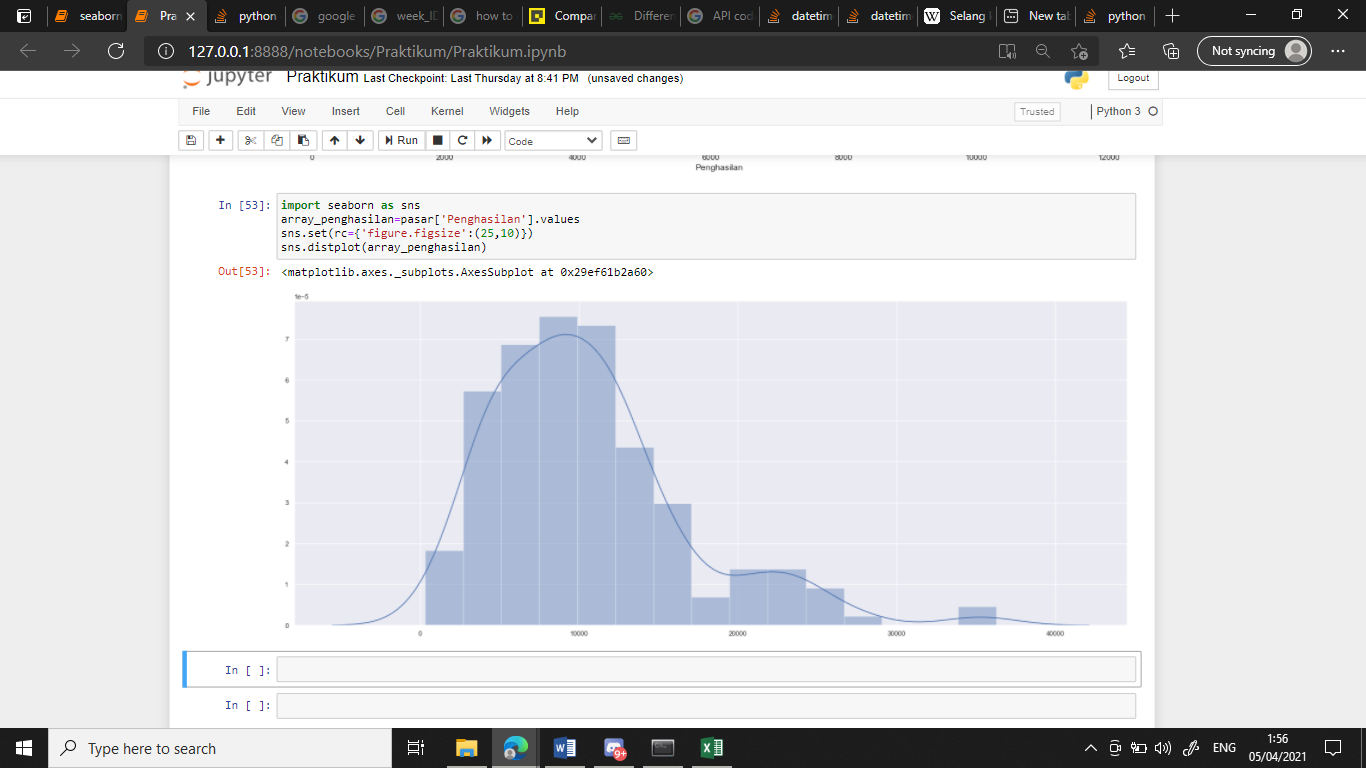
Fungsi yang digunakan untuk membuat grafik dari distribusi plot

sns.displot(variabel)



Untuk mengubah ukuran grafik dapat menggunakan fungsi

sns.set(rc={'figure.figsize':(baris,kolom)})



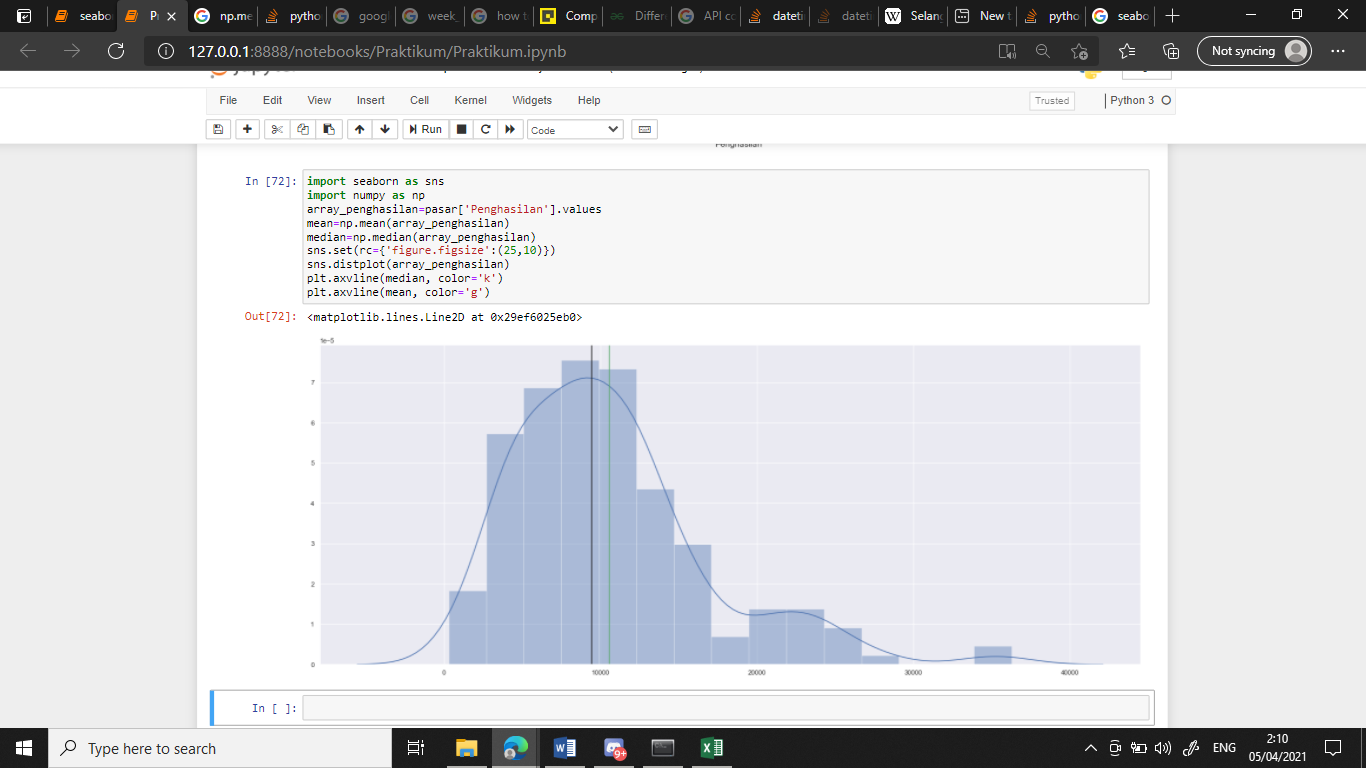
Untuk memberikan nilai median dan mean dapat menggunakan fungsi

Variabel.mean()

Variabel.median()

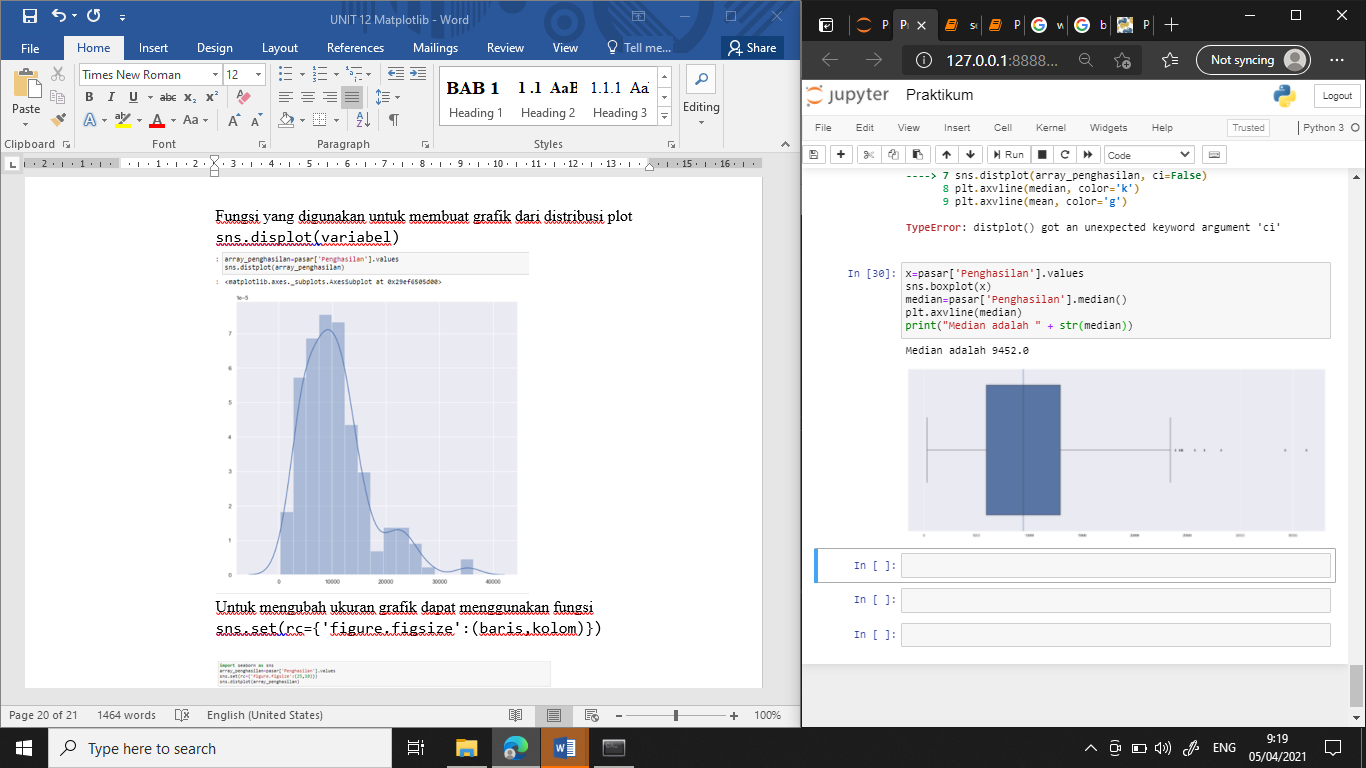
Diberikan garis vertikal untuk memudahkan proses identifikasi dalam mencari letak median dan mean menggunakan fungsi

Plot.axvline(variabel)

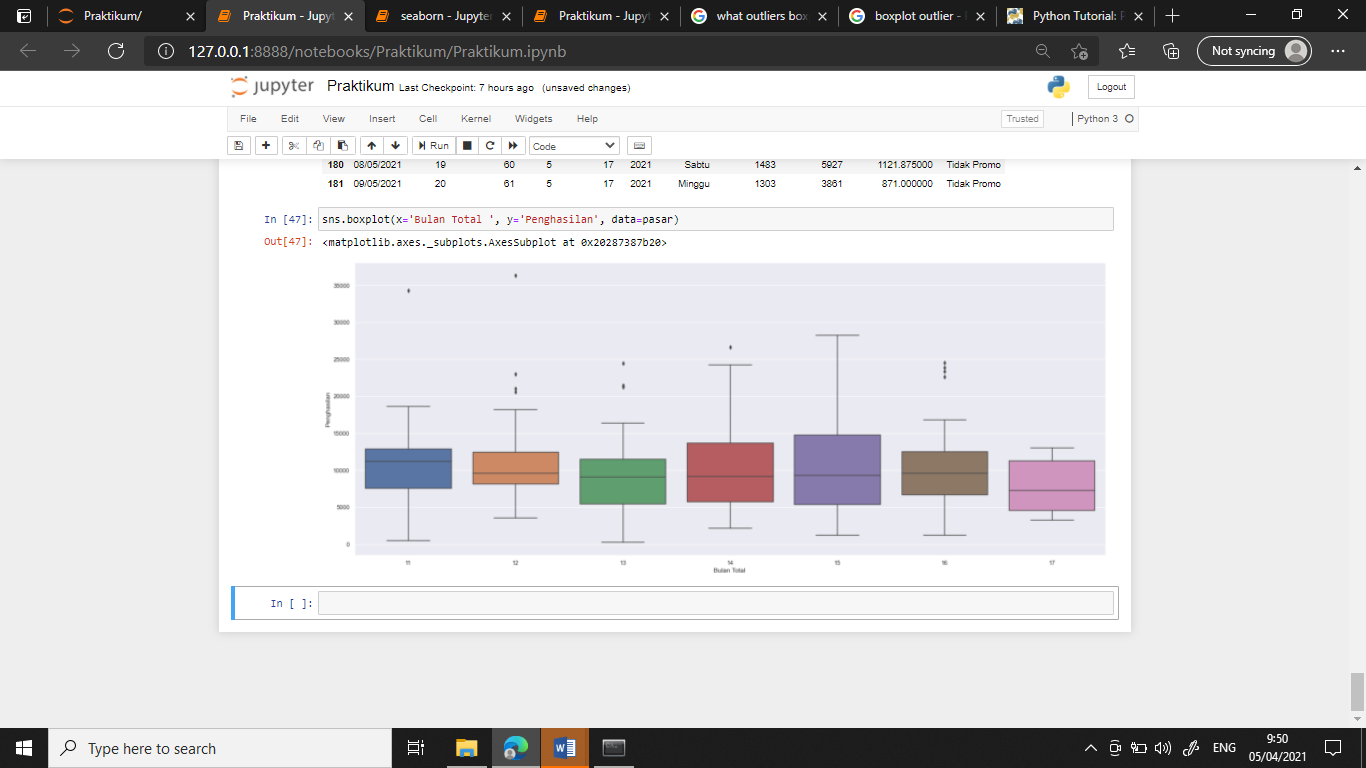


## Plot Box dengan Seaborn

Membuat grafik boxplot yang memuat informasi dari nilai yang sudah di transformasikan ke dalam bentuk array, kemudian menambahkan fungsi median untuk menambah informasi pada grafik.

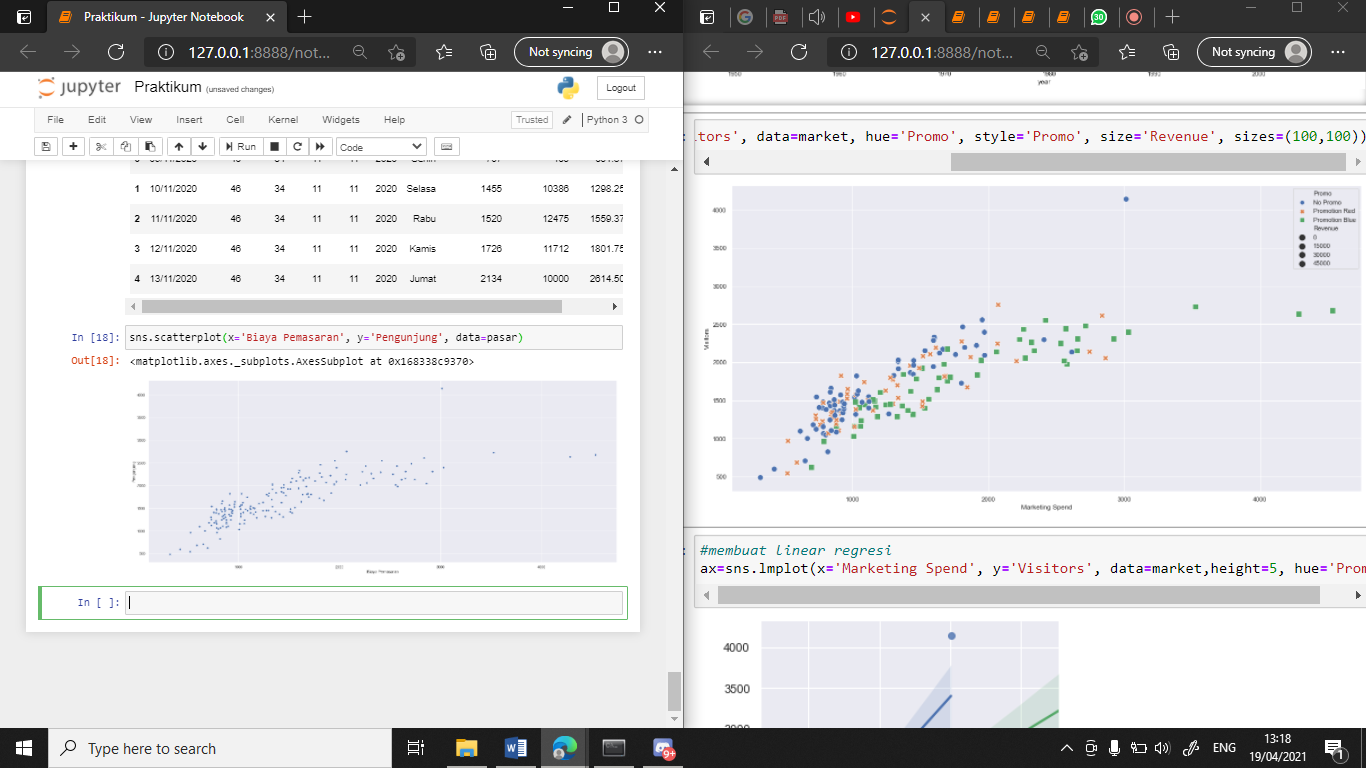


Apabila ingin membuat grafik dengan dua variabel dapat menggunakan fungsi



## Scatterplot dengan Seaborn

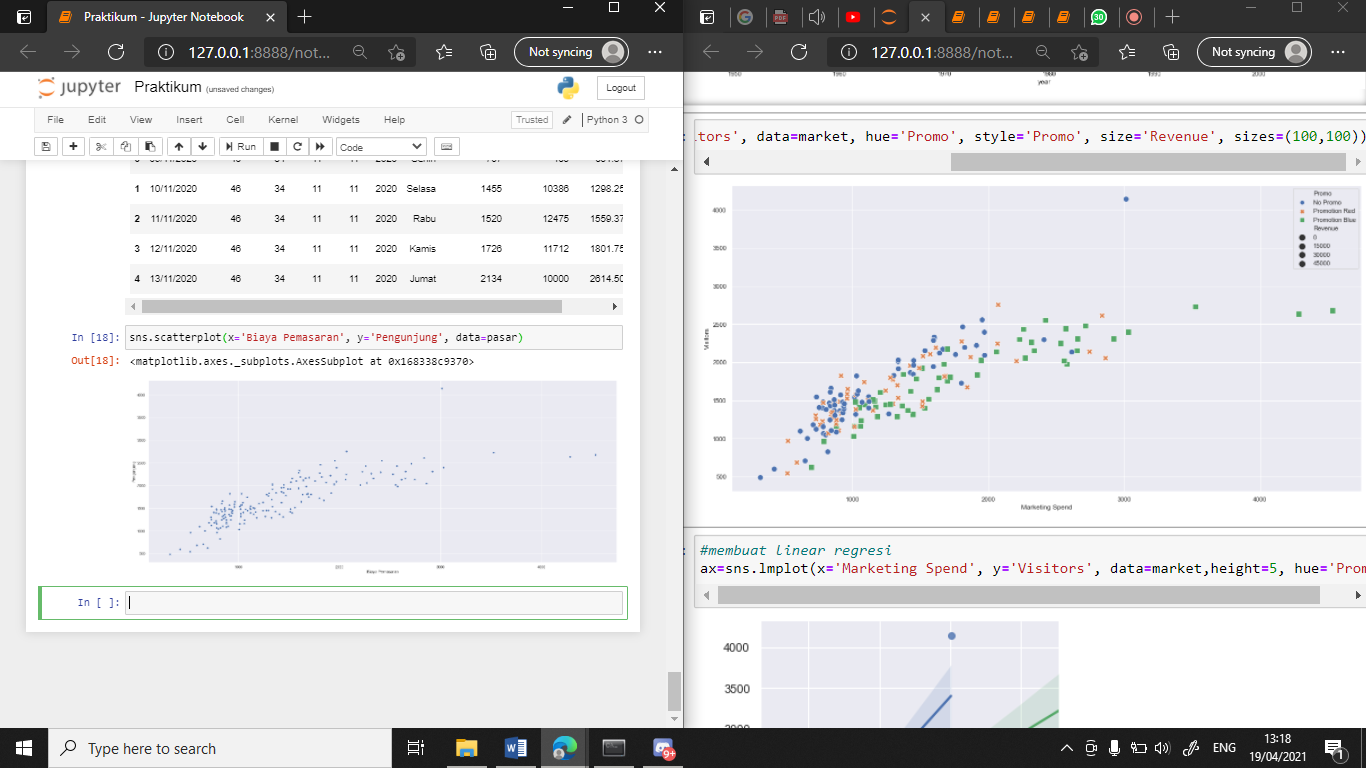
Scatterplot merupakan grafik yang memiliki pola titik-titik dalam merepresentasikan atau memvisualisasikan suatu data

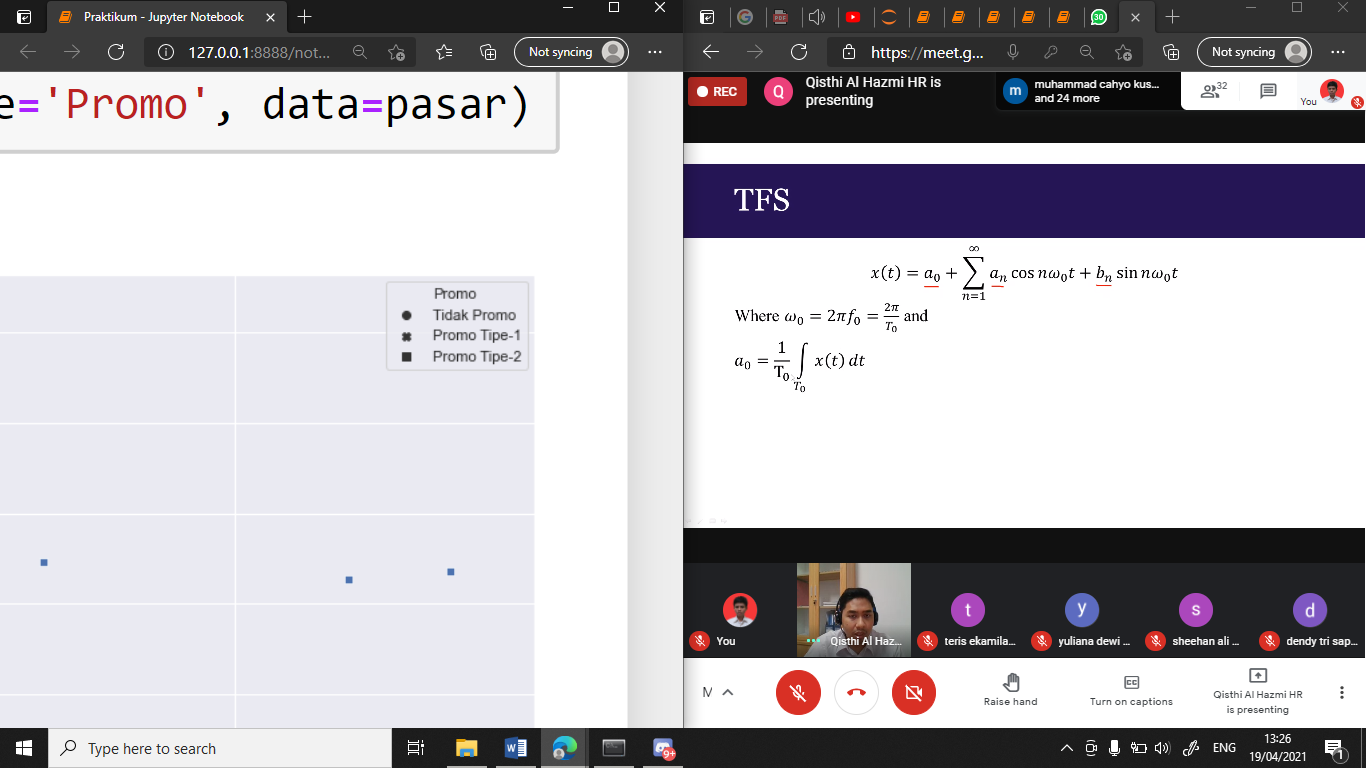
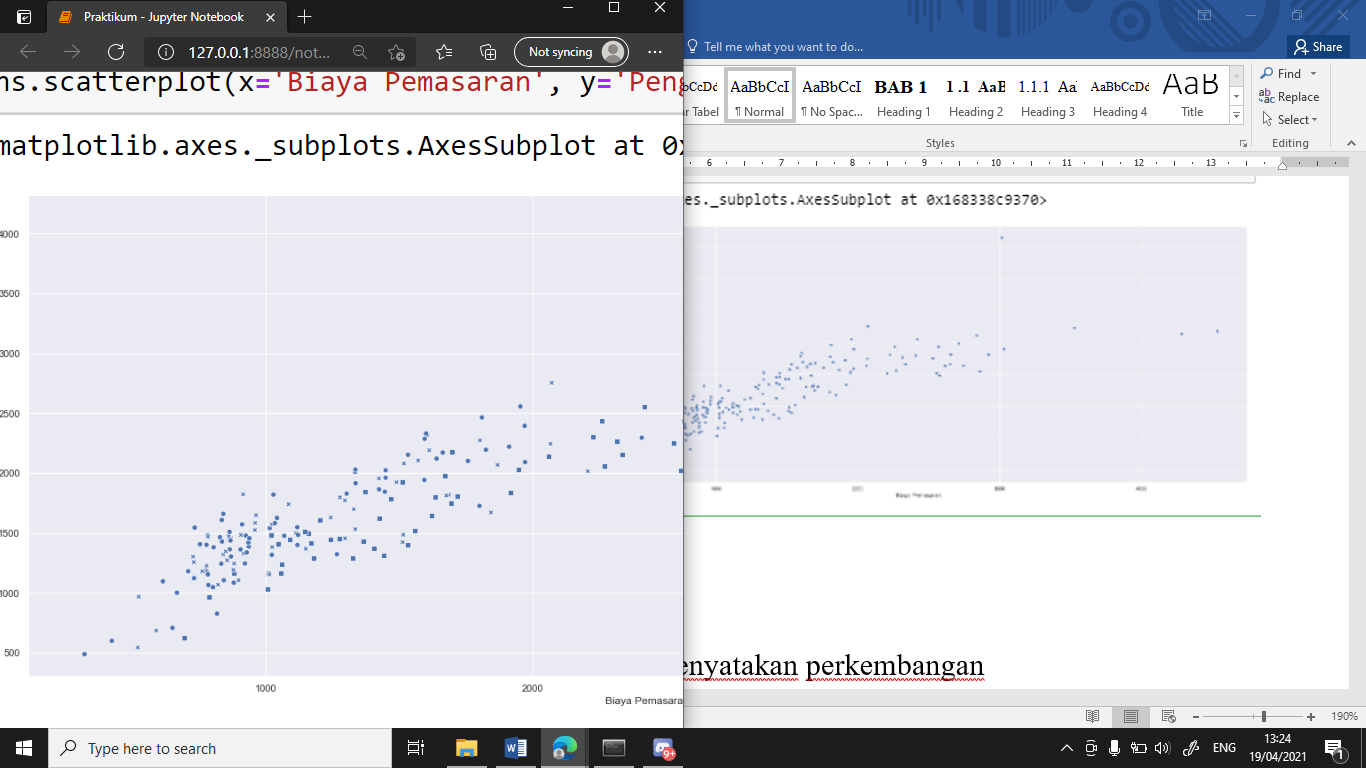


Apabila ingin mengklasifikasikan data menjadi tiga kategori dapat dengan menambahkan fungsi

Style=’String pada Kolom’

Sehingga dapat dilihat, masing-masing data, memiliki bentuk yang berbeda beda seperti bentuk bulat, silang maupun kotak.



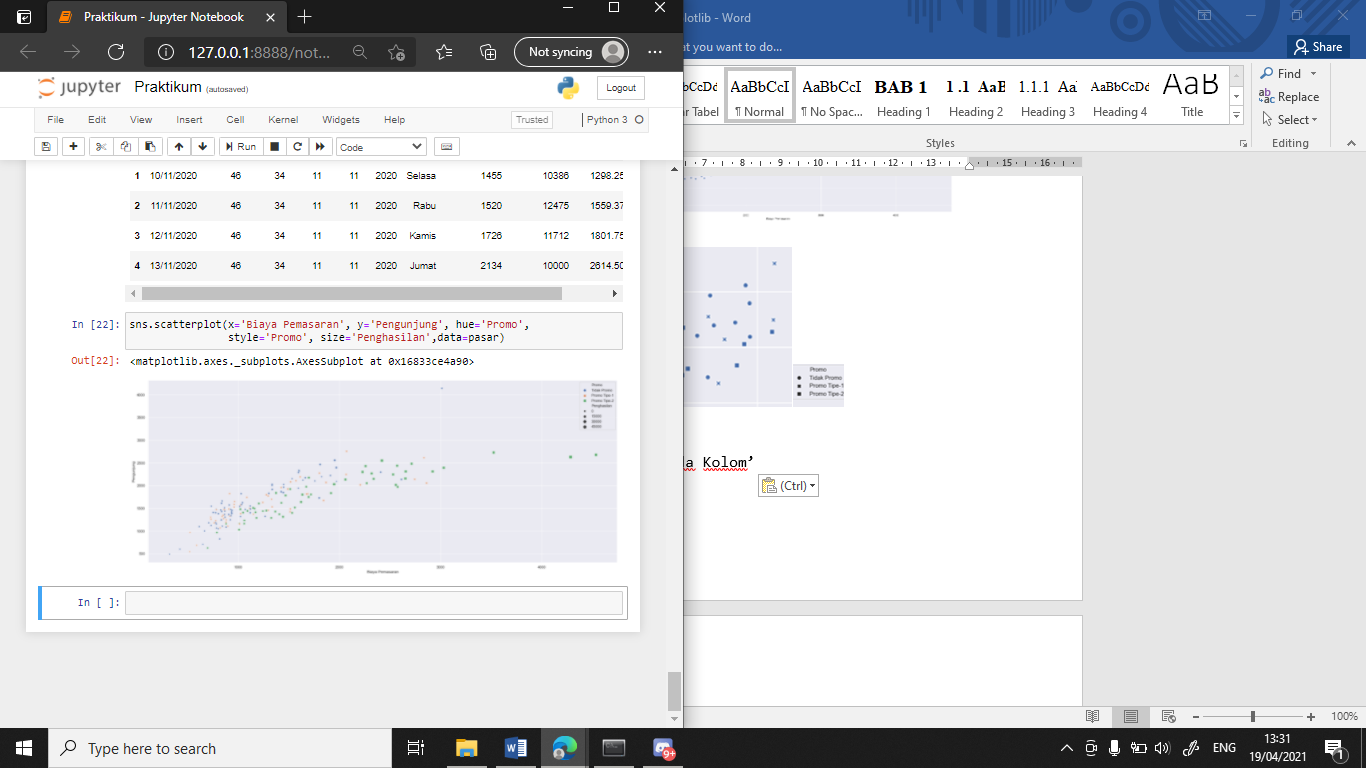


Untuk memberikan warna pada masing-masing kategori dapat menggunakan:

hue=’String pada Kolom’

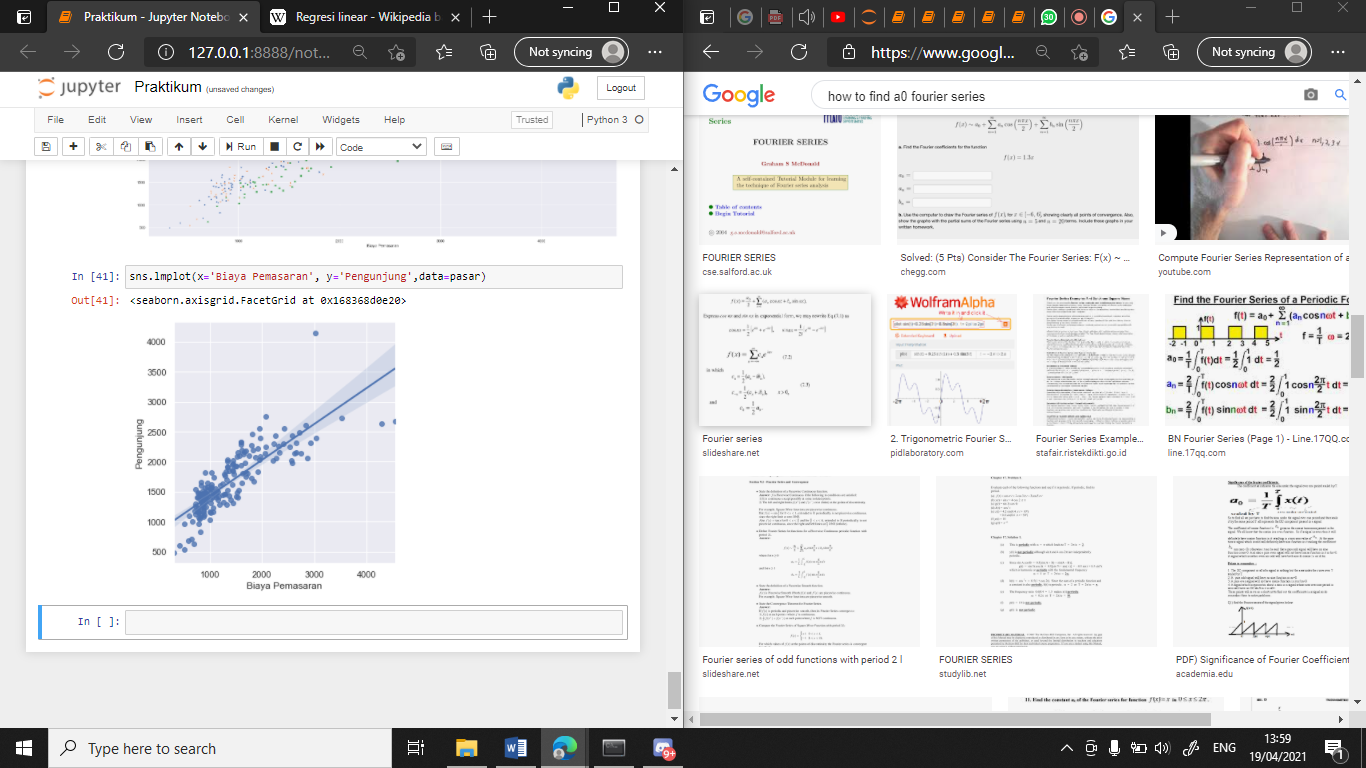
Fungsi Size digunakan untuk memberikan perbedaan pada besar nilai data yang di referensikan, pada contoh dibawah semakin besar Penghasilan didapat maka besar bentuk scatter akan semakin besar terlihat.

size=’String pada Kolom’



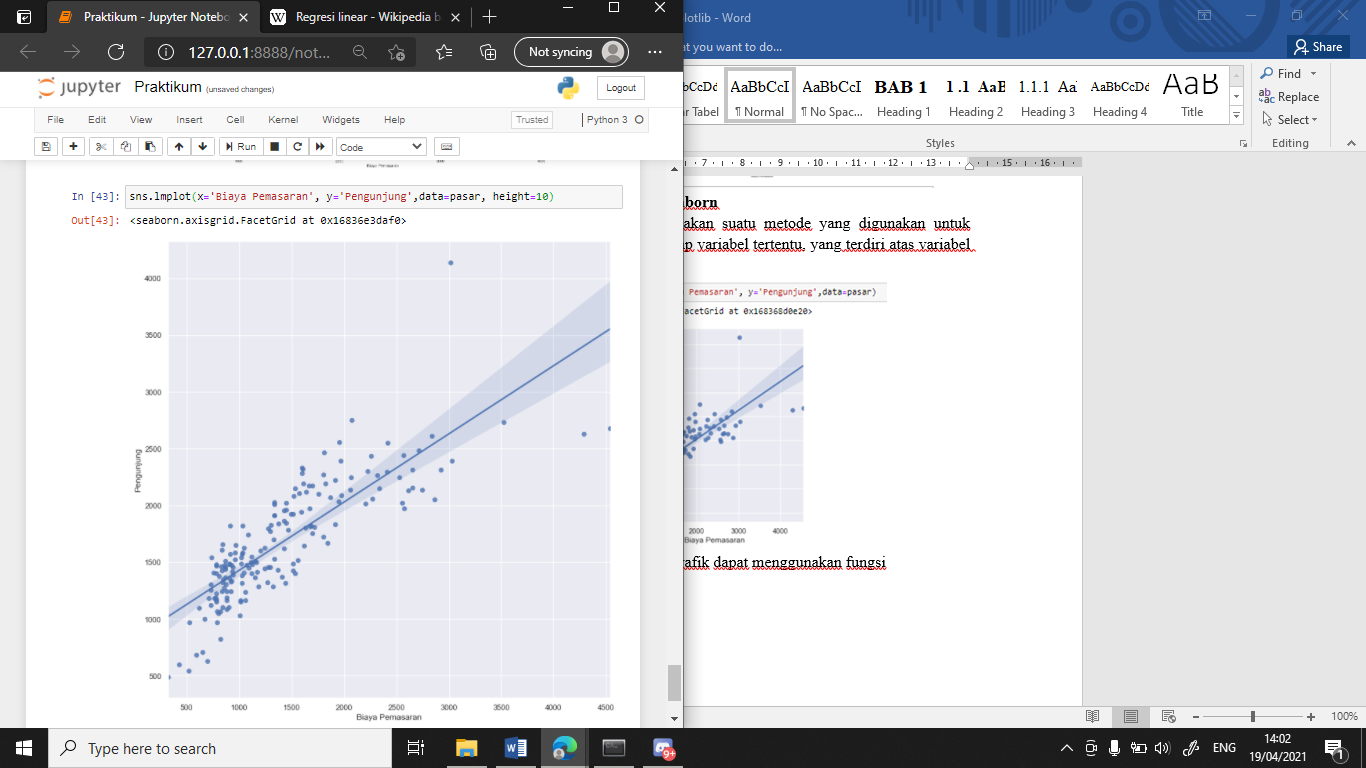
## Linear Regresi dengan Seaborn

Linear Regresi merupakan suatu metode yang digunakan untuk melakukan prediksi terhadap variabel tertentu, yang terdiri atas variabel bebas dan variabel terikat.



Untuk memperbesar grafik dapat menggunakan fungsi

height = Integer



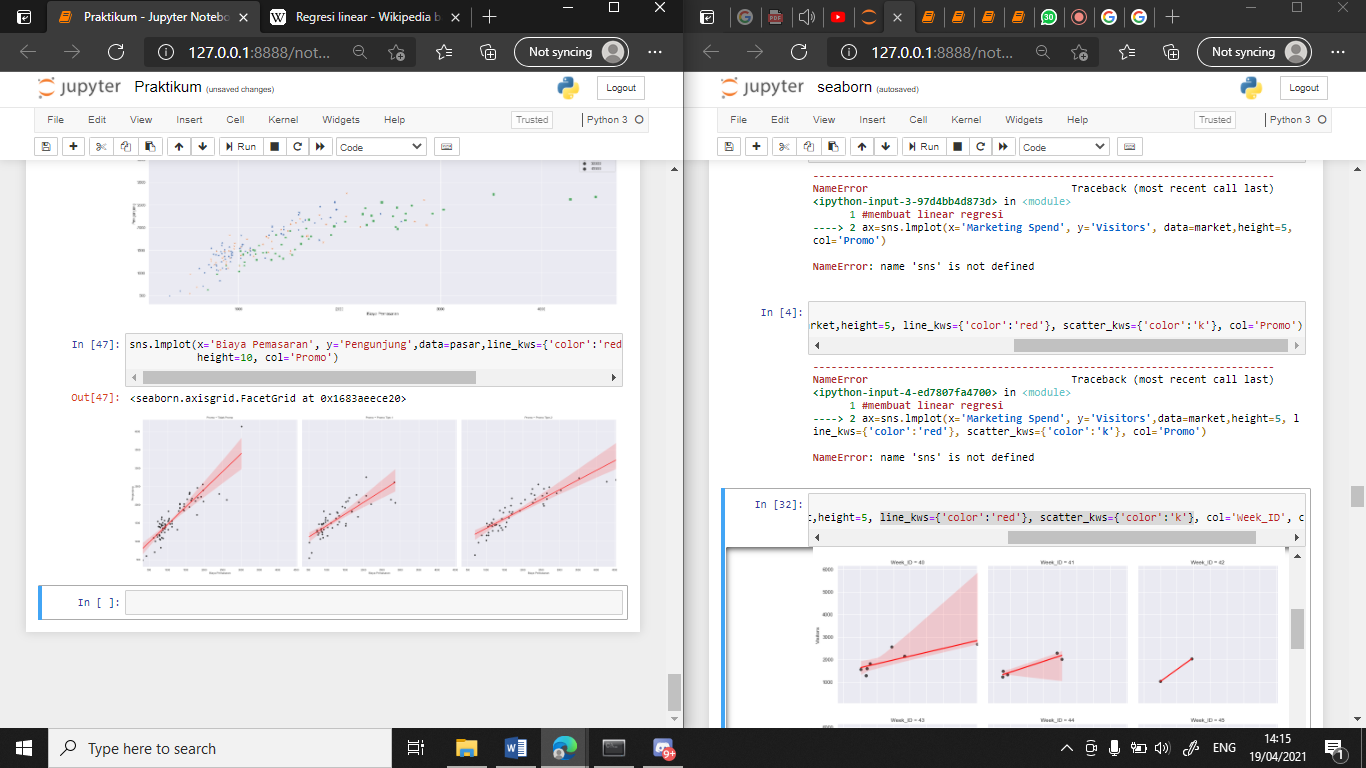
Untuk membagi grafik ke dalam suatu kategori dapat digunakan fungsi  
col=’String dalam kolom’



Untuk mengatur agar mengganti warna pada garis Linear Regresi dan Scatter plot tidak sama, maka dapat menggunakan fungsi

line\_kws={'color':'red'}

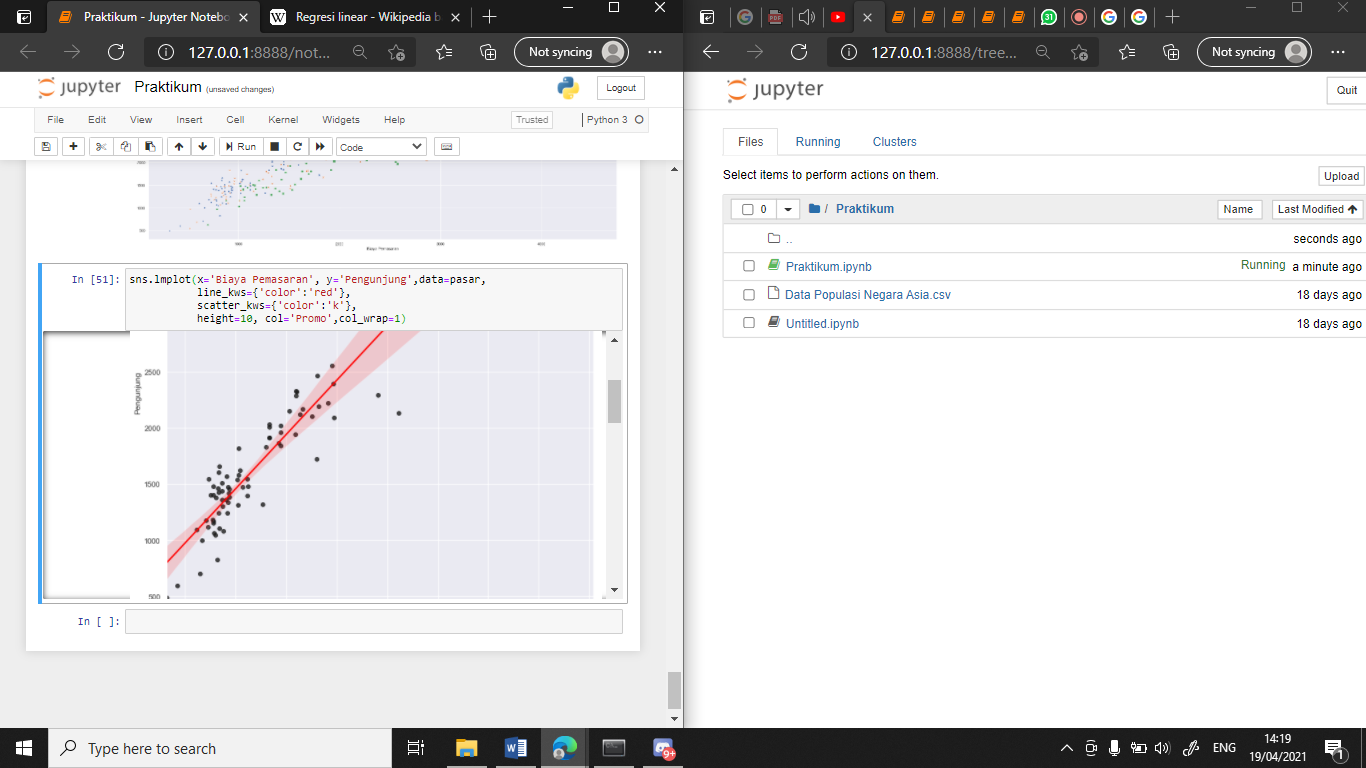
scatter\_kws={'color':'k'}



Untuk mengatur agar berapa jumlah grafik yang dapat ditunjukkan pada

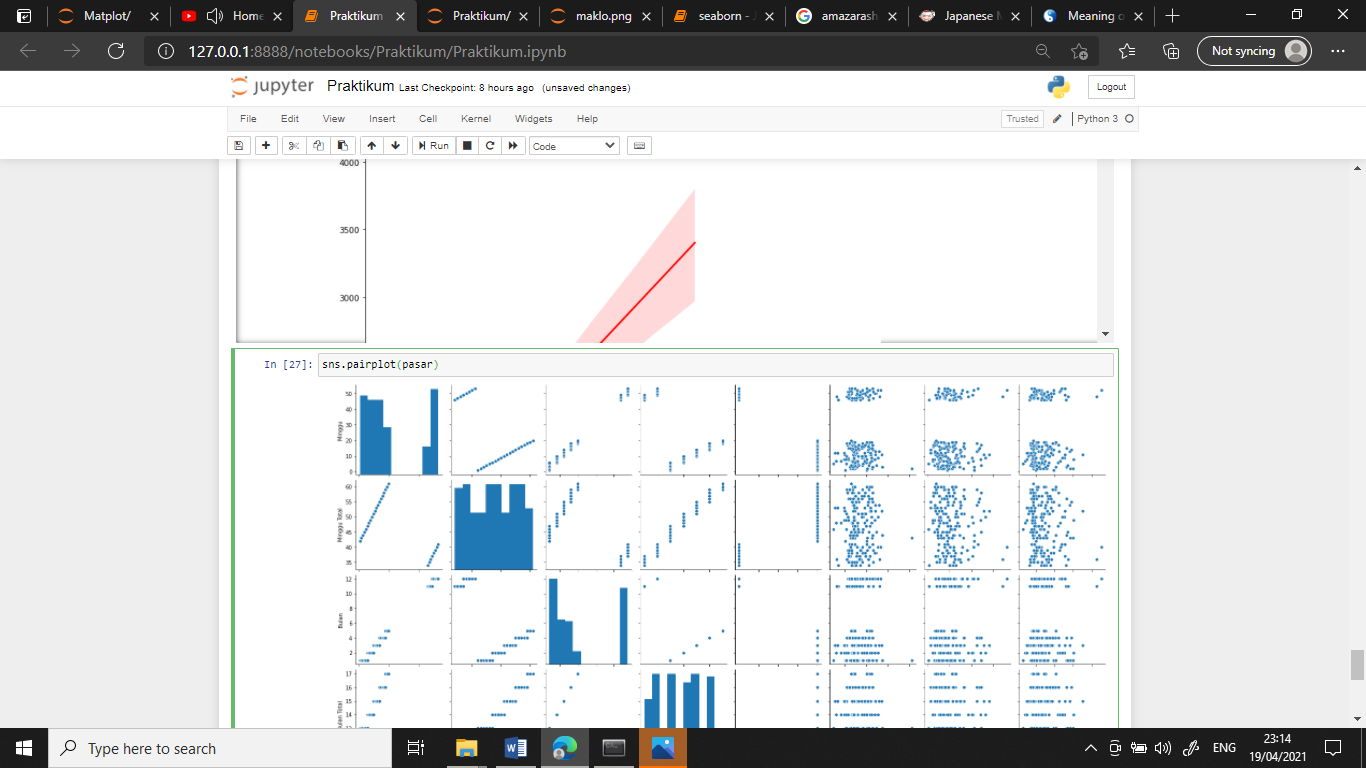
satu baris dapat menggunakan

col\_wrap=Integer



## Pairplot dengan Seaborn

Fungsi pairplot bertujuan untuk menghasilkan grafik terhadap pada seluruh data dengan waktu yang bersamaan.



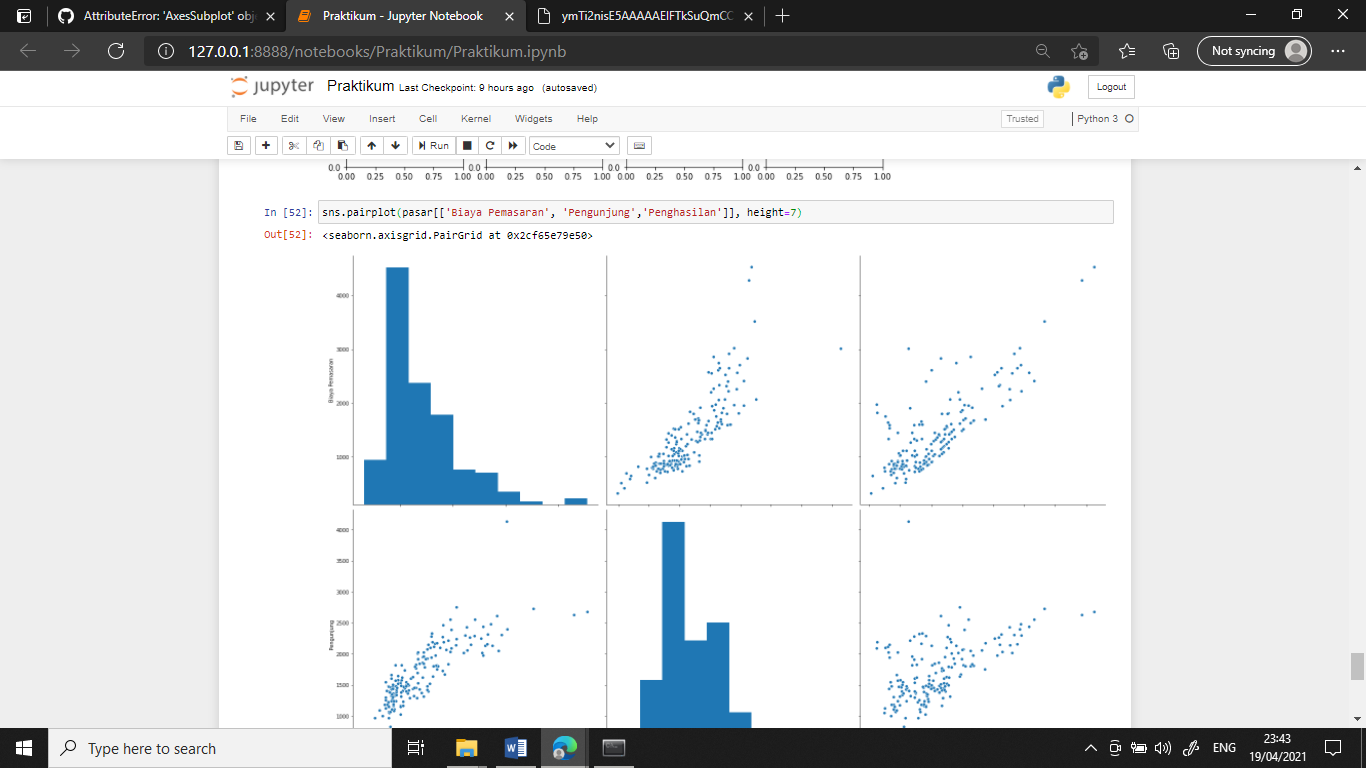
Fungsi tersebut akan menghasilkan grafik yang memvisualisasikan seluruh data dari data yang telah di impor sebelumnya

sns.pairplot(variabel)



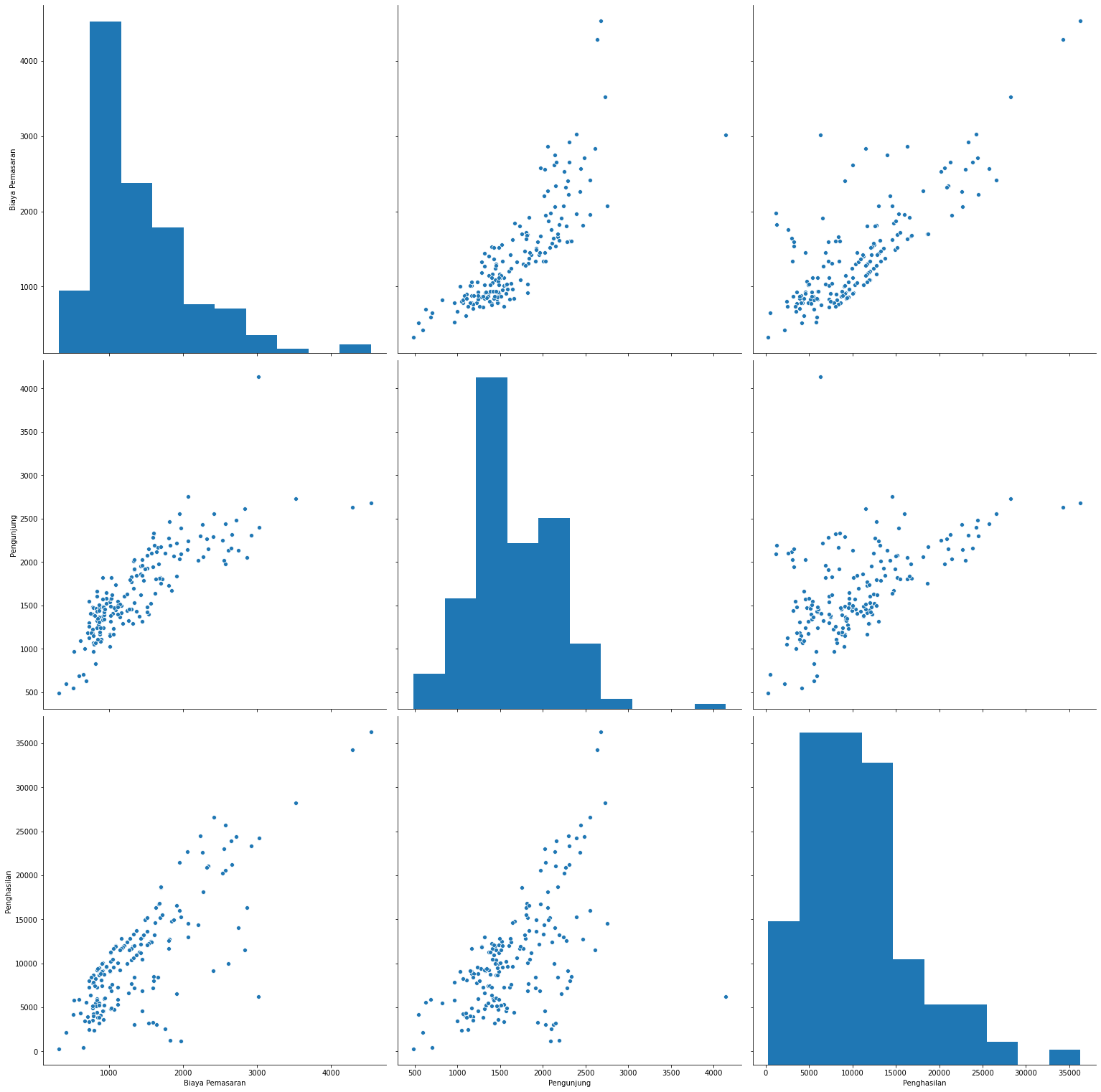
Untuk memilah data tertentu yang ingin di visualisasikan dapat menggunakan fungsi list terhadap data yang ingin dimasukkan

Data\_impor[[‘String pada kolom’]]



Fungsi height digunakan untuk mengubah ukuran dari grafik

height=integer

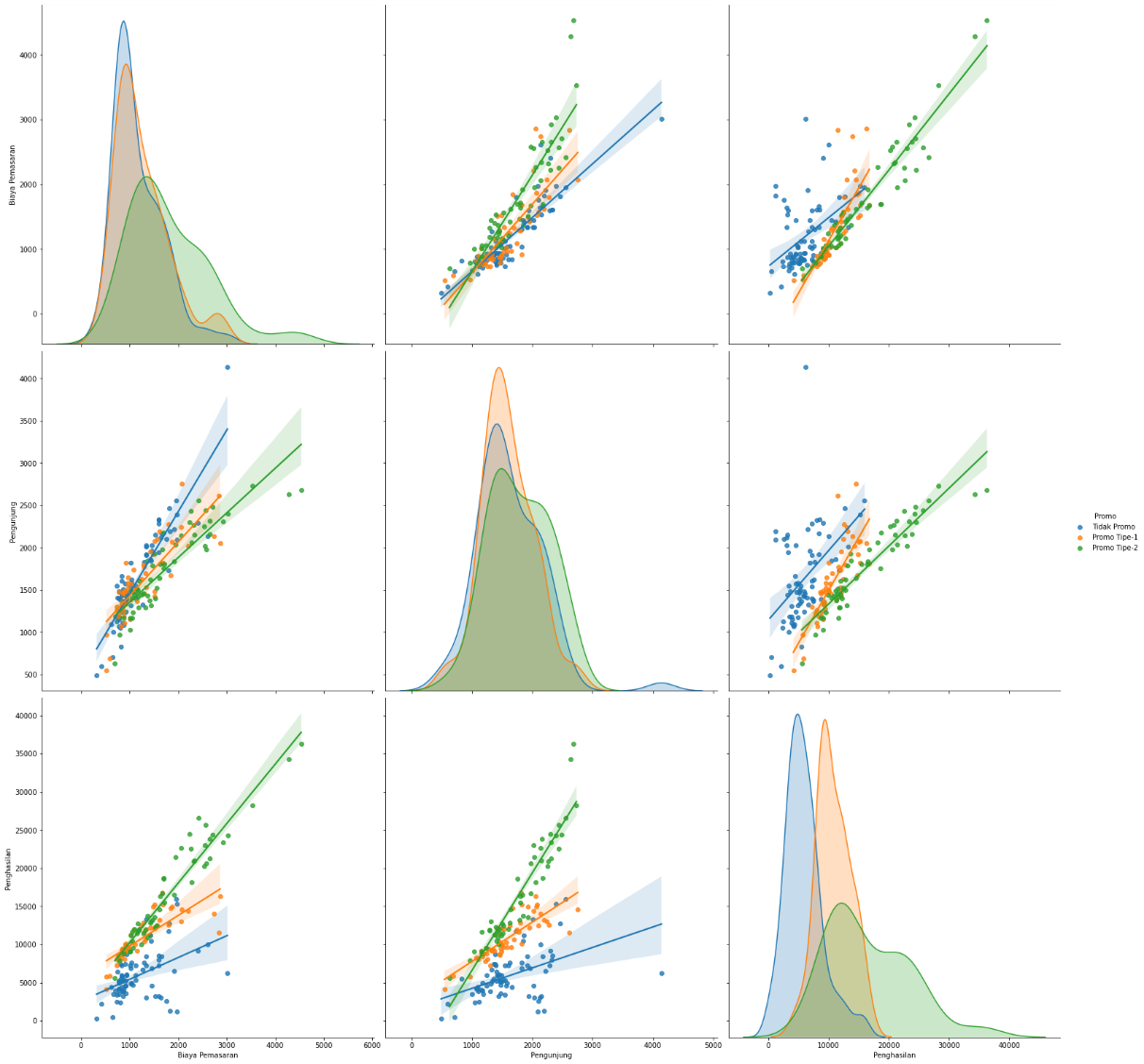


Untuk membagi kedalam beberapa kategori dapat ditambahkan fungsi hue dan menambahkan string yang ingin dimasukkan ke dalam fungsi list, yang akan menampilkan scatterplot pada grafik dua variabel dan grafik distribusi untuk grafik satu variabel



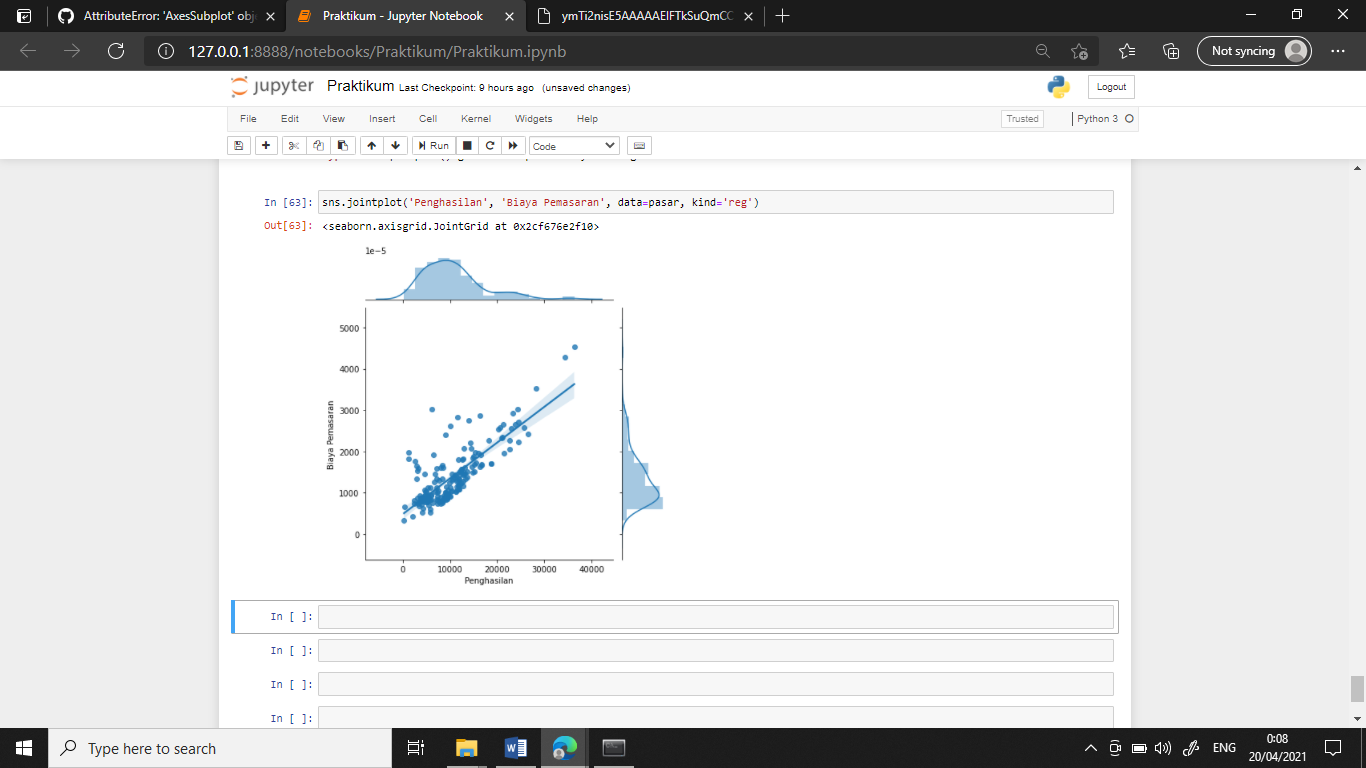
Kind=’reg’

Fungsi diatas digunakan untuk menampilkan ligresi linear pada scatterplot



## Jointplot pada Seaborn

Fungsi Jointpot merupakan fungsi yang menggabungkan grafik distribusi data terhadap masing-masing variabel dengan grafik scatter sehingga analisis dapat mudah dilakukan.



Analisis Praktikum

1. Membuat Dataframe dari fungsi

import matplotlib.pyplot as plt

import numpy as np

# Buatlah suatu interval dari [-2𝝅, 2𝝅] yang dibagi ke dalam 100

x = np.linspace(\_\_\_\_, \_\_\_\_, \_\_\_\_)

# Buatlah suatu fungsi sin(2x + 90)

y = \_\_\_\_\_\_\_\_\_\_

# Buatlah suatu dictionary dataframe

dataframe = {

    "Nilai x " : \_\_\_,

    "f(x)" : \_\_\_\_

}

# Buatlah suatu dataframe menggunakan pandas

df\_fungsi = \_\_\_\_\_\_(\_\_\_\_\_\_)

# Menampilkan tabel dataframe

df\_fungsi

2. Import File Pandas dan DIstribusi Data

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Bacalah data marketing csv

pasar = pd.\_\_\_\_\_(\_\_\_\_\_)

# Mengubah data Penghasilan ke dalam bentuk array

x = pasar['Penghasilan'].array

# Carilah fungsi mean dan median dari data kolom Penghasilan

mean = pasar['Penghasilan'].\_\_\_\_

median = pasar['Penghasilan'].\_\_\_\_

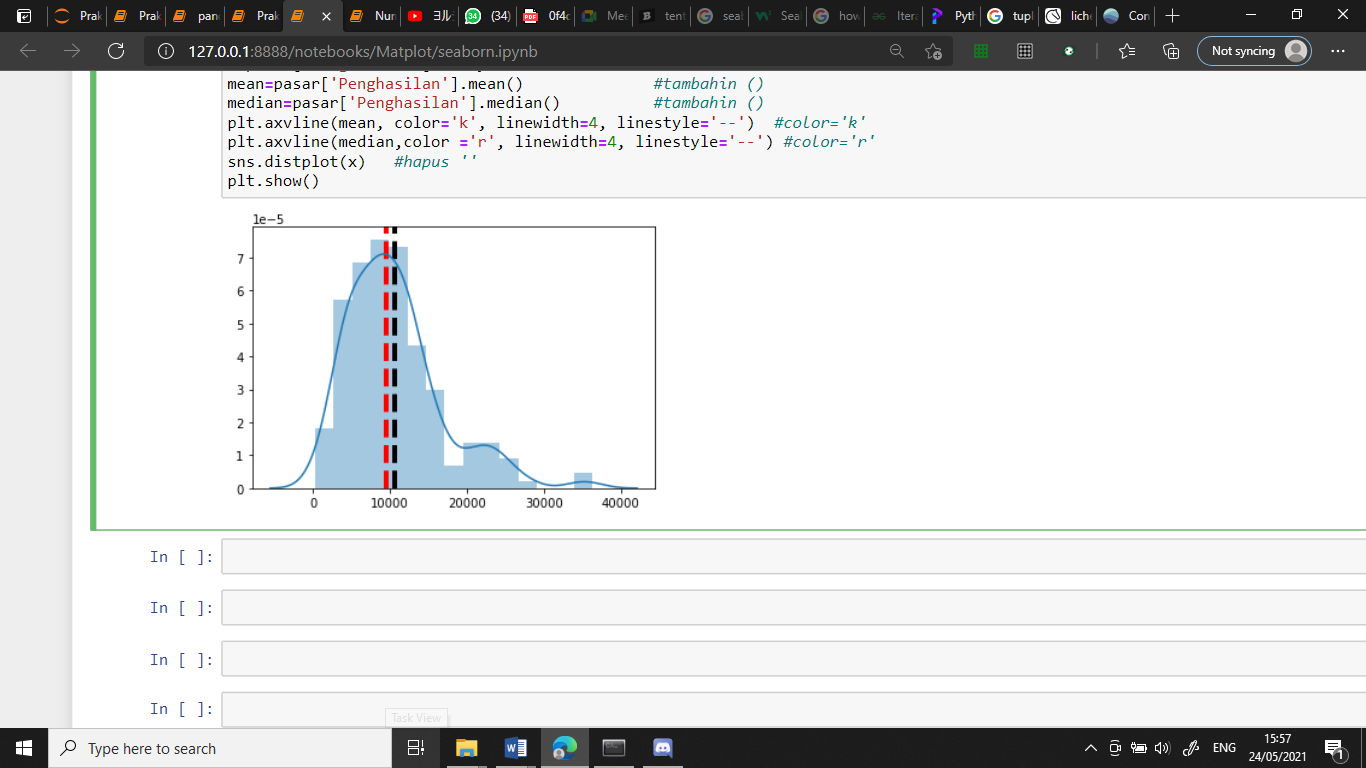
plt.axvline(mean, color = 'k', linewidth = 4, linestyle = '--')

plt.axvline(median, color = 'r', linewidth = 4, linestyle = '--')

# Buatlah tabel distribusi dari data kolom Penghasilan

sns.distplot(\_\_)

plt.show



3. Penyusunan Indeks pada Grafik

import seaborn as sns

import pandas as pd

import matplotlib.pyplot as plt

pasar = pd.read\_csv('Data Marketing.csv')

# Membuat suatu multi-plot 2x2

fig, axis = plt.subplots(2,2, figsize = (12, 7))

# Buatlah grafik boxplot

sns.\_\_\_( y = 'Penghasilan', x = 'Nama Hari', data = \_\_\_, ax = axis[0,0])

# Buatlah grafik lineplot

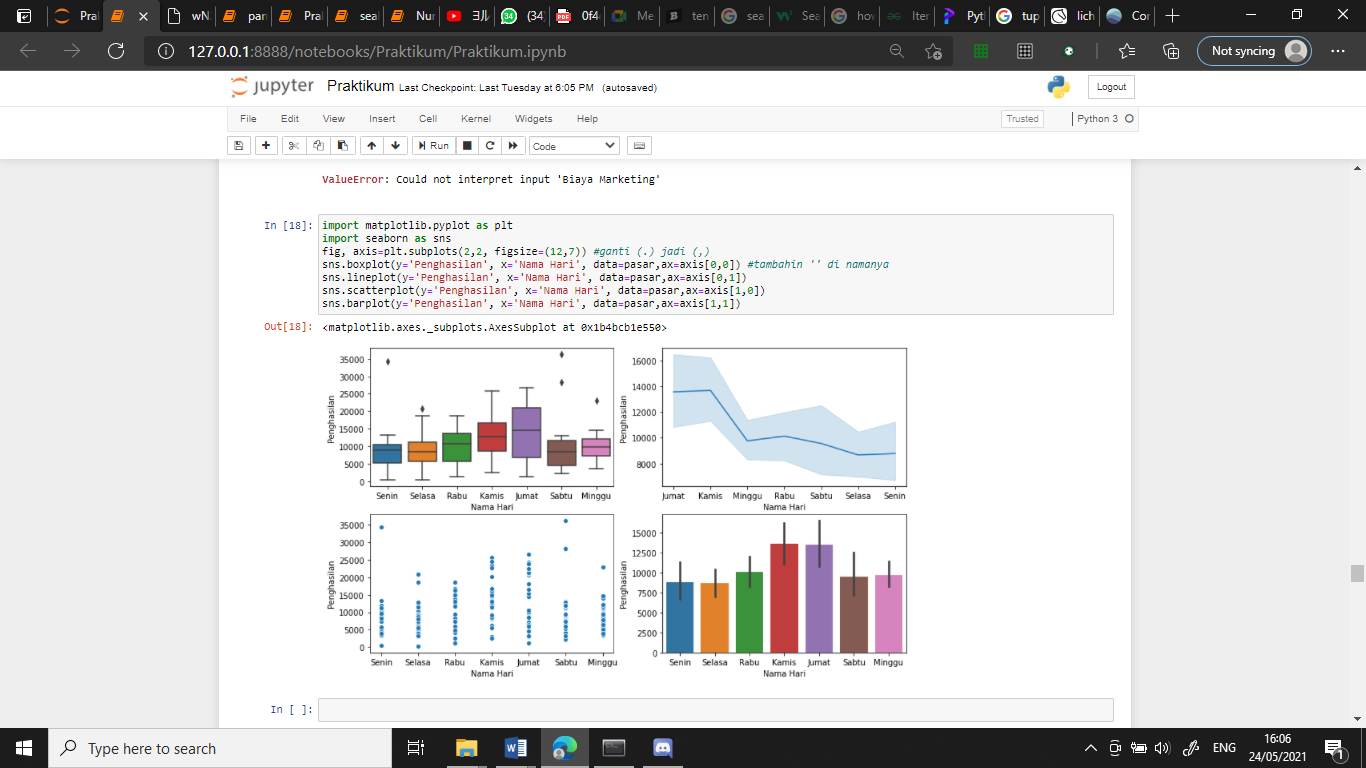
sns.\_\_\_( y = 'Penghasilan', x = 'Nama Hari', data = \_\_\_, ax = axis[0,1])

# Buatlah grafik Scatterplot

sns.\_\_\_( y = 'Penghasilan', x = 'Nama Hari', data = \_\_\_, ax = axis[1,0])

# Buatlah grafik Barplot

sns.\_\_\_( y = 'Penghasilan', x = 'Nama Hari', data = \_\_\_, ax = axis[1,1])

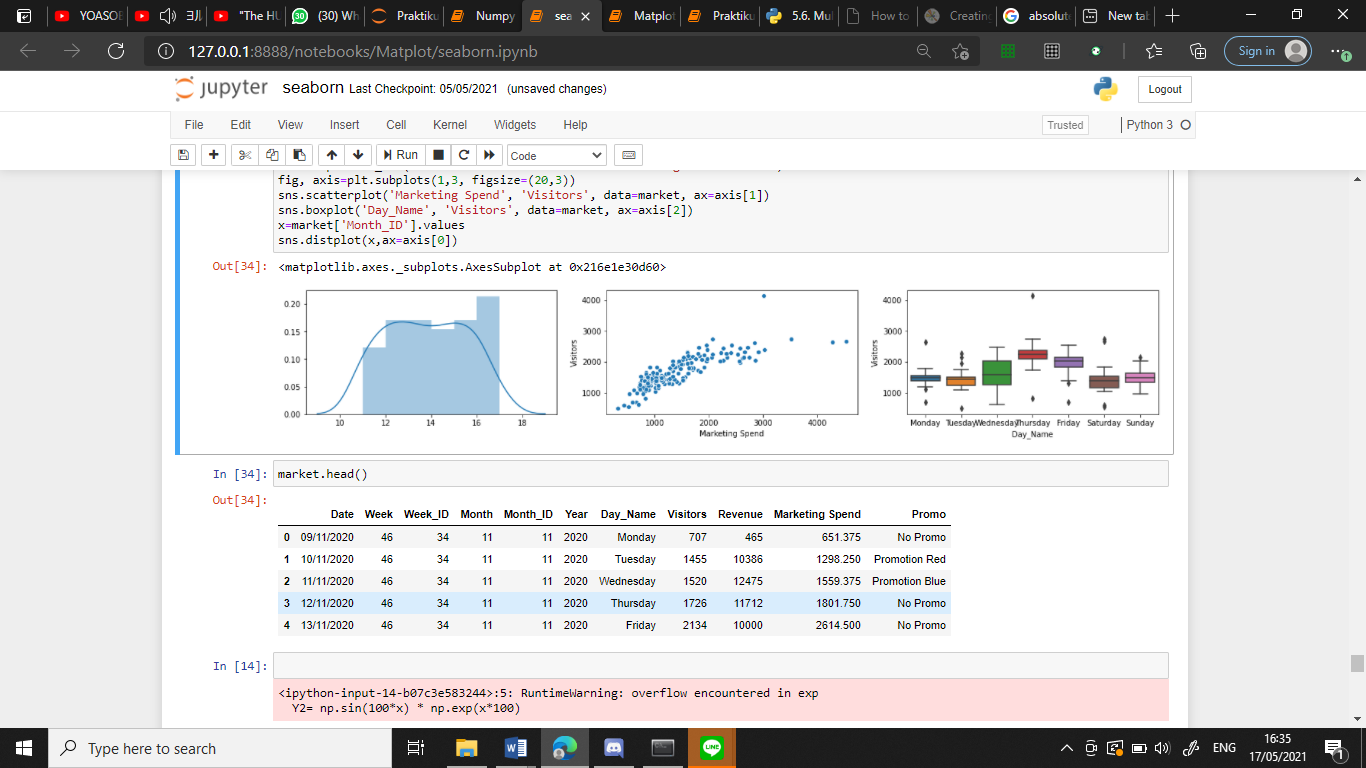


Tugas Akhir:

(Program beserta gambar keluaran dilampirkan pada saat pengumpulan tugas akhir)

1. Carilah sebuah data dalam format csv di Internet yang memiliki minimal 3 kolom data dan 10 baris data, lalu buatlah plot scatter, plot box dan plot distribusi terhadap satu data yang dipilih pada satu gambar menggunakan indeks pada satu keluaran!

Contoh Output:



2. Carilah sebuah data dalam format csv di Internet yang memiliki minimal 3 kolom data dan 10 baris data, dan lakukanlah analisis pada setiap kategori dari salah satu kolom, dan carilah nilai mean, median, dan std dari kategori tersebut, kemudian buatlah *Dataframe* dari nilai mean, median, dan std.

Contoh :

Menggunakan Data Marketing.csv



