**LAPORAN DATA WRANGLING C**

**A DEEP DIVE INTO DATA WRANGLING WITH PYTHON**



**DISUSUN OLEH:**

Mohamad Ibnu Fajar Maulana (21083010106)

**DOSEN PENGAMPU:**

KARTIKA MAULIDA HINDRAYANI, S.Kom, M.Kom

**PROGRAM STUDI SAINS DATA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”**

**JAWA TIMUR**

**2022**

# DAFTAR ISI

[DAFTAR ISI i](#_Toc130760045)

[**T04 - A Deep Dive into Data Wrangling with Python** 1](#_Toc130760046)

[Input-Output: 1](#_Toc130760047)

[1. Load Dataset 1](#_Toc130760048)

[2. Menemukan Missing Value 2](#_Toc130760049)

[3. Subsetting DataFrame 4](#_Toc130760050)

[4. Outliers 6](#_Toc130760051)

[5. Fungsi GroupBy 9](#_Toc130760052)

# **T04 - A Deep Dive into Data Wrangling with Python**

**106\_Mohamad Ibnu   
Fajar Maulana**

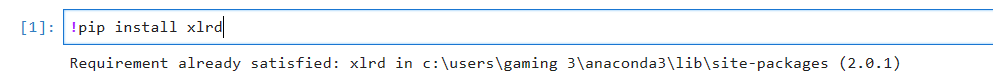
1. Load dataset yang kamu punya boleh sumbernya dari Kaggle dll
2. Temukan missing values pada dataset tersebut
3. Buat kolom baru dengan menggunakan teknik subsetting
4. Temukan outliers pada dataset tersebut
5. Gunakan fungsi groupBy

## Input-Output:

### Load Dataset

Dataset yang digunakan pada laporan ini mengangkat permasalahan harga penutupan saham dan nilai jualnya, data tersebut diambil dari website:

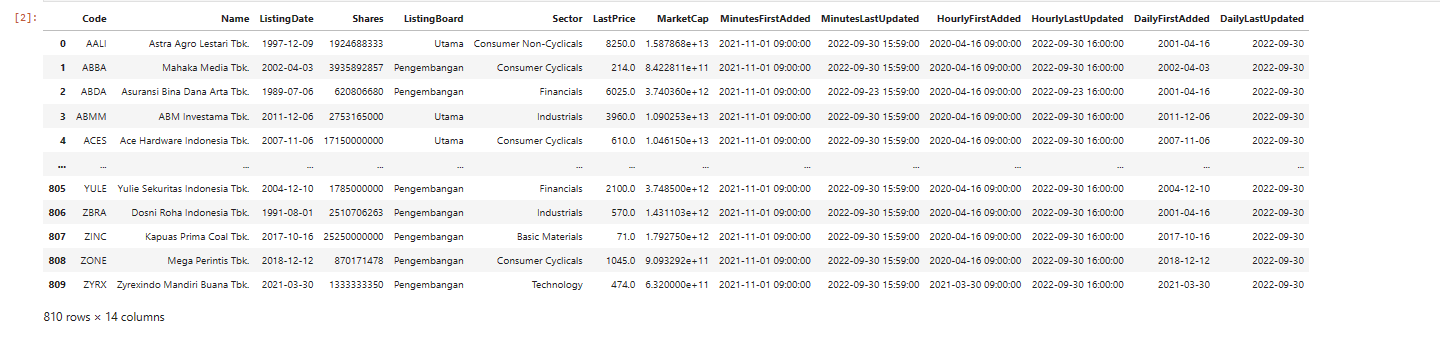
<https://www.kaggle.com/code/haidhiangkawijana/prediksi-saham-lstm/notebook>



* Pada Gambar diatas jika tipe file xls bisa menggunakan *!pip install xlrd*, akan tetapi dataset yang saya gunakan memiliki tipe file csv, berikut caranya:



* Gambar ditas kita menggunakan import library python yakni *import pandas as pd* guna menampilkan sebuah DataFrame(csv). Kemudian data tersebut menghasilkan output:



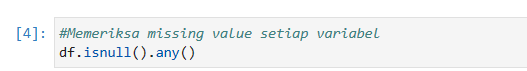
**106\_Mohamad Ibnu   
Fajar Maulana**

* Pada gambar diatas merupakan DataFrame saham-saham yang ada di Indonesia, dengan 810 baris dan 14 kolom data.

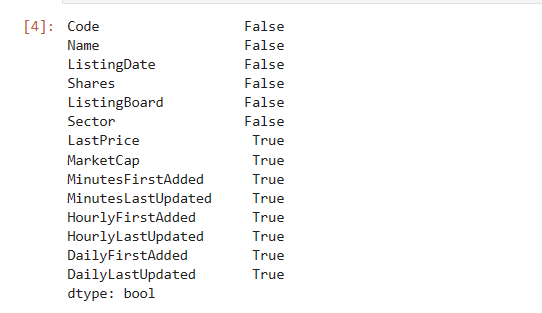
**106\_Mohamad Ibnu   
Fajar Maulana**

### Menemukan Missing Value

* Untuk menemukan missing value kita dapat melakukan kodingan seperti gambar dibawah ini:



* Gambar diatas menggunakan method *df.isnull().any()* yang merupakan sebuah method pada objek pandas DataFrame yang digunakan untuk memeriksa apakah terdapat nilai null atau kosong pada setiap kolom DataFrame. Kemudian mendapatkan output sebagai berikut:



* Gambar diatas terdapat True-False yang artinya menentukan kolom mana yang terdapat *missing value*. Kolom yang dilabeli *True* artinya terdapat *missing value*
* Penjelasan lebih lanjut alangkah baiknya melakukan kode seperti dibawah ini:



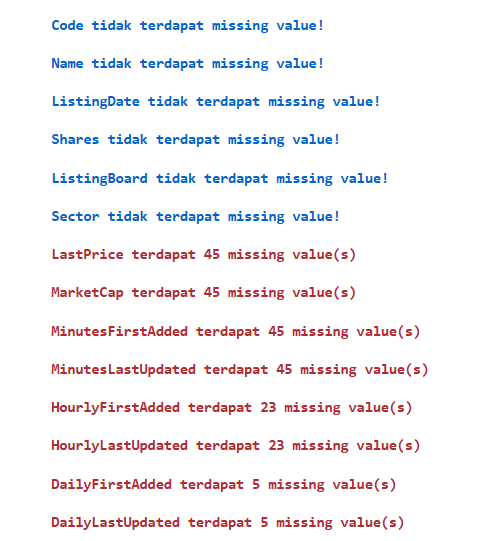
* Gambar tersebut merupakan kode yang mengandung sebuah loop yang digunakan untuk memeriksa apakah terdapat nilai null atau kosong pada setiap kolom DataFrame yang disimpan pada variabel df.
* Pertama, kode mengambil setiap nama kolom dari DataFrame menggunakan fungsi df.columns.

**106\_Mohamad Ibnu   
Fajar Maulana**

* Kemudian, kode memeriksa jumlah nilai null pada setiap kolom dengan menggunakan fungsi df[i].isnull().sum(). Fungsi ini menghitung jumlah nilai null pada kolom yang sedang diperiksa.
* Jika jumlah nilai null pada kolom tersebut lebih besar dari nol, maka kode akan mencetak pesan yang menunjukkan bahwa terdapat nilai null pada kolom tersebut menggunakan format string dan warna merah (dengan escape sequence \033[1m dan \33[31m). Pesan ini akan mencantumkan nama kolom dan jumlah nilai null pada kolom tersebut.
* Jika jumlah nilai null pada kolom tersebut sama dengan nol, maka kode akan mencetak pesan yang menunjukkan bahwa tidak ada nilai null pada kolom tersebut menggunakan format string dan warna biru (dengan escape sequence \033[1m dan \33[34m).

Dengan menggunakan loop ini, kita dapat dengan mudah memeriksa setiap kolom pada DataFrame untuk mengetahui apakah terdapat nilai null pada kolom tersebut.

* Kemudian mendapatkan output:

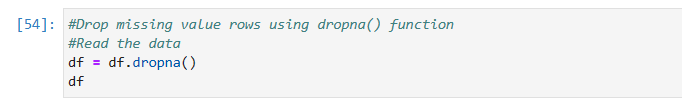


* Gambar diatas mengintepretasikan kolom-kolom yang terdapat *missing value* (berwarna merah) dan kolom-kolom yang tidak terdapat *missing value*(berwarna biru) seperti penjelasan sebelumnya, silahkan simak baik-baik hasil outputnya dan laporan ini.

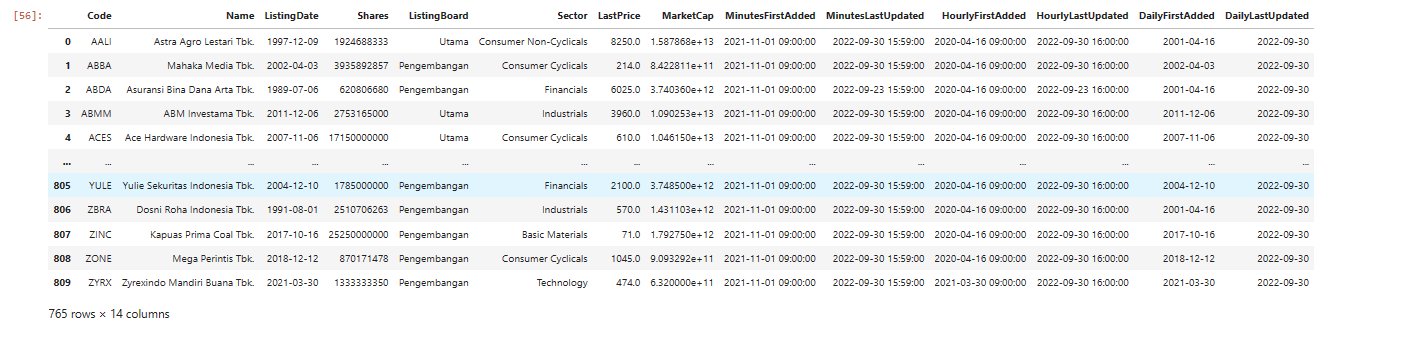
**106\_Mohamad Ibnu   
Fajar Maulana**

#### Mengatasi Missing Value

* Untuk mengatasi *missing value* lakukan kodingan dibawah ini:



* Pada gambar diatas menggunakan *df.dropna()* adalah sebuah method pada objek pandas DataFrame yang digunakan untuk menghapus baris atau kolom yang mengandung nilai null atau kosong. Method ini akan mengembalikan DataFrame baru dengan nilai null dihapus. Kemudian mendapatkan output:



* Gambar diatas sudah menghapus beberapa baris yang terdapat *missing value* yang tadinya berjumlah 810 baris dan 14 kolom menjadi 765 baris dan 14 kolom

### Subsetting DataFrame

* Lakukan kodingan seperti dibawah ini untuk melakukan subsetting DataFrame saham:



* Gambar diatas odingan tersebut merupakan sebuah kode untuk membuat DataFrame baru yang terdiri dari lima baris pertama dari DataFrame df dan kolom-kolom tertentu yaitu 'Code', 'Name', 'ListingDate', 'Shares', 'LastPrice', dan 'MarketCap'.

Secara detail, kode tersebut melakukan beberapa hal sebagai berikut:

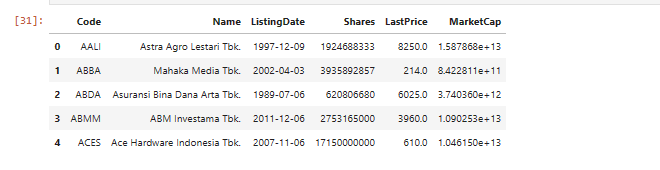
* *df.loc*: method untuk melakukan indexing DataFrame dengan label (misalnya nama kolom atau nama indeks).

**106\_Mohamad Ibnu   
Fajar Maulana**

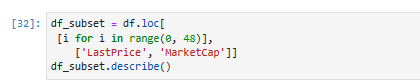
* *[i for i in range(0,5)]*: sebuah list comprehension yang digunakan untuk membuat list dari nilai indeks baris yang akan dipilih. Pada contoh ini, list tersebut berisi nilai indeks baris dari 0 sampai dengan 4, yaitu lima baris pertama dari DataFrame df.

**106\_Mohamad Ibnu   
Fajar Maulana**

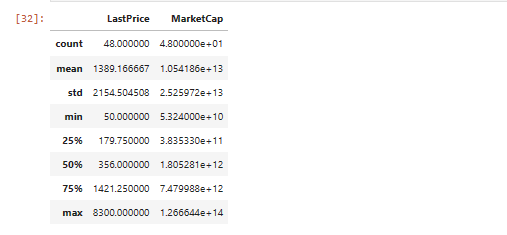
* *['Code','Name','ListingDate','Shares','LastPrice', 'MarketCap']*: list yang berisi nama-nama kolom yang akan dipilih. Hanya kolom dengan nama yang tercantum dalam list ini yang akan diambil untuk DataFrame baru.
* *df\_subset*: variabel yang digunakan untuk menyimpan DataFrame baru yang terdiri dari lima baris pertama dan kolom-kolom tertentu dari DataFrame.
* Kemudian mendapatkan output:



* Gambar diatas merupakan data baru dan hasil dari teknik subsetting.
* Kemudian buat data baru kembali untuk mendeskripsikan data harga penutupan saham dan nilai market saham perusahaan dengan code *df.describe*, seperti gambar dibawah ini:



* Kemudian mendapatkan output:

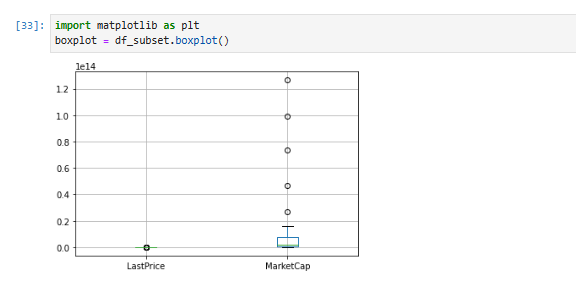


* Gambar diatas dengan teknik subsetting sehingga jadi data baru dan juga mendiskripsikan secara statistik pada kolom *LastPrice*(harga penutupan saham) dan kolom *MarketCap*(nilai saham pada pasar).

**106\_Mohamad Ibnu   
Fajar Maulana**

* Kemudian perbandingan 2 kolom tersebut dapat divisualisasikan boxplot akan tetapi, tetap menggunakan teknik subsetting, dengan kodingan dibawah ini:

**106\_Mohamad Ibnu   
Fajar Maulana**



* Gambar diatas merupakan *input-output* untuk memvisualisasikan perbandingan harga penutupan saham dan nilai saham pada pasar.

### Outliers

* Lakukan kodingan dibawah ini untuk menemukan outliers pada kolom *MarketCap*:



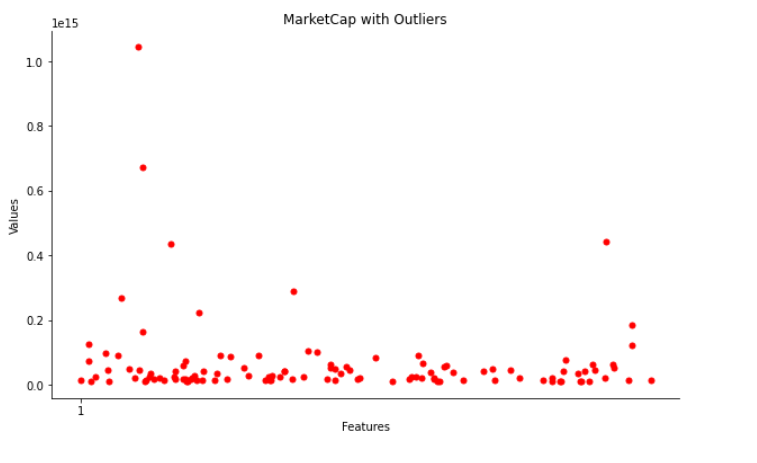
* Kode di atas digunakan untuk melakukan deteksi dan visualisasi outlier pada data pasar saham. Secara detail, kode tersebut melakukan beberapa hal sebagai berikut:
* *import pandas as pd dan import matplotlib.pyplot as plt*: Mengimport modul pandas dan matplotlib untuk membaca data dan membuat visualisasi.
* *df = pd.read\_csv('DaftarSaham.csv')*: Membaca data dari file CSV yang bernama DaftarSaham.csv dan menyimpannya ke dalam DataFrame df.
* *column = 'MarketCap'*: Mengambil kolom MarketCap sebagai kolom yang akan diperiksa terhadap outlier.

**106\_Mohamad Ibnu   
Fajar Maulana**

* *q1 = df[column].quantile(0.25), q3 = df[column].quantile(0.75), dan iqr = q3 - q1*: Menghitung nilai kuartil pertama (Q1), kuartil ketiga (Q3), dan interquartile range (IQR) pada kolom MarketCap.

**106\_Mohamad Ibnu   
Fajar Maulana**

* *lower\_limit = q1 - 1.5 \* iqr dan upper\_limit = q3 + 1.5 \* iqr*: Menghitung nilai batas bawah (lower limit) dan batas atas (upper limit) untuk mendeteksi nilai yang dianggap sebagai outlier pada kolom MarketCap.
* *outliers = df[(df[column] < lower\_limit) | (df[column] > upper\_limit)]*: Mencari nilai-nilai yang dianggap sebagai outlier pada kolom MarketCap.
* *fig, ax = plt.subplots(figsize=(10, 6)) dan ax.boxplot(df[column], showfliers=True)*: Membuat boxplot dari kolom MarketCap yang menampilkan nilai outlier.
* *ax.plot(list(outliers.index), outliers[column], 'ro', markersize=5)*: Menandai nilai-nilai outlier dengan tanda titik merah pada boxplot.
* *ax.set\_title("Boxplot with Outliers"), ax.set\_xlabel("Features"), dan ax.set\_ylabel("Values")*: Memberikan judul dan label pada sumbu-sumbu grafik.
* *ax.spines['top'].set\_visible(False) dan ax.spines['right'].set\_visible(False)*: Menghilangkan sumbu atas dan kanan pada grafik.
* *plt.show()*: Menampilkan boxplot yang telah dibuat.
* Kemudian mendapatkan output:

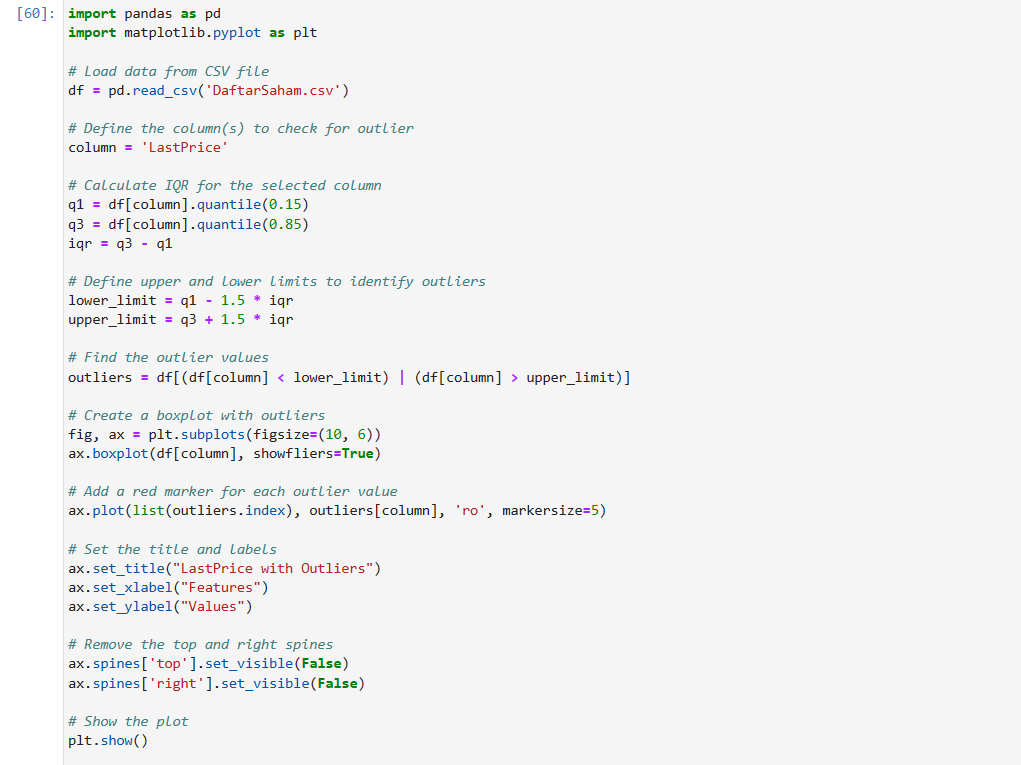


**106\_Mohamad Ibnu   
Fajar Maulana**

* Gambar diatas terlihat beberapa outliers pada *MarketCap* namun, tidak terlalu menyebar yang artinya nilai pasar yang terjadi pada beberapa saham saling bersaing satu sama lain.

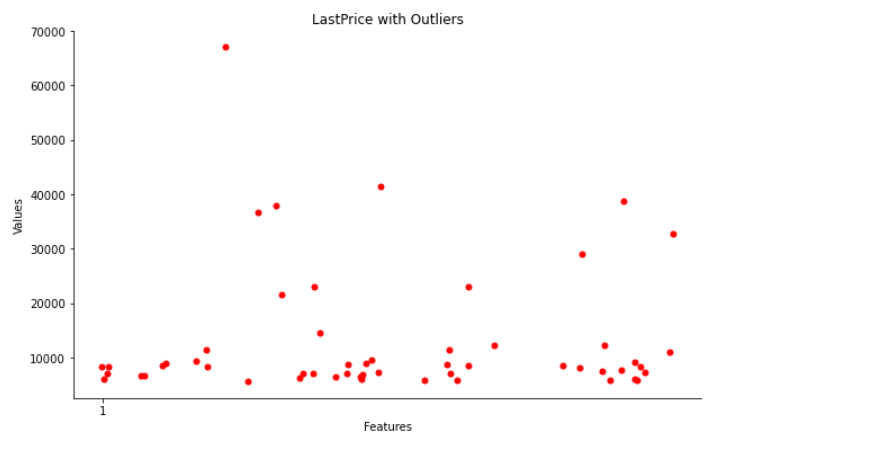
**106\_Mohamad Ibnu   
Fajar Maulana**

* Kemudian kita harus menemukan outliers pada kolom *LastPrice* karena saling berhubungan dengan kolom *MarketCap.*
* Lakukan kodingan seperti dibawah ini untuk menemukan outliers:



* Gambar diatas sama dengan penjelasan untuk menemukan outliers pada kolom *MarketCap*, sama-sama menggunakan konsep IQR (interquartile range) dan definisi upper dan lower limits. Upper dan lower limits didefinisikan sebagai 1,5 \* IQR di atas Q3 dan di bawah Q1. Data yang di luar upper dan lower limits dianggap sebagai outlier. Kemudian outlier ditandai dengan titik merah pada boxplot dan ditampilkan pada plot.
* Yang membedakan adalah nilai quartilnya yang mana pada *MarketCap* memiliki nilai , sedangkan untuk *LastPrice* memiliki nilaisehingga mendapatkan output outliers seperti gambar dibawah ini:

**106\_Mohamad Ibnu   
Fajar Maulana**

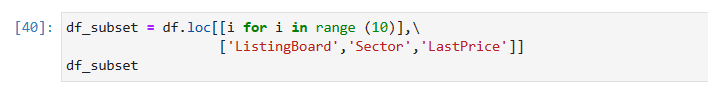


**106\_Mohamad Ibnu   
Fajar Maulana**

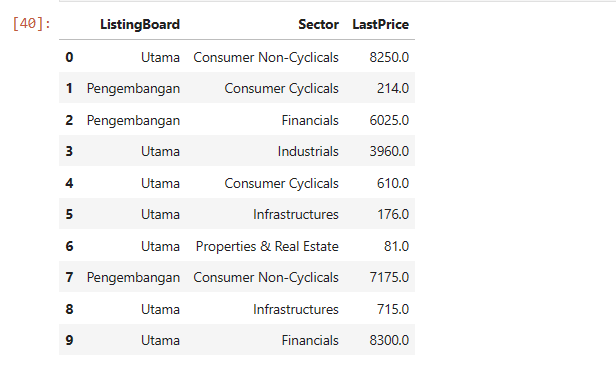
* Gambar diatas memvisualisasikan titik-titik merah pada outliers *LastPrice* yang artinya harga penutupan pada saham saling berbeda secara signifikan satu sama lain.

### Fungsi GroupBy

* Load data baru dengan teknik subsetting yang sudah dijelaskan diatas, seperti gambar dibawah ini:



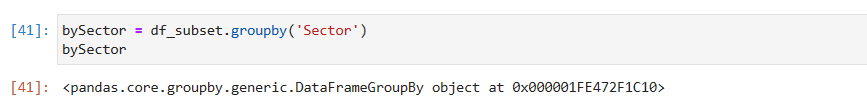
* Gambar diatas memanggil 10 data dan menggabungkan kolom *ListingBoard, Sector,* dan *LastPrice*.
* Kemudian mendapatkan output:



**106\_Mohamad Ibnu   
Fajar Maulana**

* Kemudian membuat DataFrame dengan method *GroupBy*, dengan kode seperti gambar dibawah ini:

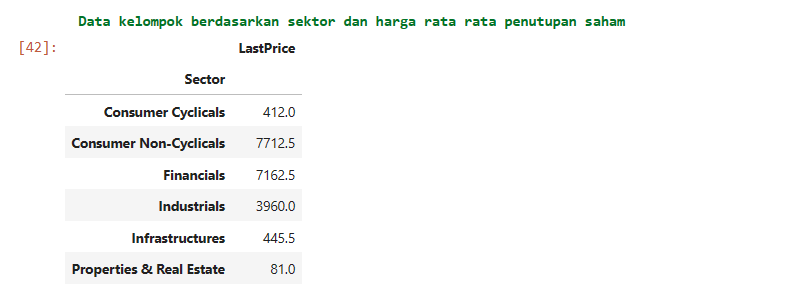
**106\_Mohamad Ibnu   
Fajar Maulana**



* Gambar diatas berdasarkan kolom *Sector*
* Kemudian kita menghitung rata-rata harga penutupan saham berdasarkan *Sector*, dengan kode *BySector.mean().* Sebagai berikut:



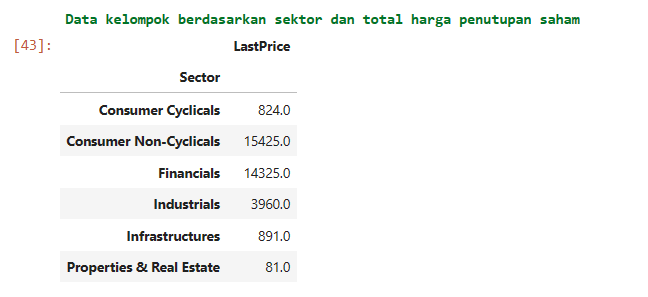
* Sehingga mendapatkan output:



* Gambar diatas berdasarkan sektornaya seperti sektor industri, keunagan dll (silahkan lihat gambar) dapat diketahui harga rata -rata penutupan saham.
* Selanjutnya kita dapat menghitung total harga penutupan saham dari berbagai sektor dengan kode *bySector.sum()*, sebagai berikut:



* Kemudian mendapatkan output:



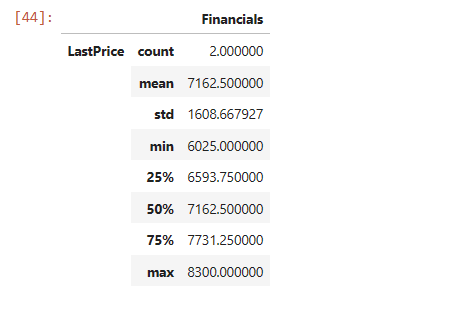
**106\_Mohamad Ibnu   
Fajar Maulana**

* Setelah itu, kita melihat pendeskripsian secara statistik pada sektor keuangan(Financials), dengan kode berikut ini:

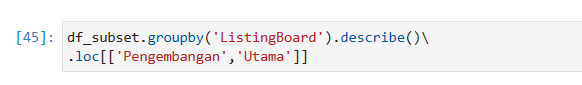
**106\_Mohamad Ibnu   
Fajar Maulana**



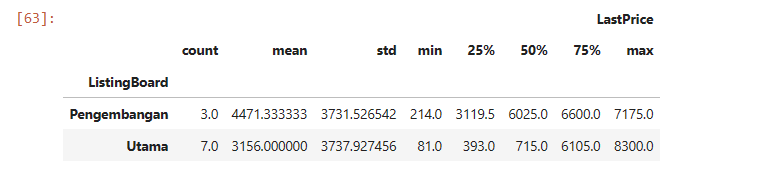
* Kemudian mendapatkan hasil:



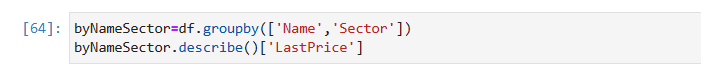
* Pada gambar diatas pendeskripsian nilai min-max, rata rata dan standar deviasi pada sektor keuangan.
* Kemudian jika ingin mengetahui informasi terkait saham pada bursa efek dalam tahap pengembangan atau utama maka lakukan kodingan dibawah ini dengan menggunakan metode *Groupby:*



* Setelah itu mendapatkan data baru atau informasi baru mengenai daftar atau board saham pada bursa efek secara statistik.



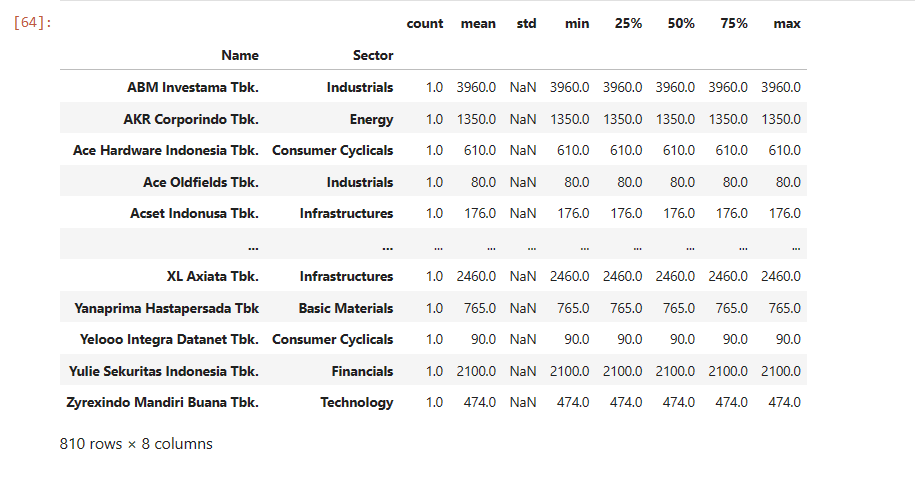
* Kemudian kita menggunakan metode *Groupby* untuk membuat informasi atau data baru berdasarkan nama, sektor dan harga penutupan sahamnya secara statistik dengan kode berikut:



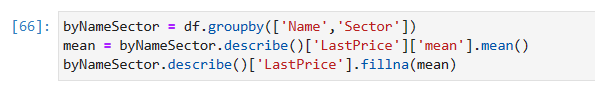
**106\_Mohamad Ibnu   
Fajar Maulana**

* Kemudian mendapatkan hasil informasi baru secara statistik sebagai berikut:

**106\_Mohamad Ibnu   
Fajar Maulana**



* Pada gambar diatas ktia melihat informasi beberapa perusahan dan sektornya yang memiliki harga penutupan saham secara statistik, jadi kita bisa mengetahui nilai minimum dan maksimum, rata-rata dan sebagainya mengenai harga penutupan saham. Akan tetapi, nilai standar deviasi harga penutupan saham(LastPrice) terdapat *missing value*.
* Tenang saja walaupun terdapat *missing value* hal tersebut bisa diatasi dengan kode skrip berikut ini:



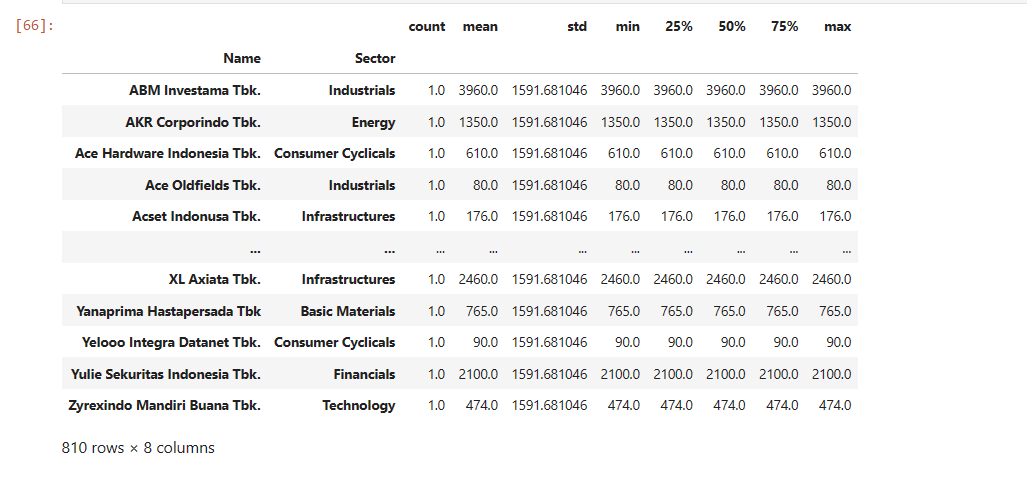
* *.fillna(mean)* pada kode tersebut digunakan untuk mengisi nilai yang hilang (NaN) pada suatu objek dengan nilai rata-rata (mean) dari objek tersebut. Dalam kasus ini, fillna(mean) digunakan untuk mengisi nilai-nilai NaN pada hasil keluaran describe()['LastPrice'] yang sebelumnya telah diambil hanya kolom 'mean'-nya saja. Oleh karena itu, setiap nilai NaN pada kolom 'mean' akan diganti dengan nilai rata-rata dari kolom 'mean' tersebut. Dalam hal ini, mean mengacu pada rata-rata nilai dari kolom 'mean' hasil keluaran sebelumnya.

Dengan demikian, penggunaan fillna(mean) pada kode di atas bertujuan untuk memperbaiki kelengkapan data sehingga data dapat digunakan secara lebih akurat dan lengkap dalam analisis selanjutnya.

**106\_Mohamad Ibnu   
Fajar Maulana**

* Sehingga mendapatkan output:

**106\_Mohamad Ibnu   
Fajar Maulana**



* Nilai standar deviasi tersedia pada harga penutupan saham(LastPrice)

**106\_Mohamad Ibnu   
Fajar Maulana**