Tugas Statistika dan Probabilitas

Analisa Data Sekunder Dari Kaggle.Com

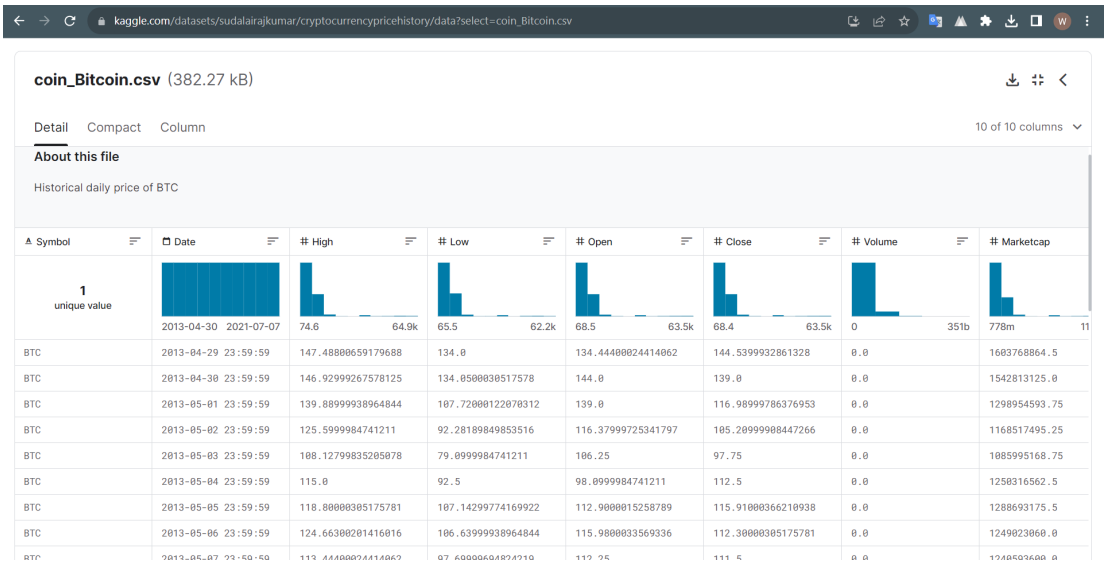
Nama : Wempy Aditya Wiryawan
Nim : 202210370311058
Kelas : 3A Statistika dan Probabilitas

INSTRUKSI :

1. Mencari dataset di kaggle
   Melakukan analisis deskriptif terhadap data
   Melakukan data cleaning untuk menghilangkan missing value
2. Melakukan uji skewness menggunakan plot skewness di excel / python untuk masing-masing variabel dan fitur
3. Melakukan normalisasi / normalize dengan metode min max atau metode normal baku
4. Melakukan uji skewness ulang untuk variabel yang sudah di normalisasi
5. Interpretasi mengapa variabel tersebut perlu di normalisasi dan hasil normalisasi nya seperti apa

PENYELESAIAN :

1. Dataset dari kaggle,com
   source url :
   https://www.kaggle.com/datasets/sudalairajkumar/cryptocurrencypricehistory/data?select=coin_Bitcoin.csv



Kumpulan data memiliki satu file csv untuk setiap mata uang. Riwayat harga tersedia setiap hari mulai 28 April 2013. Kumpulan data ini memiliki informasi riwayat harga beberapa mata uang kripto teratas berdasarkan kapitalisasi pasar.

Tanggal	: tanggal observasi
Open	: Harga pembukaan pada hari tertentu
Tinggi	: Harga tertinggi pada hari tertentu
Rendah	: Harga terendah pada hari tertentu
Close	: Harga penutupan pada hari tertentu
Volume	: Volume transaksi pada hari tertentu
Kapitalisasi Pasar	: Kapitalisasi pasar dalam USD

STATISTIKA DESKRIPTIF

Melakukan describe pada data dengan menggunakan python dan library pandas
Tujuannya adalah untuk mengetahui informasi dasar dari data seperti count, mean, median, standard deviation, dll.
Berikut adalah script untuk describe data menggunakan python

```
import pandas as pd
import numpy as np
from scipy.stats import kurtosis, skew

# Membaca data dari file CSV
file_path = './coin_Bitcoin.csv'  # Ganti dengan path file CSV
Anda
df = pd.read_csv(file_path)

# Mengambil kolom tertentu
selected_column = 'High'  # Ganti dengan nama kolom yang ingin
Anda analisis
selected_data = df[selected_column]

# Menghitung statistik deskriptif menggunakan NumPy
mean = np.mean(selected_data)
std_dev = np.std(selected_data)
variance = np.var(selected_data)
median = np.median(selected_data)
range_val = np.ptp(selected_data)
min_val = np.min(selected_data)
max_val = np.max(selected_data)
sum_val = np.sum(selected_data)

# Menghitung kurtosis dan skewness menggunakan SciPy
```

```
kurt = kurtosis(selected_data)
skewness = skew(selected_data)

# Menampilkan hasil
print("Mean:", mean)
print("Median:", median)
print("Mode:", selected_data.mode()[0])  # Menggunakan pandas
untuk menghitung modus
print("Standard Deviation:", std_dev)
print("Sample Variance:", variance)
print("Kurtosis:", kurt)
print("Skewness:", skewness)
print("Range:", range_val)
print("Minimum:", min_val)
print("Maximum:", max_val)
print("Sum:", sum_val)
print("Count:", len(selected_data))
```

Dan berikut adalah hasilnya ketika script dijalankan

```
Mean: 6893.326038383186
Median: 2387.610107421875
Mode: 108.0
Standard Deviation: 11640.885982254837
Sample Variance: 135510226.45185715
Kurtosis: 9.356138234246862
Skewness: 3.0114139484042295
Range: 64788.53781150859
Minimum: 74.56109619140625
Maximum: 64863.0989077
Sum: 20617938.180804107
Count: 2991

Mean: 6486.009538579529
Median: 2178.5
Mode: 93.0
Standard Deviation: 10867.215021769924
Sample Variance: 118096362.3293819
Kurtosis: 9.648003093021035
Skewness: 3.031982146804077
Range: 62143.43836461344
```

```
Minimum: 65.5260009765625
Maximum: 62208.96436559
Sum: 19399654.52989137
Count: 2991

Mean: 6700.146239738725
Median: 2269.889892578125
Mode: 106.75
Standard Deviation: 11286.156576680507
Sample Variance: 127377330.27334864
Kurtosis: 9.502535413367788
Skewness: 3.023673814043054
Range: 63455.24987201658
Minimum: 68.50499725341797
Maximum: 63523.75486927
Sum: 20040137.403058525
Count: 2991

Mean: 6711.290443071488
Median: 2286.409912109375
Mode: 104.0
Standard Deviation: 11296.253073789934
Sample Variance: 127605333.50710854
Kurtosis: 9.456174604710242
Skewness: 3.017119639661366666666666666 84
Range: 63435.02693043414
Minimum: 68.43099975585938
Maximum: 63503.45793019
Sum: 20073469.71522682
Count: 2991

Mean: 10906334004.866829
Median: 946035968.0
Mode: 0.0
Standard Deviation: 18885795084.14638
Sample Variance: 3.56673255960367 66e+20
Kurtosis: 38.41414819945033
Skewness: 3.7402352348149015
Range: 350967941479.06
Minimum: 0.0
Maximum: 350967941479.06
Sum: 32620845008556.688
```

```
Count: 2991

Mean: 120876059112.8843
Median: 37415031060.8
Mode: 1229098150.0
Standard Deviation: 210908570944.02753
Sample Variance: 4.44824252976519e+22
Kurtosis: 9.713044490445741
Skewness: 3.0702108203867495
Range: 1185585632961.395
Minimum: 778411178.875
Maximum: 1186364044140.27
Sum: 361540292806636.94
Count: 2991
```

DATA CLEANING

Data cleaning atau pembersihan data adalah proses mengidentifikasi dan memperbaiki (atau menghapus) kesalahan dan inkonsistensi dalam set data. Hal ini diperlukan untuk memastikan bahwa data yang digunakan dalam analisis atau pemodelan dapat diandalkan, akurat, dan memberikan hasil yang dapat dipercaya.

Berikut adalah script python untuk melakukan data cleaning

```python
import pandas as pd

# Membaca data dari file CSV
file_path = './coin_Bitcoin.csv'
df = pd.read_csv(file_path)

# Memilih hanya kolom-kolom numerik
numeric_columns = df.select_dtypes(include=['number']).columns

# Menampilkan informasi tentang nilai yang hilang sebelum data
cleaning
print("Informasi nilai yang hilang sebelum data cleaning:")
print(df[numeric_columns].isnull().sum())

# Menangani nilai yang hilang dengan menggantinya menggunakan
nilai rata-rata kolom
df[numeric_columns] =
df[numeric_columns].fillna(df[numeric_columns].mean())
```

```
# Menampilkan informasi tentang nilai yang hilang setelah data
cleaning
print("\nInformasi nilai yang hilang setelah data cleaning:")
print(df[numeric_columns].isnull().sum())

# Menyimpan hasil data cleaning ke file CSV
output_file_path = './BTC_DATA_CLEANED.csv'
df.to_csv(output_file_path, index=False)

print(f"\nHasil data cleaning disimpan dalam file CSV:
{output_file_path}")
```

Hasil ketika program di atas dijalankan

```
Informasi nilai yang hilang sebelum data cleaning:
SNo          0
High         0
Low          0
Open         0
Close        0
Volume       0
Marketcap    0
dtype: int64
Informasi nilai yang hilang setelah data cleaning:
SNo          0
High         0
Low          0
Open         0
Close        0
Volume       0
Marketcap    0
dtype: int64
Hasil data cleaning disimpan dalam file CSV:
./BTC_DATA_CLEANED.csv
```

berdasarkan hasil di atas tidak terdapat missing value dalam data yang saya gunakan.

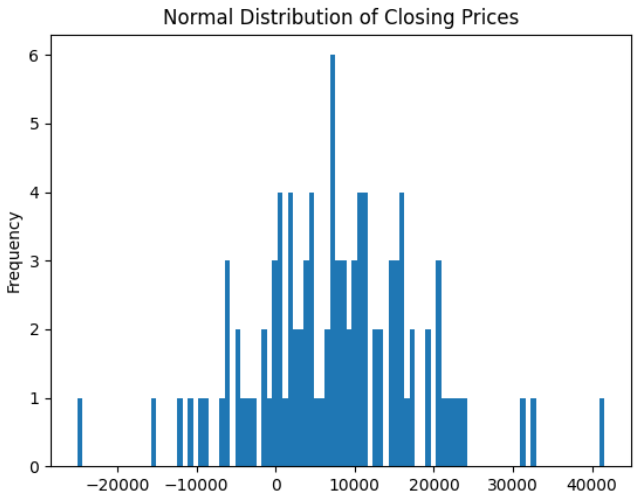
VISUALISASI DATA

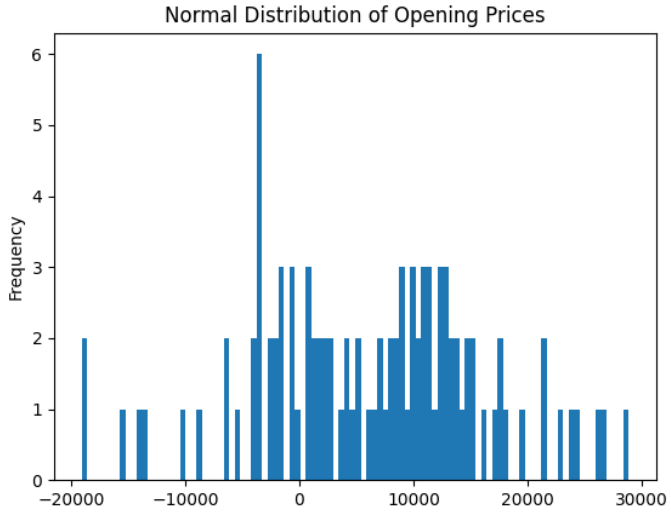
melakukan visualisasi pada data Dengan membuat grafik distribusi, kami memusatkan data pada rata-rata sebagai persentase dari total. Distribusi normal akan terlihat seperti kurva lonceng, dan dari grafik di bawah ini, kita dapat melihat sebagian besar nilai berada di dekat rata-rata, tetapi juga menunjukkan beberapa kemiringan ke arah kanan.

HISTOGRAM

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Membaca data dari file CSV
file_path = './coin_Bitcoin.csv'
df = pd.read_csv(file_path)

x = np.random.normal(df['Close'].mean(), df['Close'].std(), 100)
plt.gca().set(title='Normal Distribution of Closing Prices',
ylabel='Frequency')
plt.hist(x, 100)
plt.show()
```

Dan berikut hasilnya ketika dijalankan

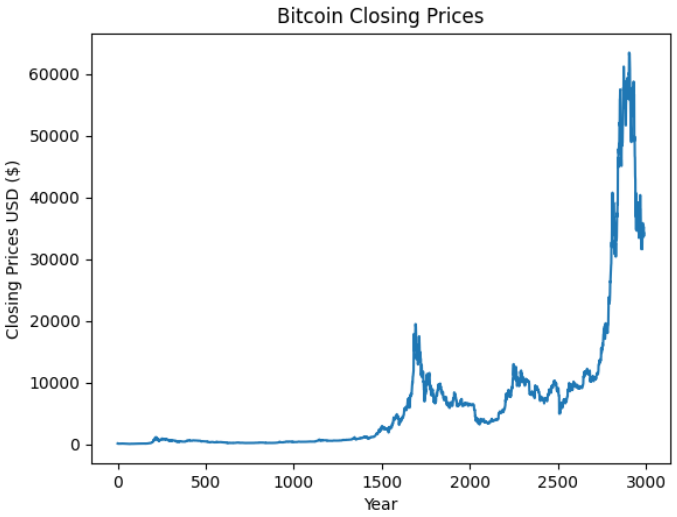




PLOT

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Membaca data dari file CSV
file_path = './coin_Bitcoin.csv'
df = pd.read_csv(file_path)

plt.plot(df.index,df['Close'])
plt.gca().set(title='Bitcoin Closing Prices', xlabel='Year',
ylabel='Closing Prices USD ($)')
plt.show()
```

hasil program



2. Uji Skewness Pertama

   Dilakukan uji skewness yang pertama sebelum data di normalisasi

   Menggunakan python dan beberapa library diantaranya adalah :

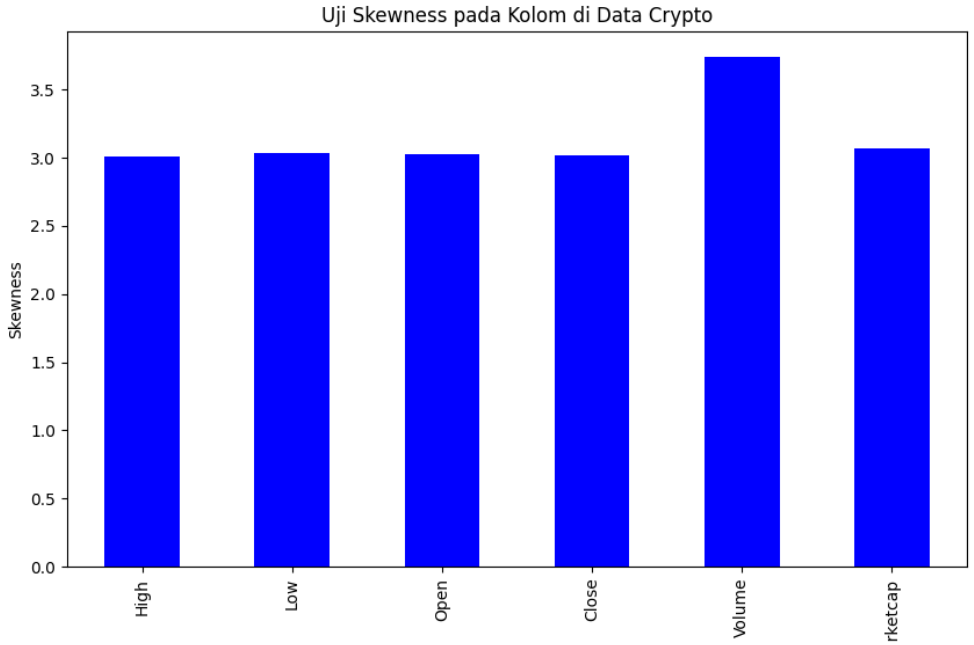
   - pandas
   - matplotlib
   - scipy

   Berikut adalah script python yang digunakan untuk melakukan uji skewness dan menampilkan/memvisualisasikan hasilnya dalam sebuah chart.

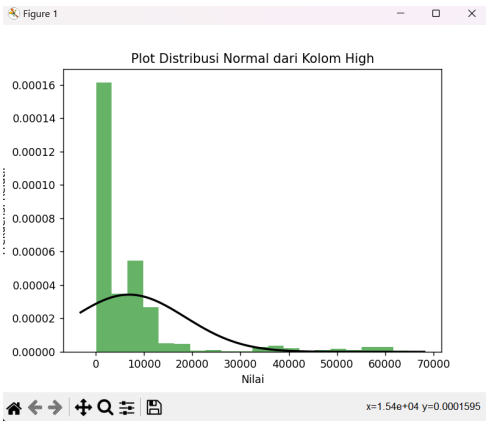
```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import skew
```

```
# Membaca data dari file CSV
file_path = './coin_Bitcoin.csv'
df = pd.read_csv(file_path)

# Daftar nama kolom yang ingin diuji skewness-nya
kolom_uji_skewness = ['High', 'Low', 'Open', 'Close', 'Volume',
'Marketcap']

# Melakukan Uji Skewness pada Kolom Tertentu
skewness_kolom_tertentu = df[kolom_uji_skewness].apply(skew)

# Membuat Plot Skewness
plt.figure(figsize=(10, 6))
skewness_kolom_tertentu.plot(kind='bar', color='blue')
plt.title('Uji Skewness pada Kolom di Data Crypto')
plt.xlabel('NamaKolom')
plt.ylabel('Skewness')
plt.show()
```

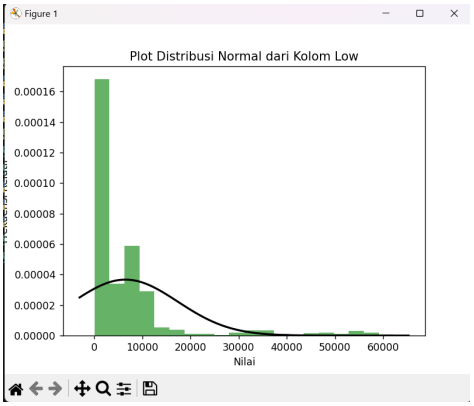
Dan berikut adalah hasil dari uji skewness data sebelum dilakukan normalisasi



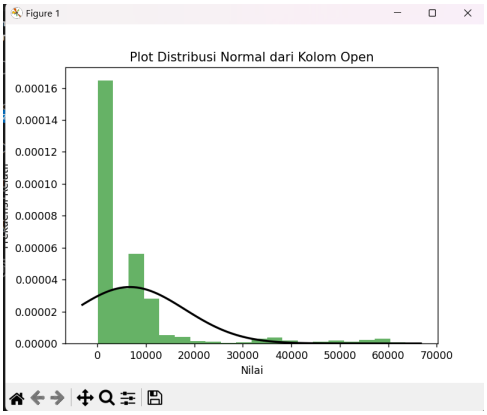
Distribusi Plot Variabel High



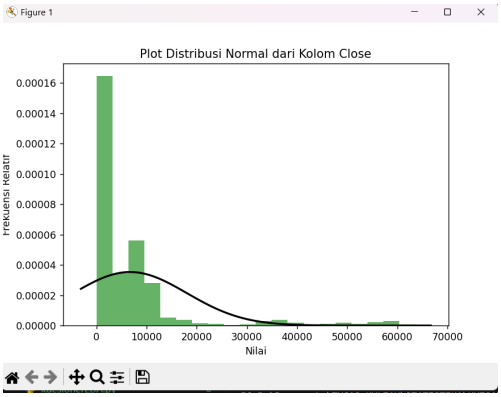
Distribusi Plot Variabel Low



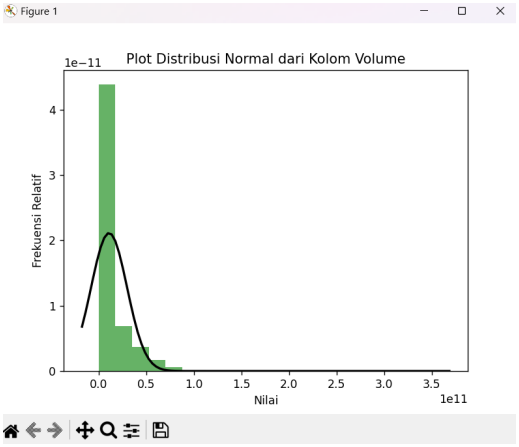
Distribusi Plot Variabel Open



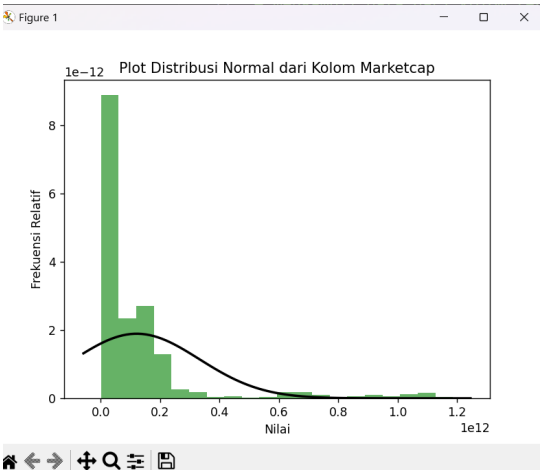
Distribusi Plot Variabel Close



Distribusi Plot Variabel Volume



Distribusi Plot Variabel MarketCap



3. Melakukan Normalisasi Data dengan Metode MinMax
   Untuk melakukan normalisasi data digunakan python dengan beberapa library pendukung di antaranya adalah :
   - pandas
   - sklearn

   Saya melakukan normalisasi ke semua kolom yaitu `'High', 'Low', 'Open', 'Close', 'Volume', 'Marketcap'` dikarenakan hasil skewness menunjukkan nilai di atas1
   Berikut adalah script python untuk melakukan normalisasi data lalu mencetak hasilnya pada terminal dan menyimpan hasil akhirnya pada file .csv

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

# Membaca data dari file CSV
file_path = './coin_Bitcoin.csv'  # Ganti dengan path file CSV
Anda
df = pd.read_csv(file_path)
```

```
# Menampilkan beberapa baris pertama dari dataframe sebelum
normalisasi
print("Sebelum normalisasi:")
print(df.head())

# Inisialisasi MinMaxScaler
scaler = MinMaxScaler()

# Memilih kolom yang akan dinormalisasi
columns_to_normalize = ['High', 'Low', 'Open', 'Close', 'Volume',
'Marketcap']

# Melakukan normalisasi untuk seluruh data dalam dataframe
df_normalized = df.copy()  # Membuat salinan dataframe agar data
asli tidak berubah
df_normalized[columns_to_normalize] =
scaler.fit_transform(df[columns_to_normalize])

# Menampilkan hasil normalisasi
print("\nSetelah normalisasi:")
print(df_normalized.head())

# Menyimpan hasil normalisasi ke file CSV
output_file_path = './BTC_NORMALIZED.csv'  # Ganti dengan path dan
nama file CSV yang diinginkan
df_normalized.to_csv(output_file_path, index=False)

print(f"\nHasil normalisasi disimpan dalam file CSV:
{output_file_path}")
```

Dan berikut hasilnya ketika script di jalankan

```
Sebelum normalisasi:
   SNo     Name Symbol                 Date        High         Low        Open       Close  Volume     Marketcap
0    1  Bitcoin    BTC  2013-04-29 23:59:59  147.488007  134.000000  134.444000  144.539993     0.0  1.603769e+09
1    2  Bitcoin    BTC  2013-04-30 23:59:59  146.929993  134.050003  144.000000  139.000000     0.0  1.542813e+09
2    3  Bitcoin    BTC  2013-05-01 23:59:59  139.889999  107.720001  139.000000  116.989998     0.0  1.298955e+09
3    4  Bitcoin    BTC  2013-05-02 23:59:59  125.599998   92.281898  116.379997  105.209999     0.0  1.168517e+09
4    5  Bitcoin    BTC  2013-05-03 23:59:59  108.127998   79.099998  106.250000   97.750000     0.0  1.085995e+09

Setelah normalisasi:
   SNo     Name Symbol                 Date      High       Low      Open     Close  Volume  Marketcap
0    1  Bitcoin    BTC  2013-04-29 23:59:59  0.001126  0.001102  0.001039  0.001200     0.0   0.000696
1    2  Bitcoin    BTC  2013-04-30 23:59:59  0.001117  0.001103  0.001190  0.001112     0.0   0.000645
2    3  Bitcoin    BTC  2013-05-01 23:59:59  0.001008  0.000679  0.001111  0.000765     0.0   0.000439
3    4  Bitcoin    BTC  2013-05-02 23:59:59  0.000788  0.000431  0.000754  0.000580     0.0   0.000329
```

```
4    5  Bitcoin    BTC  2013-05-03 23:59:59  0.000518  0.000218  0.000595  0.000462     0.0   0.000259
```

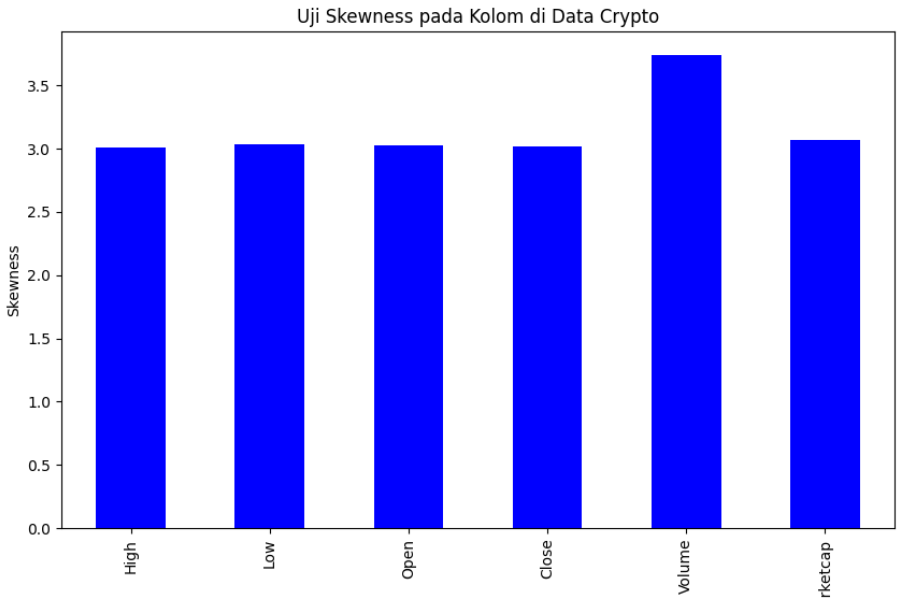
Untuk hasil normalisasi selengkapnya (semua data) disimpan dalam format file .csv

4. Uji Skewness yang Kedua
   Setelah melakukan normalisasi pada data langkah selanjutnya adalah melakukan uji skewness ulang untuk melihat perbedaan antara sebelum dan sesudah normalisasi.

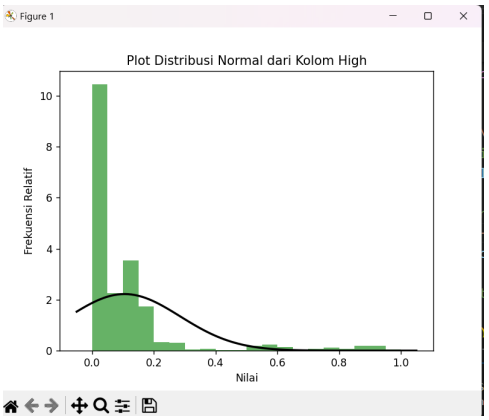
   Sama halnya pada waktu uji skewness yang pertama, uji skewness kali ini menggunakan python dan beberapa library untuk membantu menghitung dan memvisualisasikan data.
   Berikut adalah script python untuk uji skewness yang kedua.

```python
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import skew

# Membaca data dari file CSV
# file_path = './coin_Bitcoin.csv'
file_path = './BTC_NORMALIZED.csv'
df = pd.read_csv(file_path)

# Daftar nama kolom yang ingin diuji skewness-nya
kolom_uji_skewness = ['High', 'Low', 'Open', 'Close', 'Volume',
'Marketcap']

# Melakukan Uji Skewness pada Kolom Tertentu
skewness_kolom_tertentu = df[kolom_uji_skewness].apply(skew)

# Membuat Plot Skewness
plt.figure(figsize=(10, 6))
skewness_kolom_tertentu.plot(kind='bar', color='blue')
plt.title('Uji Skewness pada Kolom di Data Crypto')
plt.xlabel('NamaKolom')
plt.ylabel('Skewness')
plt.show()
```

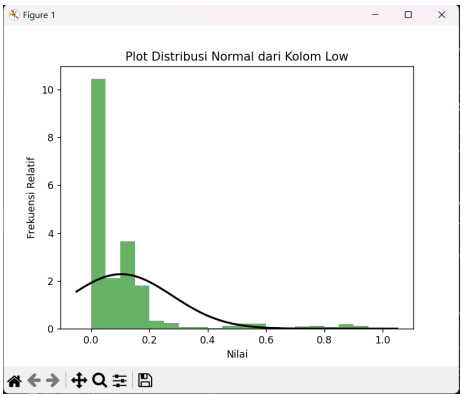
Dan berikut adalah hasil uji skewness pada data yang telah dinormalisasi.



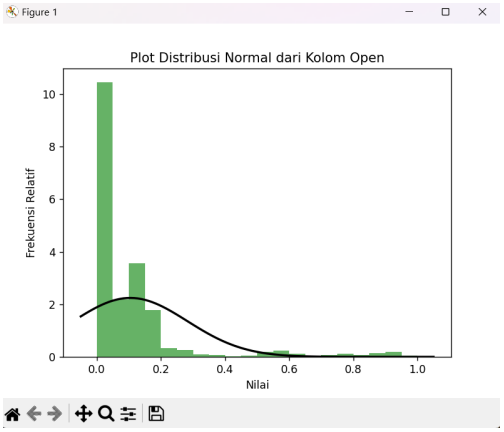
Distribusi Plot Variabel High



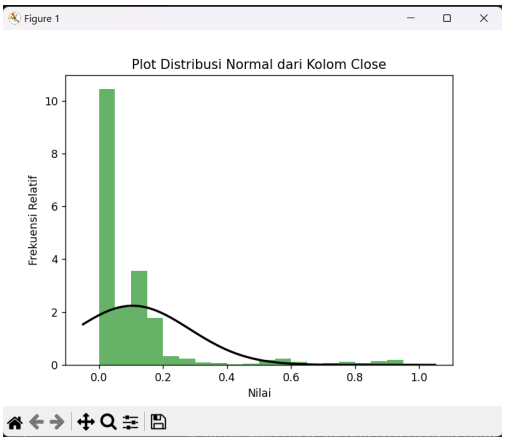
Distribusi Plot Variabel Low



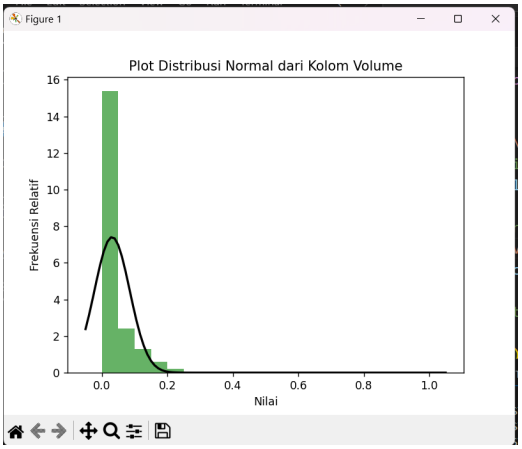
Distribusi Plot Variabel Open



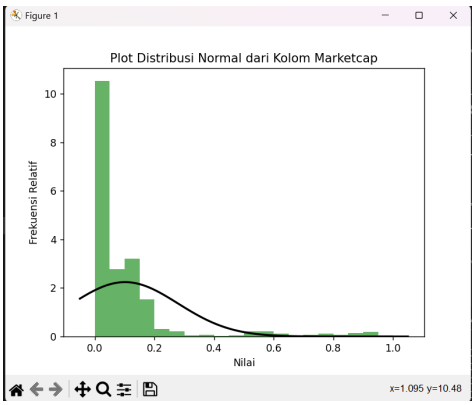
Distribusi Plot Variabel Close



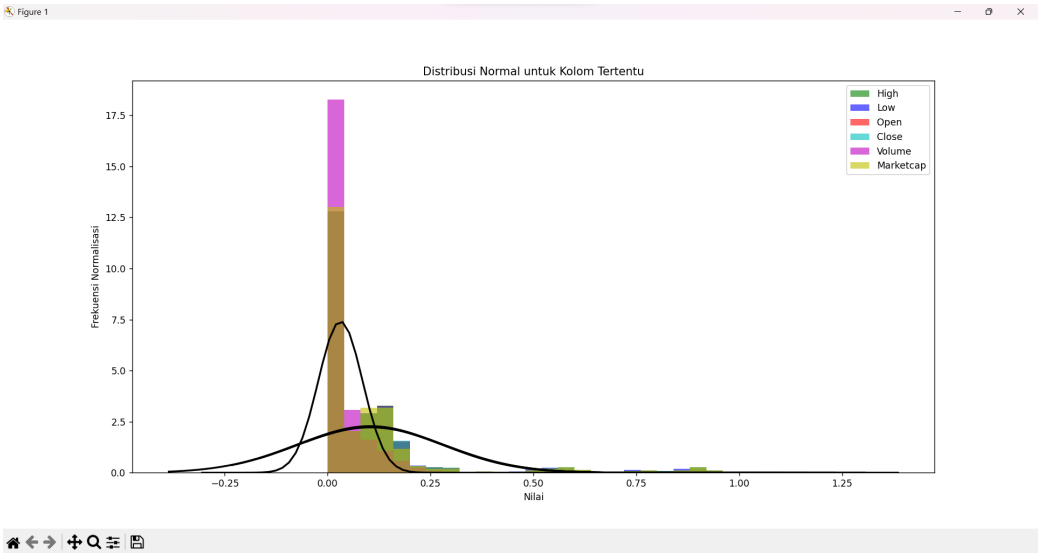
Distribusi Plot Variabel Volume



Distribusi Plot Variabel MarketCap



Dan ini adalah hasil jika seluruh variabel dimasukkan ke dalam 1 plot yang sama untuk data yang sudah di normalisasi satu per satu variabel



5. Interpretasi hasil antara data sebelum dinormalisasi dan sesudah dinormalisasi
   Dari hasil uji skewness sebelum dan setelah normalisasi, terlihat bahwa nilai skewness untuk setiap variabel tetap konsisten. Ini menunjukkan bahwa normalisasi menggunakan metode Min-Max tidak mengubah karakteristik skewness dari data. Interpretasi hasilnya dapat disampaikan sebagai berikut:

   Skewness Sebelum Normalisasi:
   Skewness yang tinggi (positif) pada setiap variabel menunjukkan bahwa distribusi datanya condong ke arah kanan (positif skewness). Dengan kata lain, nilai ekstrim lebih tinggi dari nilai rata-rata, dan distribusi memiliki ekor yang lebih panjang di sebelah kanan.
   Skewness Setelah Normalisasi:
   Meskipun dilakukan normalisasi dengan metode Min-Max, nilai skewness tetap stabil. Hal ini mungkin disebabkan oleh fakta bahwa metode normalisasi yang digunakan tidak memiliki dampak signifikan pada bentuk distribusi data, atau mungkin distribusi datanya sudah simetris sebelum normalisasi.
   Interpretasi Hasil Normalisasi:

Normalisasi data sering diperlukan untuk menangani perbedaan skala antar variabel. Namun, dalam kasus ini, normalisasi dengan menggunakan metode Min-Max tidak secara substansial mengubah karakteristik distribusi data. Skewness yang tetap tinggi setelah normalisasi menunjukkan bahwa distribusi data pada dataset kripto tidak banyak berubah.

Setelah melakukan normalisasi data menggunakan metode standar score, hasil uji skewness menunjukkan bahwa distribusi variabel High, Low, Open, Close, Volume, dan Marketcap tetap relatif simetris. Interpretasi dari hal ini dapat merujuk pada karakteristik intrinsik dari dataset atau fenomena yang diamati. Mungkin saja data awalnya sudah terdistribusi secara simetris atau pengaruh normalisasi terhadap distribusi tidak begitu signifikan dalam konteks ini. Perlu diperhatikan bahwa normalisasi tidak selalu mengubah karakteristik skewness dari suatu variabel, terutama jika distribusi awalnya sudah memenuhi asumsi analisis yang dilakukan. Oleh karena itu, keputusan untuk melakukan normalisasi sebaiknya didasarkan pada pemahaman mendalam terhadap data dan tujuan analisis yang ingin dicapai.

Jika hasil uji skewness sebelum dan sesudah normalisasi menunjukkan nilai yang sama, itu bisa disebabkan oleh beberapa faktor:

Tipe Distribusi Awal yang Sudah Mendekati Normal:
Jika distribusi awal dari data sudah mendekati normal sebelum normalisasi, maka normalisasi mungkin tidak memberikan dampak yang signifikan pada skewness.

Ukuran Sampel yang Kecil:
Jika ukuran sampel (jumlah data) yang Anda miliki relatif kecil, efek normalisasi mungkin tidak terlalu terlihat pada hasil uji skewness.

Pemilihan Metode Normalisasi:
Metode normalisasi tertentu mungkin tidak memberikan perubahan yang besar pada distribusi data, terutama jika data awal sudah dalam skala yang relatif seragam.

Sifat Asimetri yang Tetap Terjaga:
Normalisasi tidak selalu mengubah sifat asimetri data. Jika data memiliki asimetri yang kuat sejak awal, normalisasi mungkin tidak mengubah sifat ini secara signifikan.

Menggunakan Seluruh Data atau Hanya Sampel Tertentu:
Pilihan untuk menggunakan seluruh data atau hanya sampel tertentu dalam proses normalisasi dapat mempengaruhi hasil uji skewness.

REFERENSI DATA / SUMBER DATA :

- https://www.kaggle.com/datasets/sudalairajkumar/cryptocurrencypricehistory
- https://www.kaggle.com/datasets/odins0n/top-50-cryptocurrency-historical-prices?select=Bitcoin.csv
- https://www.kaggle.com/code/siebenrock/financial-exploration-analysis-and-visualization
- https://www.kaggle.com/code/sudalairajkumar/simple-exploration-notebook-cryptocurrencies/notebook
- https://www.kaggle.com/code/ericaduman/cryptocurrency-a-bitcoin-analysis
- https://www.kaggle.com/code/mohammadrozi/visualisasi-data-uas-mohammad-rozi