Introduction

Wahyu Satrio – 202131149

A. Latar Belakang :

Industri perbankan dan keuangan semakin bersaing ketat dalam merebut dan mempertahankan nasabah. Salah satu aspek penting dalam industri ini adalah pemberian kredit kepada nasabah. Bank perlu memahami profil dan perilaku nasabah dengan lebih baik agar dapat menentukan batas kredit yang tepat serta menawarkan produk dan layanan yang sesuai.

Oleh karena itu, klasterisasi nasabah menjadi sangat penting. Dengan mengelompokkan nasabah ke dalam klaster-klaster yang memiliki karakteristik serupa, bank dapat memahami kebutuhan spesifik tiap klaster nasabah dan memberikan pengalaman nasabah yang lebih personal.

Salah satu teknik klasterisasi yang populer adalah K-means clustering. K-means clustering merupakan algoritma pembelajaran terbimbing sederhana yang dapat digunakan untuk mengelompokkan data numerik yang besar ke dalam klaster-klaster berdasarkan kesamaan karakteristiknya.

Pada studi kasus ini, K-means clustering digunakan untuk mengelompokkan nasabah kartu kredit berdasarkan batas kredit rata-rata dan rasio utilisasi kredit mereka. Dengan penerapan teknik ini, diharapkan dapat diperoleh wawasan yang lebih dalam tentang pola perilaku dan kebutuhan beragam kelompok nasabah kartu kredit. Hasil klasterisasi ini nantinya dapat digunakan untuk pengambilan keputusan kredit dan pengembangan produk yang lebih tepat sasaran.

B. Tinjauan Pustaka

A. A Credits Based Scheduling Algorithm with K- means Clustering

The paper "A Credits Based Scheduling Algorithm with Kmeans Clustering" by Sharma introduces an advanced scheduling algorithm for task execution in cloud computing environments. The proposed algorithm utilizes a credits-based approach and K-means clustering to optimize task scheduling and resource allocation (National Institute of Technology (Punjab et al., n.d.). The algorithm dynamically assigns task priorities based on quality of service (QoS) parameters, such as cost of application and execution time, to overcome the limitations of static priority assignment. Additionally, the algorithm incorporates a multi-objective approach to minimize task execution time and improve system throughput while considering QoS as a primary (National Institute of Technology (Punjab et al., n.d.).

The research emphasizes the significance of cloud computing in providing utility-oriented IT services and the challenges associated with optimal resource management and utilization. It addresses the growing trend of enterprises shifting their infrastructure to cloud environments and the need for efficient scheduling strategies to handle diverse and data-intensive applications (National Institute of Technology (Punjab et al., n.d.). Furthermore, the paper highlights the use of virtualization technology and the CloudSim 3.0.3 simulator for implementing the proposed scheduling algorithm, demonstrating its empirical superiority over other job scheduling techniques through experimental results (National Institute of Technology (Punjab et al., n.d.).

In summary, the paper presents a comprehensive approach to task scheduling in cloud computing, integrating credits-based scheduling, dynamic priority assignment, K-means

clustering, and multi-objective optimization to enhance system performance and resource utilization.

B. Research on Power Market User Credit Evaluation Based on K-Means Clustering and Contour Coefficient

Untuk menilai kredit pengguna listrik, penelitian ini menggunakan model clustering K-Means, yang merupakan algoritma pembelajaran tanpa pengawasan yang mengelompokkan data ke dalam berbagai kategori. Pusat klaster berulang kali dipindahkan dan diperbarui berdasarkan kesamaan antara sampel hingga konvergensi atau kondisi terminasi terpenuhi dengan memberikan jumlah klaster dan pusat klaster awal (Lin et al., 2020, p. 6).

Proses spesifik dari algoritma K-Means clustering meliputi:

1. Menetapkan jumlah klaster k dan memilih k sampel dari n sampel sebagai pusat klaster awal.

2. Menghitung jarak antara setiap sampel dan setiap pusat klaster, kemudian mengklasifikasikan sampel ke dalam kategori klaster dengan jarak terdekat.

3. Memperbarui pusat klaster untuk membentuk pusat klaster baru dengan nilai rata-rata dari semua sampel di setiap kategori.

4. Mengulangi langkah-langkah (2) dan (3) hingga perubahan dari setiap pusat klaster kurang dari ambang tertentu atau perubahan dari jumlah kuadrat jarak antara semua sampel dan pusat klaster dari kategori kurang dari ambang tertentu (Lin et al., 2020, p. 6).

Penelitian ini menggunakan algoritma clustering K-Means untuk menilai kredit pengguna listrik. Setelah memproses dan menghitung data yang relevan dari 127 pengguna listrik, penilaian kredit tiga tingkat dan empat tipe lebih sesuai dengan keadaan sebenarnya daripada penilaian kredit tiga tingkat dan tiga tipe. Efek pengelompokan dinilai menggunakan metode koefisien kontur. Ini menggabungkan elemen kesamaan dan ketidaksesuaian (Lin et al., 2020, p. 5).

Karena keunggulannya yang relatif sederhana dan efektif dalam kondisi jumlah klaster dan pusat klaster yang ditetapkan, penelitian ini menunjukkan bahwa algoritma pengelompokan K-Means banyak digunakan. Namun, hasil pengelompokan dapat berubah secara signifikan karena pusat kelas awal yang berbeda memengaruhi algoritma (Lin et al., 2020, p. 5).

C. Credit ratings of Chinese online loan platforms based on factor scores and K-means clustering algorithm

Penelitian yang dilakukan oleh Chen (Chen et al., 2023, p. 1) berfokus pada pengembangan mekanisme pemeringkatan kredit untuk platform pinjaman online (OLP) di Cina dengan menggunakan kombinasi indikator kuantitatif dan kualitatif. Penelitian ini membangun sistem indikator peringkat kredit untuk 130 OLP utama di Tiongkok, dengan menggabungkan 12 metrik kuantitatif operasi pinjaman online dan dua indikator kualitatif yang mencerminkan karakteristik Tiongkok. Untuk mengurangi dimensi dari 14 indikator tersebut,

analisis faktor digunakan, yang menghasilkan faktor skala operasi OLP, faktor penyebaran dana, faktor keamanan, dan faktor profitabilitas. Selanjutnya, algoritma pengelompokan K-means digunakan untuk mengelompokkan skor faktor dari setiap OLP, sehingga memperoleh hasil peringkat kredit. Studi ini menemukan bahwa metode pemeringkatan kredit berbasis pembelajaran mesin yang diusulkan secara efektif memberikan peringatan dini terhadap platform yang bermasalah dan menghasilkan peringkat kredit yang lebih akurat dibandingkan dengan peringkat kredit yang diberikan oleh situs web pemeringkatan pinjaman online arus utama di Cina.

Penggunaan algoritma K-means clustering dalam penelitian ini sangat penting. Algoritma K-means clustering adalah algoritma pembelajaran tanpa pengawasan yang sering digunakan untuk penilaian kredit . Namun, penting untuk dicatat bahwa korelasi antara indikator data dapat menghasilkan noise yang mengganggu akurasi K-means clustering. Selain itu, penelitian ini juga menggunakan analisis faktor untuk mengurangi dimensi indikator, yang merupakan langkah penting dalam mengembangkan mekanisme pemeringkatan kredit (Chen et al., 2023, p. 1).

Lebih lanjut, penelitian ini menekankan pentingnya pembelajaran mesin dalam konteks pemeringkatan kredit. Penelitian ini menyoroti bahwa algoritma machine learning, seperti regresi linier dan pohon keputusan, secara otomatis meningkatkan dan mengoptimalkan melalui pelatihan. Algoritma pembelajaran yang diawasi, khususnya, memainkan peran penting dalam melakukan pembelajaran klasifikasi sesuai dengan sampel pelatihan dengan fitur berlabel, memastikan sampel yang diklasifikasikan secara akurat (Chen et al., 2023, p. 1).

Sebagai kesimpulan, penelitian oleh Chen memberikan wawasan yang berharga dalam pengembangan mekanisme peringkat kredit untuk platform pinjaman online di Cina. Dengan memanfaatkan analisis faktor dan algoritma pengelompokan K-means, penelitian ini menunjukkan keefektifan metode pemeringkatan kredit berbasis pembelajaran mesin dalam memberikan peringkat kredit yang akurat dan peringatan dini terhadap platform yang bermasalah.

D. Analysis of Agricultural Credit Performance of Turkey using K-means Clustering Algorithm

Dalam studi "Analisis Kinerja Kredit Pertanian di Turki menggunakan Algoritma Kmeans Clustering", para peneliti bertujuan untuk membandingkan kinerja kredit pertanian di 81 provinsi di Turki pada tahun 2018 dengan mempertimbangkan nilai total produksi pertanian, total area yang ditanami, dan jumlah kredit pertanian yang digunakan. Studi ini menggunakan metode K-means clustering untuk menentukan hubungan antar provinsi, dan data dikumpulkan dari Badan Regulasi dan Pengawasan Perbankan (BRSA) dan Institut Statistik Turki. Metode pengelompokan K-means memungkinkan untuk mengevaluasi kinerja kredit, mengungkapkan kesamaan dan perbedaan dengan menggunakan nilai produksi pertanian, total lahan yang dibudidayakan, dan data volume kredit pertanian (Ceylan & Sabuncu, 2019)

Pentingnya pembiayaan di bidang pertanian untuk keberlanjutan produksi disoroti, karena kredit dan dukungan input secara langsung memengaruhi produksi pertanian. Selain itu, penelitian ini menggunakan teknologi data mining, khususnya algoritma pengelompokan K-means, untuk menganalisis data dalam jumlah besar dan mengekstrak informasi atau pola dari kumpulan data. Metode standarisasi normalisasi diterapkan untuk memastikan komparabilitas

antara data yang dikumpulkan, yang selanjutnya meningkatkan ketahanan analisis (Ceylan & Sabuncu, 2019)

Selain itu, ditekankan pula pentingnya kredit pertanian karena memungkinkan petani mengakses teknologi baru dan peluang ekonomi, yang pada akhirnya berkontribusi pada peningkatan produksi dan pendapatan di sektor pertanian. Penelitian ini juga membahas proses penentuan jumlah cluster (k) yang tepat untuk analisis, menyoroti perlunya k kurang dari jumlah objek dalam dataset dan memberikan wawasan untuk mengevaluasi nilai k menggunakan metode aglomeratif dan aturan praktis (Ceylan & Sabuncu, 2019)

Secara keseluruhan, penelitian ini menggunakan algoritma pengelompokan K-means untuk menganalisis kinerja kredit pertanian di Turki, dengan menekankan pentingnya pembiayaan pertanian, teknik penggalian data, dan proses penentuan jumlah cluster yang optimal untuk analisis.

E. Determination of Rice Quality Using the K-Means Clustering Method

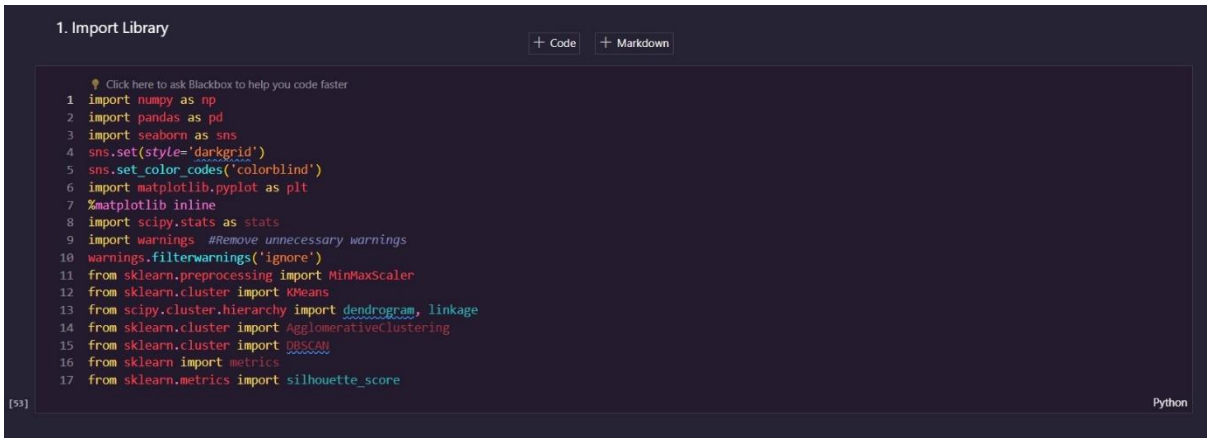
Makalah penelitian berjudul "Penentuan Kualitas Beras Menggunakan Metode K-Means Clustering" oleh Muhammad berfokus pada pemanfaatan model clustering K-Means untuk mengklasifikasikan beras berdasarkan kualitasnya, yang bertujuan untuk memudahkan pembeli dalam menilai kualitas beras dan membantu penjual dalam menentukan harga yang sesuai berdasarkan kualitas. Penelitian ini melibatkan penerapan algoritma pengelompokan K-Means pada 30 merek beras yang berbeda, yang menghasilkan identifikasi tiga cluster: kualitas sangat baik, kualitas baik, dan kualitas buruk, yang masing-masing terdiri dari sejumlah merek beras. Pendekatan ini memberikan metode sistematis untuk mengklasifikasikan beras dan menawarkan wawasan yang berharga bagi para pemangku kepentingan, terutama Bulog, untuk mengambil tindakan pencegahan dan mengatasi masalah yang berkaitan dengan penentuan kualitas beras (Fahlevi et al., 2020, p. 1).

Metode pengelompokan K-Means termasuk menghitung inisialisasi centroid, menghitung jarak pusat objek ke setiap centroid dengan menggunakan rumus jarak Euclidean, dan mengatur data setiap cluster berdasarkan jarak terdekat ke centroid. Selain itu, penelitian ini menekankan betapa pentingnya menetapkan standar kualitas beras, terutama untuk distributor gudang, untuk mengatasi kurangnya kesadaran konsumen tentang kualitas beras. Studi ini juga menekankan betapa pentingnya mengelompokkan beras berdasarkan kualitas menggunakan metode K-Means, karena algoritma ini dapat digunakan untuk berbagai tujuan (Fahlevi et al., 2020).

Selain itu, makalah ini menggarisbawahi penelitian sebelumnya tentang kualitas beras, yang termasuk penelitian tentang cara mengidentifikasi kualitas dan karakteristik beras di daerah tertentu, konversi lahan pertanian, dan kebijakan harga pembelian pemerintah untuk beras, yang masing-masing memiliki lebih banyak makna dan relevansi untuk topik ini. Selain itu, penelitian ini juga membahas masalah yang dihadapi konsumen dalam membedakan antara beras berkualitas baik dan beras berkualitas rendah (Fahlevi et al., 2020, p. 1).

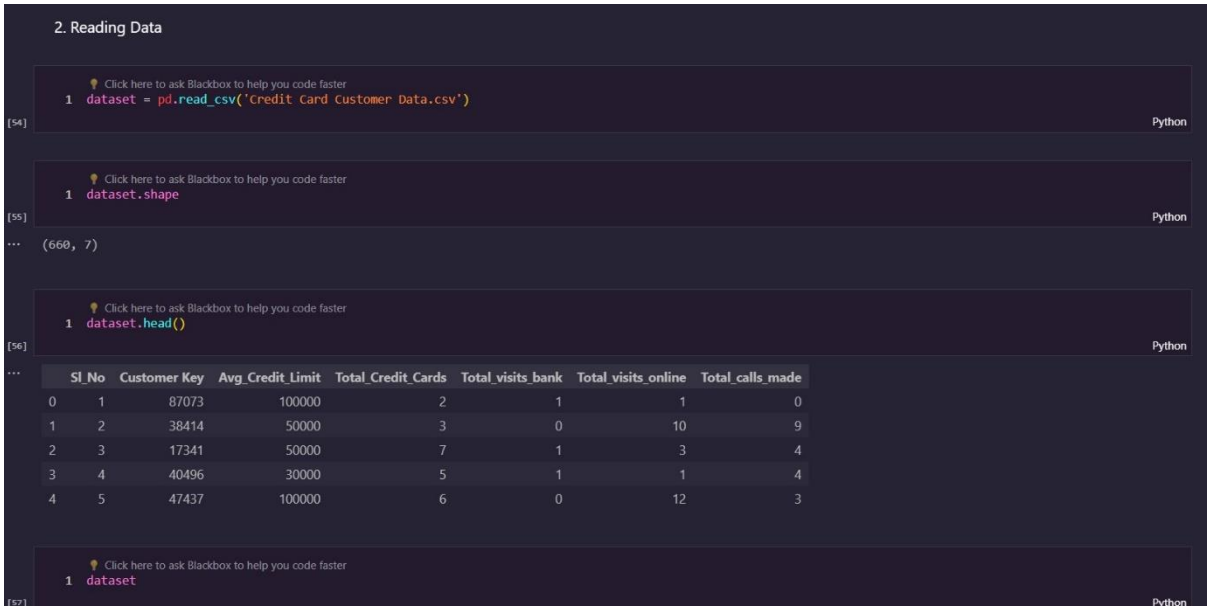
C. Hasil Pengolahan Data :

1.



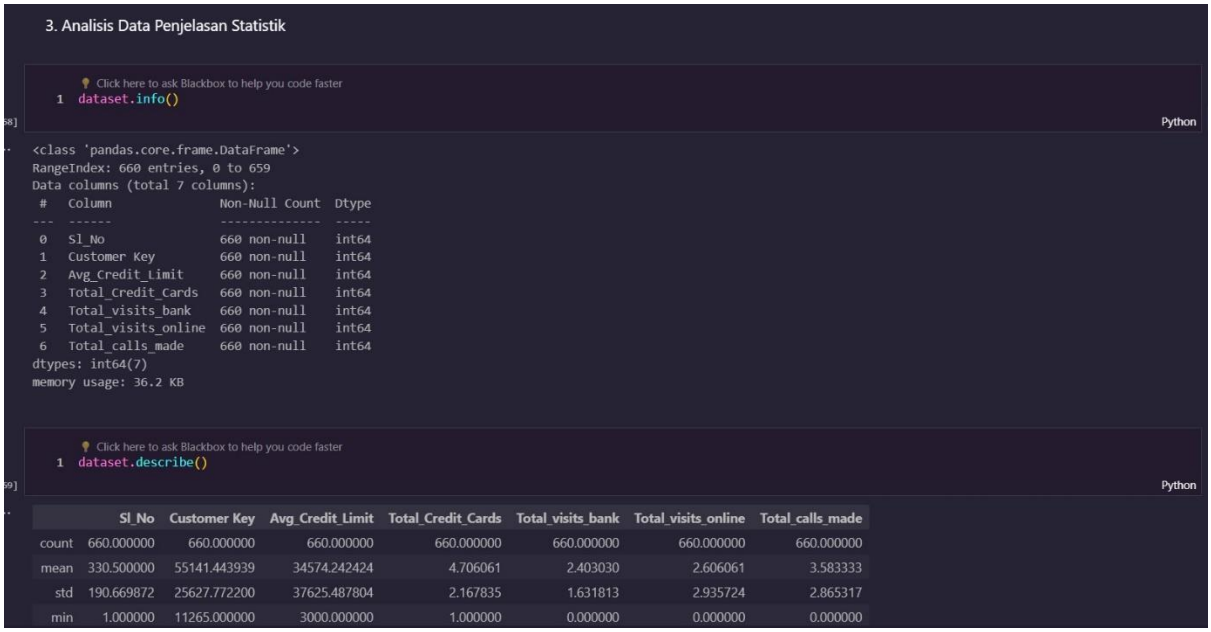
Proses di atas merupakan adalah proses import library di sini kita mengimport library yang sangat banyak dan di butuhkan untuk pengolahan data.

2.

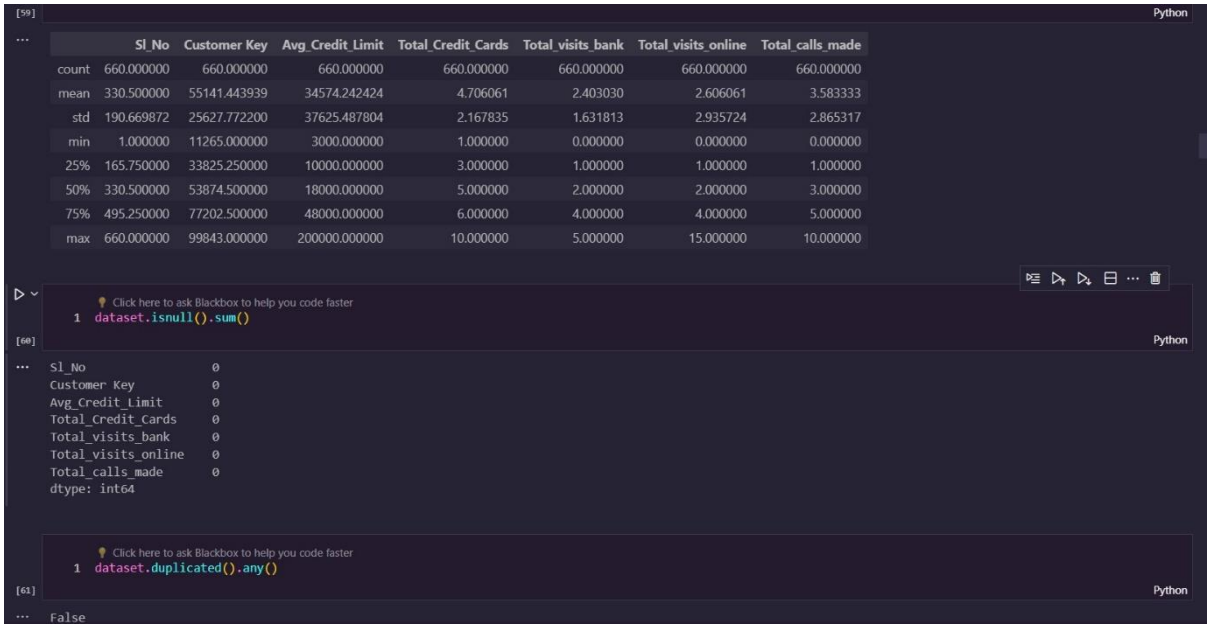


Proses di atas merupakan proses dalam membaca sebuah data yang di ambil dari dataset.

3.



Proses di atas adalah bagian untuk menganalisa dan mencari penjelasan statistic seputar dataset yang akan kita gunakan dan olah .



	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made
count	660.000000	660.000000	660.000000	660.000000	660.000000	660.000000	660.000000
mean	330.500000	55141.443939	34574.242424	4.706061	2.403030	2.606061	3.583333
std	190.669872	25627.772200	37625.487804	2.167835	1.631813	2.935724	2.865317
min	1.000000	11265.000000	3000.000000	1.000000	0.000000	0.000000	0.000000
25%	165.750000	33825.250000	10000.000000	3.000000	1.000000	1.000000	1.000000
50%	330.500000	53874.500000	18000.000000	5.000000	2.000000	2.000000	3.000000
75%	495.250000	77202.500000	48000.000000	6.000000	4.000000	4.000000	5.000000
max	660.000000	99843.000000	200000.000000	10.000000	5.000000	15.000000	10.000000

```
[59]
Click here to ask Blackbox to help you code faster
1 dataset.isnull().sum()
[60]
```

```
Sl_No                 0
Customer Key          0
Avg_Credit_Limit      0
Total_Credit_Cards    0
Total_visits_bank     0
Total_visits_online   0
Total_calls_made      0
dtype: int64
```

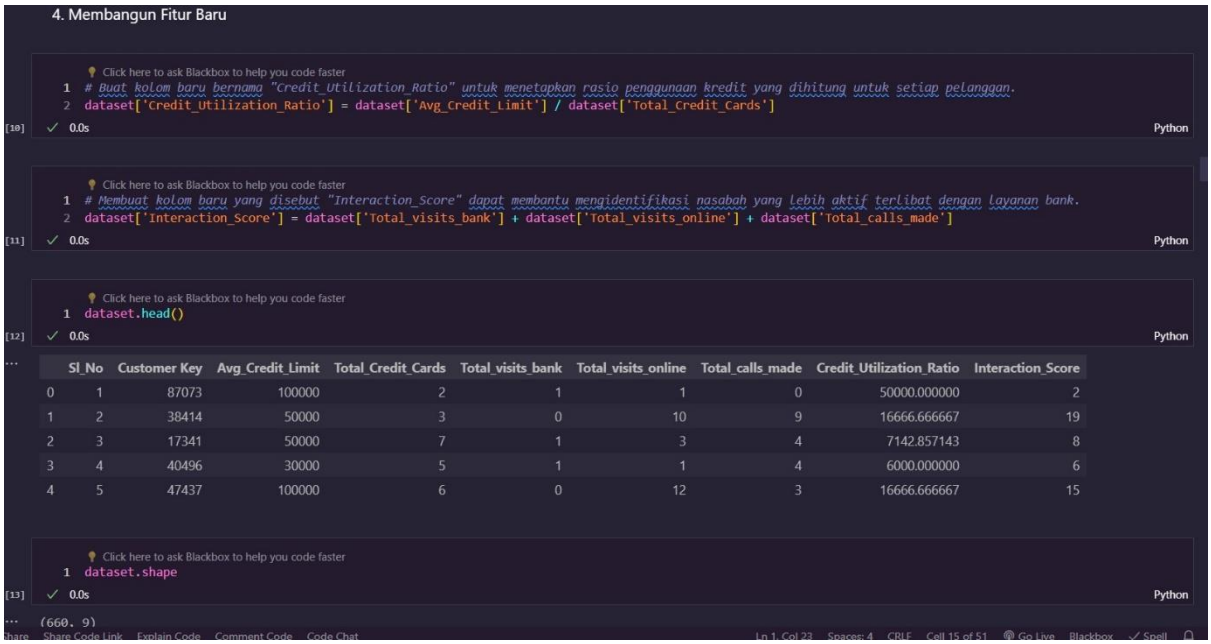
```
Click here to ask Blackbox to help you code faster
1 dataset.duplicated().any()
[61]
```

```
False
```

Proses di atas adalah untuk mengetahui apakah ada data kosong pada kolom – kolom yang ada di dataset tersebut. Untuk yang bawah nya adalah untuk memeriksa apakah ada data yang duplikat atau tidak.

Temuan : tidak ada data yang kosong pada kolom – kolom dataset dan juga tidak ada data yang duplikat pada dataset.

4.



4. Membangun Fitur Baru

```
Click here to ask Blackbox to help you code faster
1 # Buat kolom baru bernama "Credit_Utilization_Ratio" untuk menetapkan rasio penggunaan kredit yang dihitung untuk setiap pelanggan.
2 dataset['Credit_Utilization_Ratio'] = dataset['Avg_Credit_Limit'] / dataset['Total_Credit_Cards']
[10] ✓ 0.0s
```

```
Click here to ask Blackbox to help you code faster
1 # Membuat kolom baru yang disebut "Interaction_Score" dapat membantu mengidentifikasi nasabah yang lebih aktif terlibat dengan layanan bank.
2 dataset['Interaction_Score'] = dataset['Total_visits_bank'] + dataset['Total_visits_online'] + dataset['Total_calls_made']
[11] ✓ 0.0s
```

```
Click here to ask Blackbox to help you code faster
1 dataset.head()
[12] ✓ 0.0s
```

	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made	Credit_Utilization_Ratio	Interaction_Score
0	1	87073	100000	2	1	1	0	50000.000000	2
1	2	38414	50000	3	0	10	9	16666.666667	19
2	3	17341	50000	7	1	3	4	7142.857143	8
3	4	40496	30000	5	1	1	4	6000.000000	6
4	5	47437	100000	6	0	12	3	16666.666667	15

```
Click here to ask Blackbox to help you code faster
1 dataset.shape
[13] ✓ 0.0s
```

```
(660, 9)
```

Temuan : di sini kita membangun fitur baru untuk mengolah dataset tersebut agar bisa di clustersasi menjadi beberapa kelompok di sini kita membuat 2 kolom di samping kanan total_calls_made yatu Credit_Utilization_Ratio dan Interaction_Score.

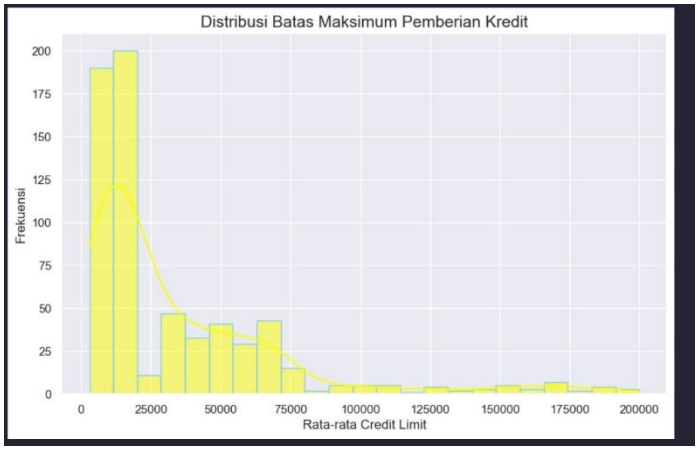
5.

5. Visulisasi Data

```
Click here to ask Blackbox to help you code faster
fig, ax = plt.subplots(figsize=(10, 6))

sns.histplot(data=dataset, x='Avg_Credit_Limit', kde=True, ax=ax, color='yellow', edgecolor='skyblue', linewidth=1)

ax.set_title('Distribusi Batas Maksimum Pemberian Kredit', fontsize=15)
ax.set_xlabel('Rata-rata Credit Limit', fontsize=12)
ax.set_ylabel('Frekuensi', fontsize=12)
ax.grid(True)

plt.show()
```

14] ✓ 0.4s Python

Proses di atas adalah untuk memvisulasasi data olah ke dalam diagram berikut hasil nya.

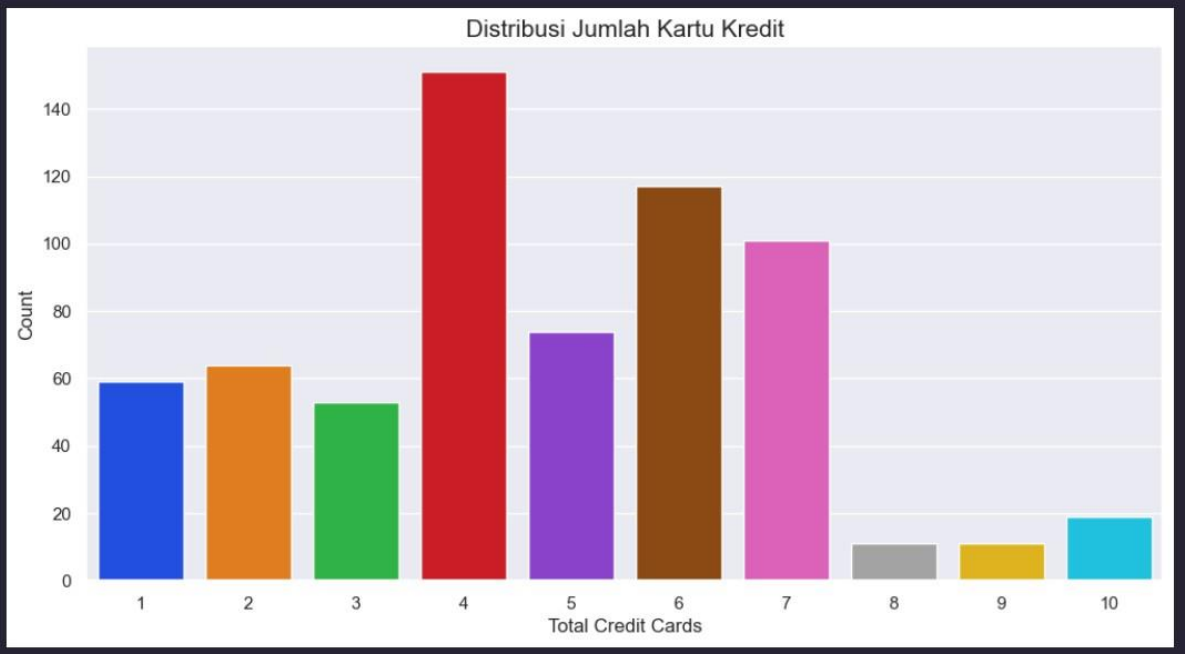


- Distribusi batas maksimum pemberian kredit (rata-rata credit limit) dalam dataset ini terlihat memusat pada satu titik (unimodal) dan sedikit miring ke kanan. Artinya, kebanyakan nilai credit limit berkumpul di sekitar nilai tertentu, dengan beberapa nilai yang lebih tinggi memanjang ke kanan.
- Puncak: Titik tertinggi distribusi, yang menunjukkan kumpulan terbanyak credit limit, berada sekitar 75.000. Ini menandakan bahwa batas kredit yang paling sering diberikan berada dalam kisaran ini.
- Rentang: Batas kredit dalam dataset ini berkisar dari 0 hingga sekitar 200.000.
- Frekuensi: Frekuensi tertinggi, yang diwakili oleh batang tertinggi, berada sekitar 175. Ini berarti bahwa sekitar 175 batas kredit berada dalam kisaran yang paling umum (sekitar 75.000).

6.

```
Click here to ask Blackbox to help you code faster
fig, ax = plt.subplots(figsize=(12, 6))

sns.countplot(data=dataset, x='Total_Credit_Cards', ax=ax, palette='bright')

ax.set_title('Distribusi Jumlah Kartu Kredit', fontsize=15)
ax.set_xlabel('Total Credit Cards', fontsize=12)
ax.set_ylabel('Count', fontsize=12)

plt.show()
```

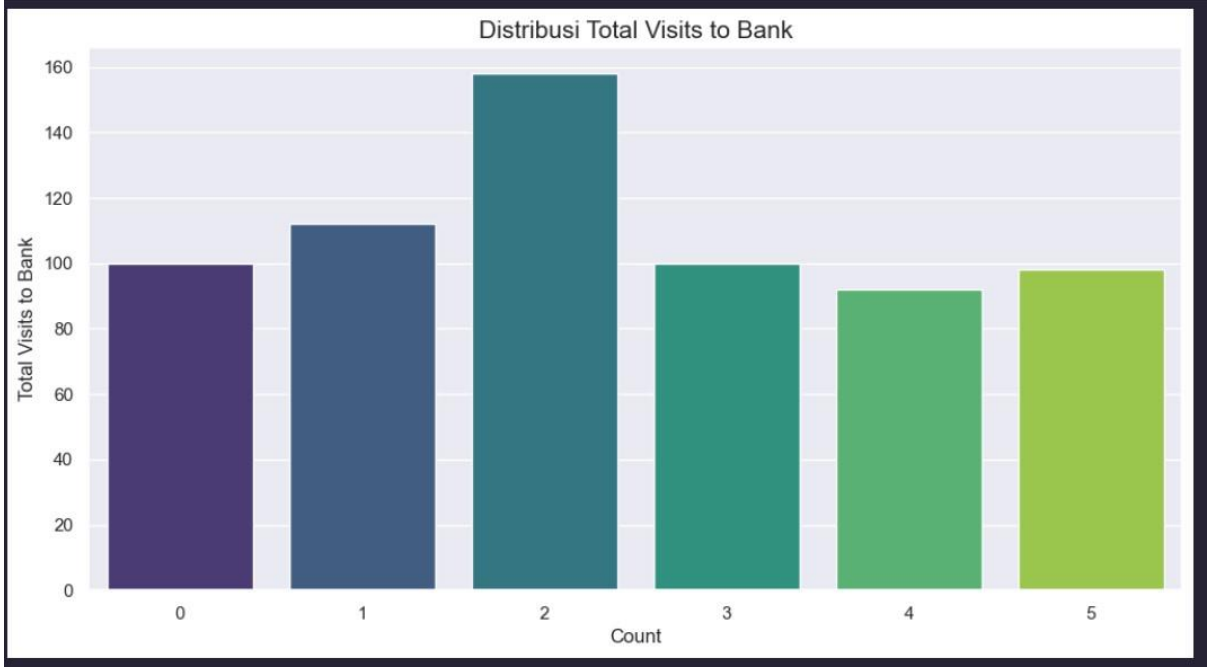
✓ 0.3s



Di atas adalah perbandingan total credit cards.

7.

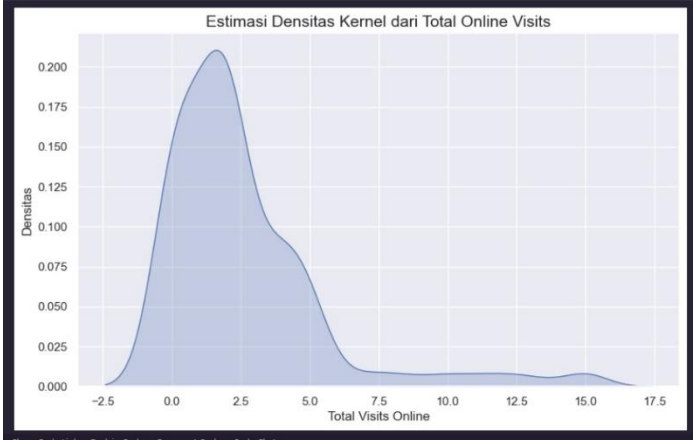
```
Click here to ask Blackbox to help you code faster
fig, ax = plt.subplots(figsize=(12, 6))

sns.countplot(data=dataset, x='Total_visits_bank', ax=ax, palette='viridis')

ax.set_title('Distribusi Total Visits to Bank', fontsize=15)
ax.set_xlabel('Count', fontsize=12)
ax.set_ylabel('Total Visits to Bank', fontsize=12)

plt.show()


✓ 0.2s
```



Perbandingan orang yang berkungjung ke bank berdasarkan beberapa level yaitu 0 sampai 5.

8.

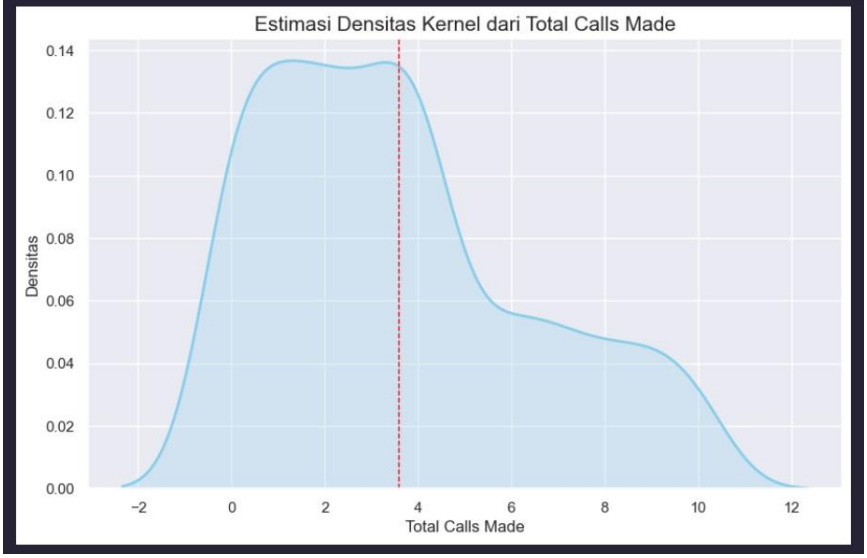
```
fig, ax = plt.subplots(figsize=(10, 6))

sns.kdeplot(data=dataset, x='Total_visits_online', ax=ax, fill=True)

ax.set_title('Estimasi Densitas Kernel dari Total Online Visits', fontsize=15)
ax.set_xlabel('Total Visits Online', fontsize=12)
ax.set_ylabel('Densitas', fontsize=12)
plt.show()


✓ 0.3s
```



Berikut adalah perbandingan jumlah visit online bank.

9.

```
Click here to ask Blackbox to help you code faster
fig, ax = plt.subplots(figsize=(10, 6))

sns.kdeplot(data=dataset, x='Total_calls_made', ax=ax, fill=True, color='skyblue', linewidth=2)
plt.axvline(dataset['Total_calls_made'].mean(), color='red', linestyle='dashed', linewidth=1)

ax.set_title('Estimasi Densitas Kernel dari Total Calls Made', fontsize=15)
ax.set_xlabel('Total Calls Made', fontsize=12)
ax.set_ylabel('Densitas', fontsize=12)

plt.show()
✓ 0.2s
```



- Perbandingan Total Calls Made atau panggilan yang di buat oleh bank itu berikut sedikit penjelasanya :
Kurva: Garis biru halus mewakili kepadatan probabilitas yang diperkirakan. Titik tertingginya, sekitar 2 panggilan, menunjukkan bahwa jumlah panggilan ini adalah yang paling sering dalam data Anda. Saat kita menjauh dari puncak ini, kurva secara bertahap turun, menunjukkan bahwa nilai yang lebih jauh dari 2 panggilan menjadi semakin jarang terjadi.
- Sumbu x: Ini menunjukkan rentang nilai yang mungkin untuk total jumlah panggilan yang dilakukan. Dalam kasus Anda, tampaknya berkisar dari sekitar -2 hingga 12 panggilan.
- Sumbu y: Ini mewakili kepadatan probabilitas yang diperkirakan. Nilai yang lebih tinggi pada sumbu y menunjukkan bahwa nilai yang sesuai pada sumbu x lebih mungkin terjadi dalam data Anda.
- Garis putus-putus merah: Garis vertikal ini menandai jumlah panggilan rata-rata yang dilakukan dalam data Anda. Dalam kasus ini, tampaknya sedikit di atas 0 panggilan.

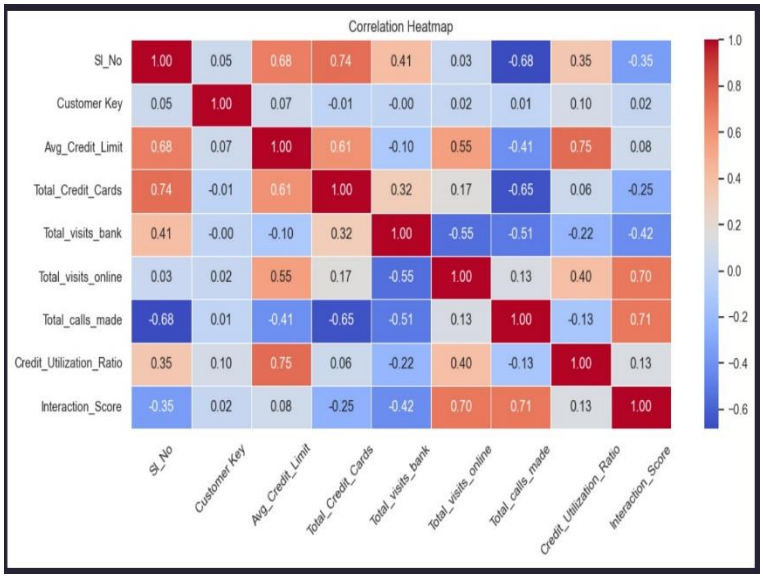
10.

6. Fitur yang di perlukan

```
Click here to ask Blackbox to help you code faster
Corr = dataset.corr()
✓ 0.0s
```

```
Click here to ask Blackbox to help you code faster
plt.figure(figsize=(12, 6))
sns.heatmap(Corr, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

plt.title('Correlation Heatmap')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()

plt.show()
✓ 0.5s
```

Berikut adalah proses untuk menjalankan fitur – fitur yang di perlukana untuk pengolahan data yaitu mengecek korelasi antar data dan juga memanggil diagram heatmap berikut penjelasan output nya :

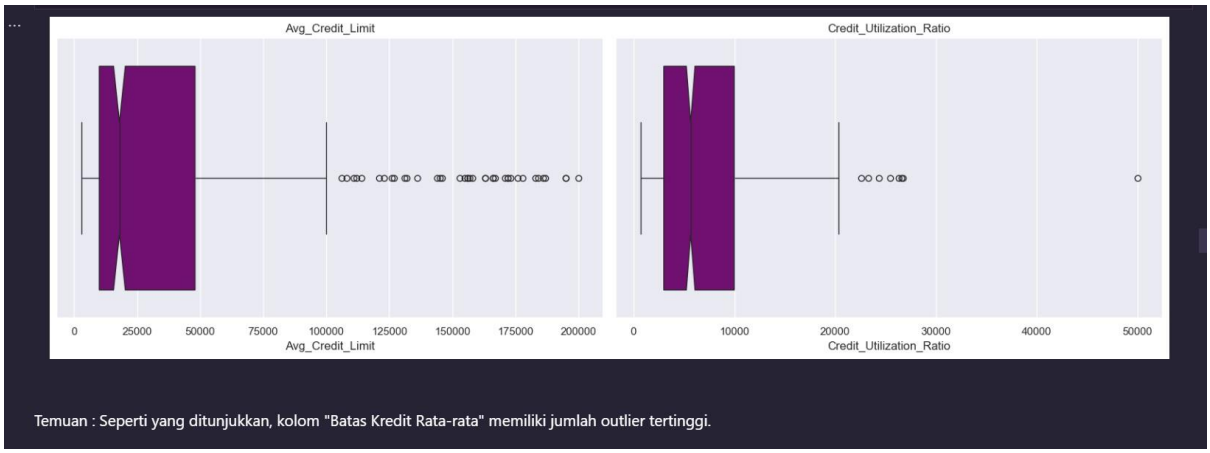


Berdasarkan matriks korelasi tersebut, dapat dilihat bahwa terdapat beberapa hubungan yang signifikan antara variabel-variabel yang ditampilkan. Beberapa hubungan tersebut adalah:

- Avg_Credit_Limit dan Total_Credit_Cards memiliki hubungan positif yang kuat. Artinya, semakin tinggi batas kredit rata-rata nasabah, semakin banyak kartu kredit yang dimiliki nasabah tersebut.
- Avg_Credit_Limit dan Credit_Utilization_Ratio memiliki hubungan negatif yang kuat. Artinya, semakin tinggi batas kredit rata-rata nasabah, semakin rendah rasio pemanfaatan kredit nasabah tersebut.
- Total_visits_bank dan Total_calls_made memiliki hubungan positif yang kuat. Artinya, semakin banyak nasabah mengunjungi bank, semakin banyak pula nasabah tersebut melakukan panggilan telepon.

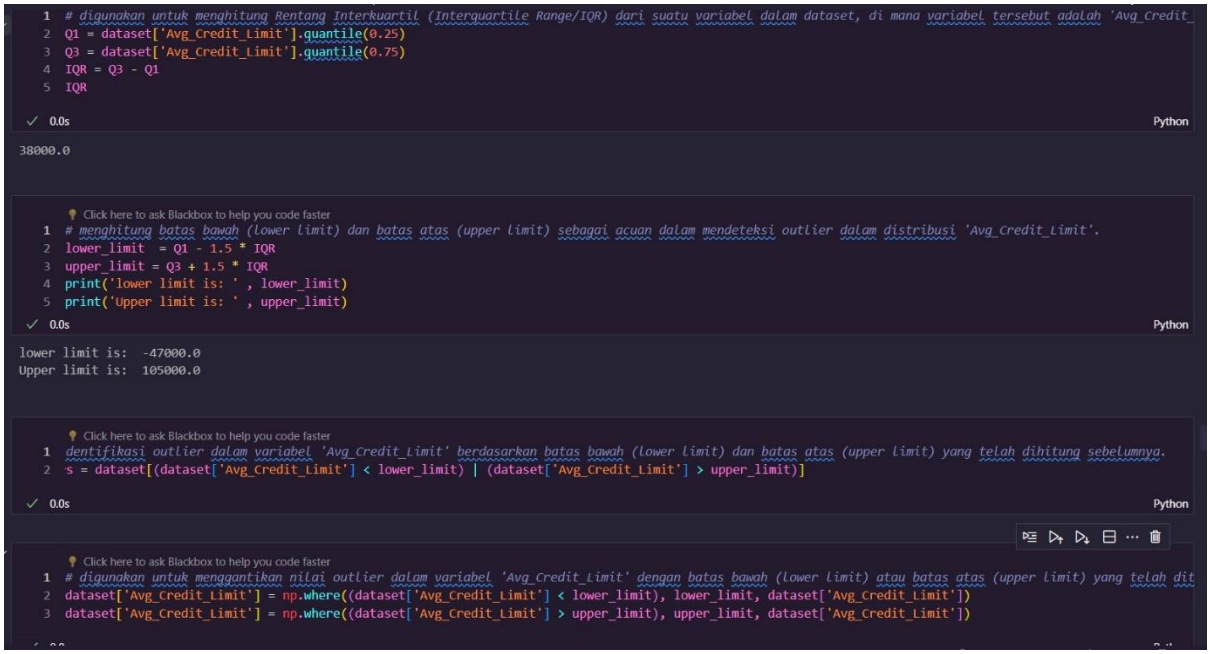
11.

Proses di atas adalah untuk mengecek outiler menggunakan box plot



Dari temuan di atas menggunakan perbandingan yaitu menggunakan data yang ada di rata – rata credit limi dan credit utilisasi ratio pada berbandingan tersebut terdapat beberapa outiler yang harus di hadapi sebelum penerapan model K-Means

Proses di bawah masih masuk ke dalam tahap ke 7



Pada tahap di atas code yang berawalan Q1 = dataset proses tersebut di gunakan untuk menghitung IQR dari suatu variabel dalam dataset, di semua variabel dan data yang di gunakan adalah.

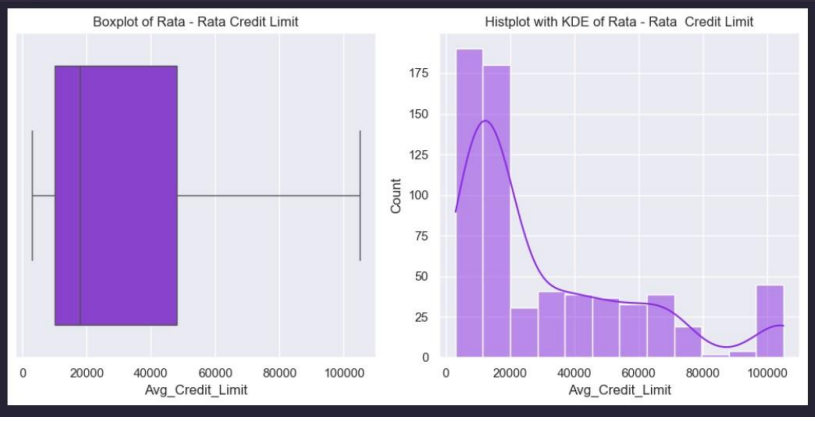
Pada code yang menghasilkan output lower limit dan upper limit adalah tahap proses untuk menghitung batas bawah dan batas atas data, sebagai acian dalam mendeteksi outiler dalam distribusi rata – rata credit limit.

Dan untuk code outilers = datset berguna untuk mengidentifikasi outiler pada varibel ‘Avg_Credit_Limit’ berdasarkan batas bawah dan batas atas yang telah di hitung sebelumnya.

Untuk bagian akhir pada foto untuk melakukan tahap yaitu menggatikan nilai outiler pada variabel dengan batas atas dan batas bawah yang telah di hitung.

```
Click here to ask Blackbox to help you code faster
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))

color = "#8A2BE2"

sns.boxplot(x=dataset["Avg_Credit_Limit"], ax=axes[0], color=color)
axes[0].set_title("Boxplot of Rata - Rata Credit Limit")

sns.histplot(x=dataset["Avg_Credit_Limit"], ax=axes[1], color=color, kde=True)
axes[1].set_title("Histplot with KDE of Rata - Rata  Credit Limit")

plt.tight_layout()
plt.show()
```

✓ 0.5s



Bagian di samping masih bagian dari tahap ke 7 yaitu hasil dari penhilangan outiler

8.

8. Scalling pada data yang sudah di hilangkan outiler agar nanti dapat di terapkan model K-Means

```
Click here to ask Blackbox to help you code faster
# adalah alat yang berguna untuk mengubah fitur-fitur dalam suatu dataset untuk membuat normalisasi data dan mengatasi masalah skala pada data
scaler = MinMaxScaler()
```

[51] ✓ 0.0s Python

```
Click here to ask Blackbox to help you code faster
# di gunakan untuk menyesuaikan kolom avg_credit_limit dan credit_utilization_ration dengan menggunakan scaler fit trasnform
dataset['Avg_Credit_Limit'] = scaler.fit_transform(dataset[['Avg_Credit_Limit']])
dataset['Credit_Utilization_Ratio'] = scaler.fit_transform(dataset[['Credit_Utilization_Ratio']])
```

[52] ✓ 0.0s Python

```
Click here to ask Blackbox to help you code faster
# tahap untuk menghitung beberapa statistik agregat (aggregated statistics) dari dua kolom, yaitu 'Avg_Credit_Limit' dan 'Credit_Utilization_Ratio', pada su
agg_dataset = dataset.agg({'Avg_Credit_Limit': ['mean', 'min', 'max'], 'Credit_Utilization_Ratio': ['mean', 'min', 'max']})
agg_dataset
```

[61] ✓ 0.0s Python

	Avg_Credit_Limit	Credit_Utilization_Ratio
mean	0.281046	0.131598
min	0.000000	0.000000
max	1.000000	1.000000

Proses di atas merupakan tahap Scaling pada data untuk nanti nya akan di terapkan ke dalam model K-Means tahap ke 8 di bagi menjadi tiga yaitu :

- Tahap untuk berguna mengubah fitur – fitur dalam suatu dataset untuk menormalisasi data dan mengatasi masalah skala pada data.
- Tahap ke dua untuk menyesuaikan kolom avg_credit_limit dan avg_utilization_ratio dengan menggunakan scale.fit_transform.
- Tahap ke tiga merupakan untuk menghitung beberapa statistic aggresgat dari kolom avg_credit_limi dan avg_utilization_ratio.

9.

9. Penerapan Model K-Means

```
Click here to ask Blackbox to help you code faster
1 X = dataset.iloc[:,[2,7]]
2 X
```

✓ 0.0s Python

	Avg_Credit_Limit	Credit_Utilization_Ratio
0	0.950980	1.000000
1	0.460784	0.323671
2	0.460784	0.130435
3	0.264706	0.107246
4	0.950980	0.323671
...	...	...
655	0.941176	0.186377
656	0.794118	0.155942
657	1.000000	0.353261
658	1.000000	0.334493
659	1.000000	0.361997

660 rows × 2 columns

Tahap di atas merupakan Langkah awal dalam penerapan model K-Means yaitu proses untuk mengambil 2 kolom tertentu dari dataframe berdasarkan posisi index-nya, dan menampung hasilnya ke dalam variabel X, lalu mencetak variabel X.

```
Click here to ask Blackbox to help you code faster
1 wcss = []
2 for i in range(1, 11):
3     kmeans = KMeans(n_clusters = i, init = 'k-means++')
4     kmeans.fit(X)
5     print('Cost_Function=',kmeans.inertia_,'with', i, 'Clusters')
6     wcss.append(kmeans.inertia_)
```

✓ 0.5s Python

```
Cost_Function= 59.54274865285163 with 1 Clusters
Cost_Function= 16.623563596494083 with 2 Clusters
Cost_Function= 8.938520171758897 with 3 Clusters
Cost_Function= 5.820590878393672 with 4 Clusters
Cost_Function= 4.105650434344396 with 5 Clusters
Cost_Function= 3.328901010425862 with 6 Clusters
Cost_Function= 2.731753360832538 with 7 Clusters
Cost_Function= 2.288166154525652 with 8 Clusters
Cost_Function= 1.9674007641605988 with 9 Clusters
Cost_Function= 1.6705257688680617 with 10 Clusters
```

Proses di atas masih termasuk ke tahap 9 yaitu mencari jumlah kluster optimal dengan metode Elbow dengan melihat nilai WCSS untuk setiap jumlah kluster yang diiterasi.

Penjelasan output nya :

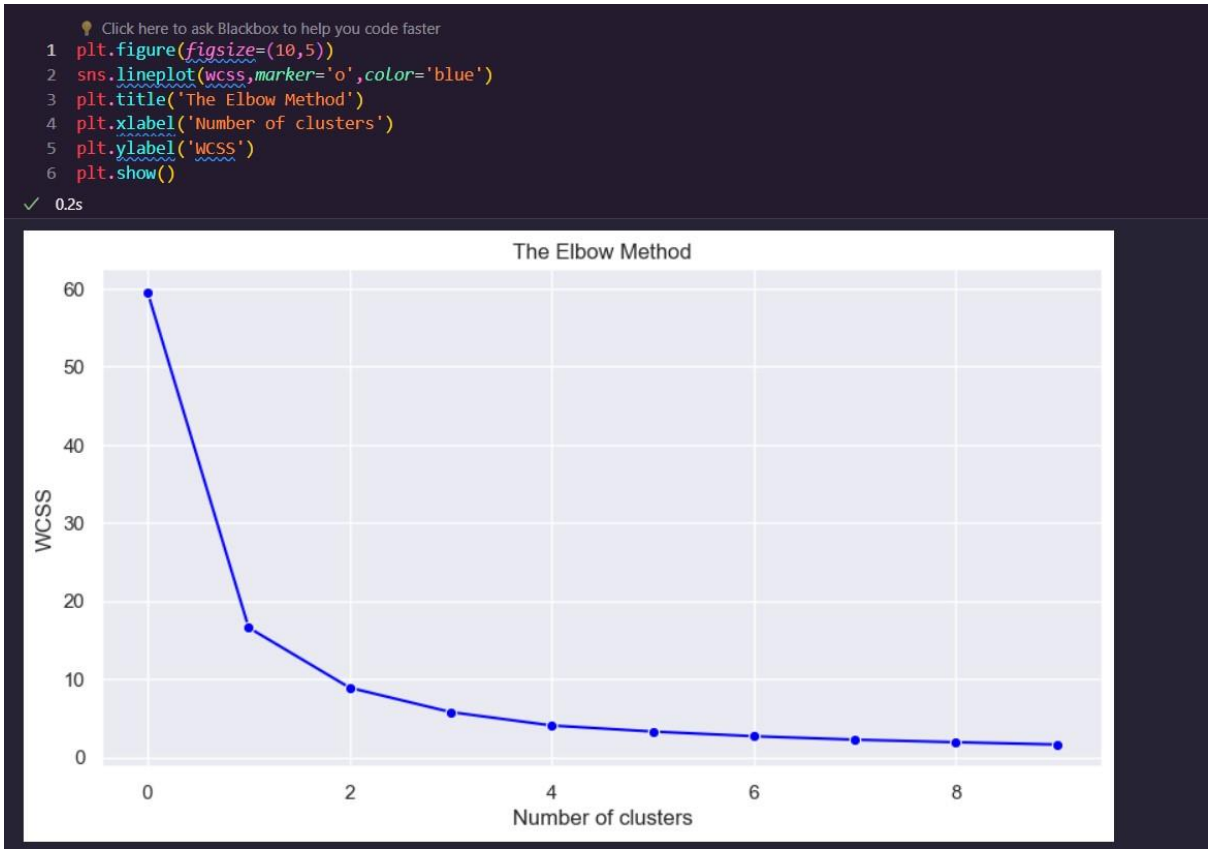
- Cost_Function adalah nilai Within Cluster Sum of Squares (WCSS) dari model k-means yang dilatih.

- Nilai WCSS menunjukkan variasi total dalam suatu kluster. Semakin kecil nilai WCSS, semakin baik.

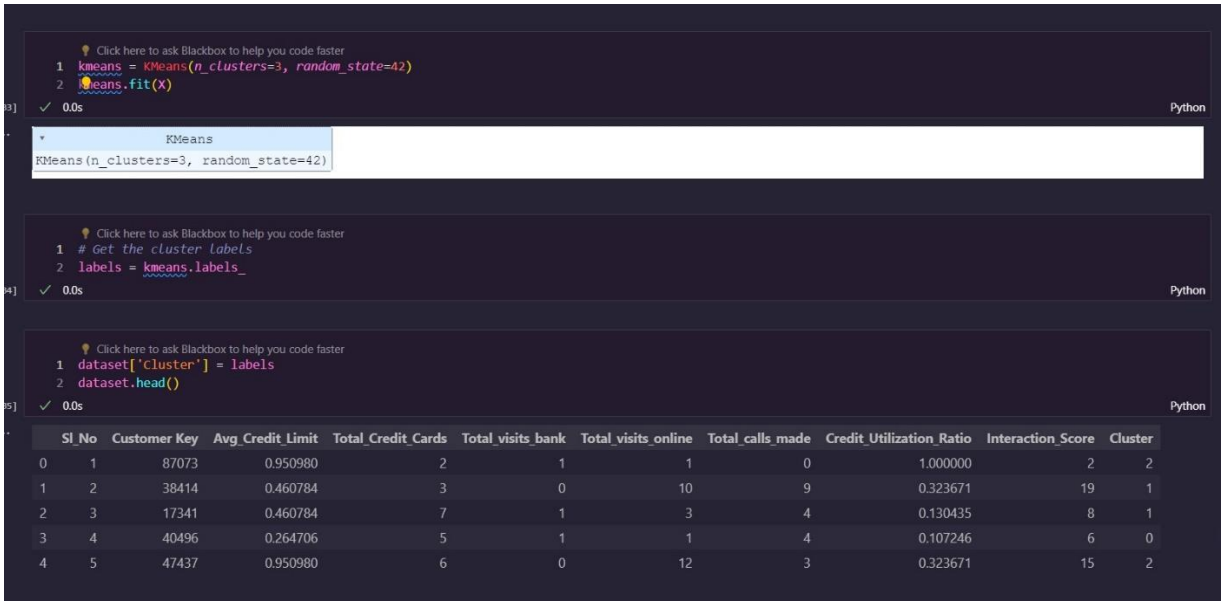
- Dari output terlihat nilai Cost_Function (WCSS) semakin menurun seiring bertambahnya jumlah kluster. Hal ini wajar karena semakin banyak kluster, semakin kecil variasi di dalam tiap kluster.
- Kluster optimal biasanya berada di titik "siku" grafik nilai WCSS, dimana penurunan nilai WCSS mulai melambat. Dari pola output ini, kemungkinan kluster optimal ada di kisaran 3-6 kluster.

- Kluster optimal sebenarnya perlu dianalisis lebih lanjut dengan melihat elbow plot, dan mempertimbangkan domain permasalahan. Output ini hanya memberikan indikasi awal.

- Output ini berguna untuk mendapatkan gambaran nilai WCSS untuk berbagai percobaan jumlah kluster. Digunakan sebagai awal analisis untuk menentukan jumlah kluster optimal.



Tahap di atas adalah Elbow Method temuanya adalah jumlah cluster yang optimal ada 3.



	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made	Credit_Utilization_Ratio	Interaction_Score	Cluster
0	1	87073	0.950980	2	1	1	0	1.000000	2	2
1	2	38414	0.460784	3	0	10	9	0.323671	19	1
2	3	17341	0.460784	7	1	3	4	0.130435	8	1
3	4	40496	0.264706	5	1	1	4	0.107246	6	0
4	5	47437	0.950980	6	0	12	3	0.323671	15	2

Proses di atas adalah Membuat model KMeans dengan 3 kluster dan random state tertentu dan Melatih model tersebut dengan data X sehingga data X terklaster ke dalam 3 kelompok/kluster. Dengan tujuan adalah melakukan proses clustering data menggunakan algoritma k-means dengan jumlah kluster yang sudah ditentukan sebelumnya yaitu 3 kluster.

Pada bagian ke dua yaitu labels = kmeans.labels_ untuk Fungsinya adalah untuk mendapatkan label kluster dari setiap data yang diprediksi oleh model KMeans.

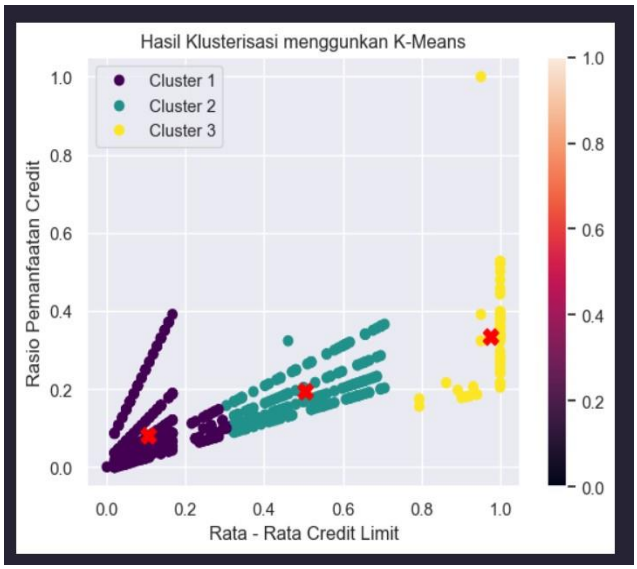
Pada bagian ke 3 pada gambar adalah tahap mengisi kolom baru pada dataframe dataset dengan nama 'Cluster'. Kolom ini diisi dengan data labels yang berisi label kluster untuk setiap data. Yang kegunaan nya adalah untuk melihat secara langsung hasil klusterisasi data berdasarkan model KMeans yang telah dilatih. Dengan menambahkan kolom Cluster pada dataset, kita dapat melihat kluster mana yang didapat oleh setiap data.

```
Click here to ask Blackbox to help you code faster
centroids = kmeans.cluster_centers_
scatter = plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=labels, cmap='viridis')
plt.scatter(centroids[:, 0], centroids[:, 1], marker='X', color='red', s=100, label='Centroids')

# Create custom legend labels for each cluster
legend_labels = [f'Cluster {i+1}' for i in range(3)]
plt.legend(handles=scatter.legend_elements()[0], labels=legend_labels)

plt.xlabel('Rata - Rata Credit Limit')
plt.ylabel('Rasio Pemanfaatan Credit')
plt.title('Hasil Klusterisasi menggunkan K-Means')
plt.colorbar()
```

✓ 0.4s Python

```
<matplotlib.colorbar.Colorbar at 0x1c763f5bfa0>
```

Tahap di atas adalah termasuk tahap ke 9 juga di mana tahap tersebut kita melakukan kode yang digunakan untuk memvisualisasikan hasil klastering secara visual dengan scatter plot, sehingga pola dan performa klastering lebih mudah diinterpretasi.



- Klaster 1: Pelanggan dengan batas kredit dan utilisasi kredit rendah dengan rasio pemanfaat credit 0.4 dan rata – rata credit limit sampai 0.2.

- Klaster 2: Pelanggan dengan batas kredit dan utilisasi kredit sedang, dengan rasion pemanfaatan credit sampai 0.4 juga namun rata – rata credit limit nya sangat besar mencapai 0.6 poin.

- Klaster 3: Pelanggan dengan batas kredit dan utilisasi kredit tinggi, untuk cluster ini pemanfaatan credit nya cukup tingi yaitu hamper menyentuh angka 0.6 sedangkan level rata credit nya yaitu maksimal di angka 1.0 poin.

Kesimpulan :

Studi kasus ini menggunakan metode pembelajaran mesin tanpa pengawasan yang dikenal sebagai clustering K-means algoritma untuk mengelompokkan pelanggan kartu kredit ke dalam klaster-klaster dengan karakteristik yang sebanding. Tujuannya adalah untuk membantu bank memahami bagaimana berbagai segmen pelanggan berperilaku, sehingga mereka dapat membuat keputusan kredit yang lebih baik dan membuat produk yang lebih sesuai dengan kebutuhan pelanggan. Pra-pemrosesan data pelanggan kartu kredit terdiri dari pengukuran data untuk normalisasi dan pengecekan nilai yang hilang dan outlier. Dengan menggunakan metode Elbow, nilai Within Cluster Sum of Squares (WCSS) dihitung untuk menghasilkan tiga klaster yang ideal. Hasilnya membagi klien ke dalam tiga kelompok utama: klaster dengan batas kredit dan utilisasi kredit rendah, klaster dengan batas kredit dan utilisasi kredit sedang, dan klaster dengan batas kredit dan utilisasi kredit tinggi. Secara keseluruhan, studi kasus ini menunjukkan bahwa clustering K-means dapat mengekstraksi wawasan penting dari data untuk tujuan segmentasi nasabah yang lebih baik. Ini dapat digunakan oleh bank untuk membuat strategi pemasaran dan produk yang lebih baik yang sesuai dengan karakteristik masing-masing klaster nasabah.

Daftar Pustaka

1. Ceylan, Z., & Sabuncu, S. (2019). Analysis of Agricultural Credit Performance of Turkey using K-means Clustering Algorithm. *European Journal of Science and Technology*, 478–484. https://doi.org/10.31590/ejosat.638434
2. Chen, R., Wang, S., Zhu, Z., Yu, J., & Dang, C. (2023). Credit ratings of Chinese online loan platforms based on factor scores and K-means clustering algorithm. *Journal of Management Science and Engineering*, *8*(3), 287–304. https://doi.org/10.1016/j.jmse.2022.12.003
3. Fahlevi, M. R., Putri, D. R. D., Putri, F. A., Rahman, M., Sipahutar, L., & Muhatri, M. (2020, October 27). Determination of Rice Quality Using the K-Means Clustering Method. *2020 2nd International Conference on Cybernetics and Intelligent System, ICORIS 2020*. https://doi.org/10.1109/ICORIS50180.2020.9320839
4. Lin, Z., Xingzhong, B., Yajun, C., Ting, W., Fei-Hu, H., Mingzhu, L., & Jian, P. (2020). Research on Power Market User Credit Evaluation Based on K-Means Clustering and Contour Coefficient. *2020 3rd International Conference on Robotics, Control and Automation Engineering, RCAE 2020*, 64–68. https://doi.org/10.1109/RCAE51546.2020.9294725
5. National Institute of Technology (Punjab, I., National Institute of Technology (Punjab, I. D. of C. S. & E., Institute of Electrical and Electronics Engineers. Delhi Section, & Institute of Electrical and Electronics Engineers. (n.d.). *ICSCCC 2018 : International Conference on Secure Cyber Computing and Communication : December 15-17, 2018*.