Kelompok 4

CREDIT CARD CUSTOMER CLUSTERING

Data Mining

LA08

Meet The Team

- Sarah Safitri - 2440027511
- Sharlene Regina - 2440032070
- Tricia Estella - 2440003695

Background of the Study

BACKGROUND OF STUDY

Kartu Kredit merupakan salah satu metode pembayaran yang dikeluarkan oleh lembaga keuangan.

Segmentasi pasar membagi pasar yang lebih besar menjadi kelompok konsumen yang lebih kecil dan lebih homogen berdasarkan karakteristik atau perilaku yang sama dan segmentasi pasar adalah strategi utama dalam pemasaran, karena memungkinkan bisnis menargetkan upaya pemasaran dan produk mereka secara lebih efektif ke kelompok konsumen tertentu.

BACKGROUND OF STUDY

Problem Definition:
Penggunaan kartu kredit yang meningkat menjadikan pelaksanaan analisis secara manual memakan banyak waktu sehingga pemanfaatan informasi untuk segmentasi pasar tidak tercapai.

Solution:
Penggunaan Algoritma K-Means untuk melakukan Clustering untuk mencari segmentasi pasar.

Analisis clustering ini berguna untuk meningkatkan efisiensi dan efektivitas proses segmentasi. Studi ini akan berusaha mengevaluasi keefektifan algoritma k-means untuk data kartu kredit, dan menilai potensi manfaat penggunaan k-means untuk segmentasi pasar.

Dataset

Dataset yang dipakai adalah dataset dari kaggle, dengan data sebanyak 9000 kartu kredit aktif selama 6 bulan terakhir. Data berfokus pada customer, dan terdapat 18 variabel:

Dataset

- **CUST_ID:** Identifikasi pemegang kartu kredit (Kategorikal)
- **BALANCE**: Banyak saldo yang tersisa di akun untuk membuat transaksi
- **BALANCE_FREQUENCY**: Seberapa sering balance terupdate, score terdiri dari 0 (not frequently updated) dan 1 (frequently updated)
- **PURCHASES**: Besar transaksi yang dilakukan
- **ONEOFF_PURCHASES**: Maksimal pembelian dalam satu transaksi
- **INSTALLMENTS_PURCHASES**: Banyaknya pembelian dengan cicilan
- **CASH_ADVANCE**: Uang muka yang diberikan user
- **PURCHASES_FREQUENCY**: Seberapa sering pembelian dilakukan, score terdiri dari 0 (not frequently updated) dan 1 (frequently updated)
- **ONEOFF_PURCHASES_FREQUENCY**: Seberapa sering pembelian dalam satu transaksi dilakukan, score terdiri dari 0 (not frequently updated) dan 1 (frequently updated)
- **PURCHASES_INSTALLMENTS_FREQUENCY**: Seberapa sering pembelian dengan cicilan dilakukan. score terdiri dari 0 (not frequently updated) dan 1 (frequently updated)
- **CASH_ADVANCE_FREQUENCY**: Seberapa sering uang muka dibayarkan
- **CASH_ADVANCE_TRX**: Banyaknya transaksi dengan uang muka
- **PURCHASES_TRX**: Banyaknya transaksi pembelian dilakukan
- **CREDIT_LIMIT**: Limit dari credit card untuk user
- **PAYMENTS**: Jumlah dari pembayaran yang telah dibayar user
- **MINIMUM_PAYMENTS**: Minimal pembayaran yang telah dibayar user
- **PRC_FULL_PAYMENT**: Persentase pembayaran penuh yang dibayarkan oleh user

BAGIAN 2: Preprocessing

Agar data dapat digunakan untuk membuat model yang baik, maka data tersebut harus dibuat menjadi baik juga, dengan cara preprocessing. Adapun kita harus mengetahui terlebih dahulu langkah preprocessing yang sesuai dengan data kita.

Column	Non-Null Count	Dtype
CUST_ID	8950 non-null	object
BALANCE	8950 non-null	float64
BALANCE_FREQUENCY	8950 non-null	float64
PURCHASES	8950 non-null	float64
ONEOFF_PURCHASES	8950 non-null	float64
INSTALLMENTS_PURCHASES	8950 non-null	float64
CASH_ADVANCE	8950 non-null	float64
PURCHASE_FREQUENCY	8950 non-null	float64
ONEOFF_PURCHASES_FREQUENCY	8950 non-null	float64
PURCHASES_INSTALLMENT_FREQUENCY	8950 non-null	float64
CASH_ADVANCE_FREQUENCY	8950 non-null	float64
CASH_ADVANCE_TRX	8950 non-null	int64
PURCHASES_TRX	8950 non-null	int64
CREDIT_LIMIT	8949 non-null	float64
PAYMENTS	8950 non-null	float64
MINIMUM_PAYMENTS	8637 non-null	float64
PRC_FULL_PAYMENT	8950 non-null	float64
TENURE	8950 non-null	int64

Dataset Exploratory

Menurut hasil df.info(),
variabel CREDIT_LIMIT (8949) dan MINIMUM_PAYMENTS (8637) memiliki data dengan NaN, yang perlu kita isi nantinya.

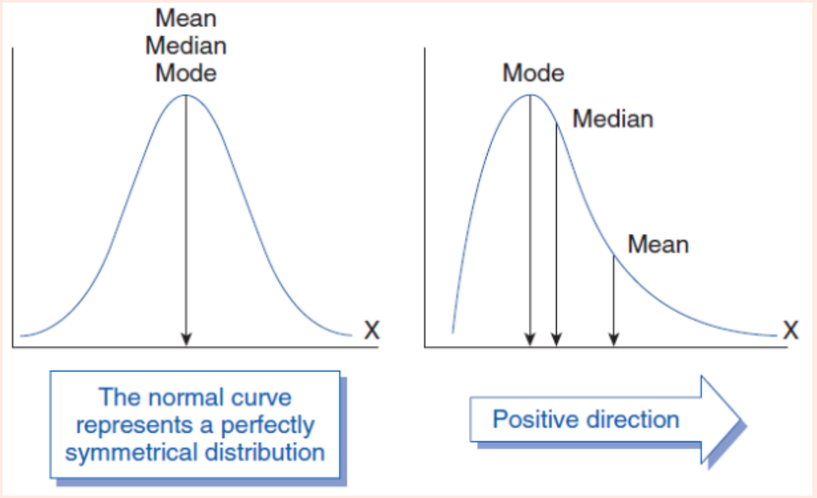
Hal tersebut dikarenakan count pada variabel tersebut lebih sedikit daripada baris dataset yang ada (8950)

	min	median	max	mean
BALANCE	0.00	873.385	19043.13	1564.47
BALANCE_FREQUENCY	0.00	1.00	1.0	0.87
PURCHASES	0.00	361.28	49039.57	1003.20
ONEOFF_PURCHASES	0.00	38.00	40761025	592.44
INSTALLMENTS_PURCHASES	0.00	89.00	22500.00	411.06
CASH_ADVANCE	0.00	0.00	47137.21	978.87
PURCHASE_FREQUENCY	0.00	0.50	1.00	0.49
ONEOFF_PURCHASES_FREQUENCY	0.00	0.083	1.00	0.02
PURCHASES_INSTALLMENT_FREQUENCY	0.00	0.167	1.00	0.364
CASH_ADVANCE_FREQUENCY	0.00	0.00	1.50	0.13
CASH_ADVANCE_TRX	0.00	0.00	123.00	3.25
PURCHASES_TRX	0.00	7.00	358.00	14.71
CREDIT_LIMIT	50.00	3000.00	30000.00	4494.45
PAYMENTS	0.00	856.901	50721.48	1733.14
MINIMUM_PAYMENTS	0.0192	312.34	76406.2001.00	864.21
PRC_FULL_PAYMENT	0.00	0.00	1.00	0.15
TENURE	6.00	12.00	12.00	11.52

Dataset Exploratory (2)

Menurut hasil df.describe(), hampir semua feature memiliki mean yang lebih besar daripada nilai median, yang berarti terdapat **skewness** pada dataset, yang bisa kita normalisasi.

Berdasarkan hasil penelusuran dataset, langkah preprocessing yang akan dilakukan antara lain adalah mengisi **nilai NULL** dan **normalisasi data.**



MENGISI NILAI NULL

CREDIT_LIMIT (8949)

```
df.dropna(subset=['CREDIT_LIMIT'], inplace=True)
```

Karena hanya ada satu data yang hilang pada CREDIT_LIMIT (0.01%), maka kita bisa **drop saja baris tersebut**. Persentase nilai NULL sangat sedikit, pembuangan record ini tidak begitu berarti pada model yang akan dibuat.

Oleh karena itu, record yang tersisa sekarang adalah 8949 baris.

MINIMUM_PAYMENTS (8637)

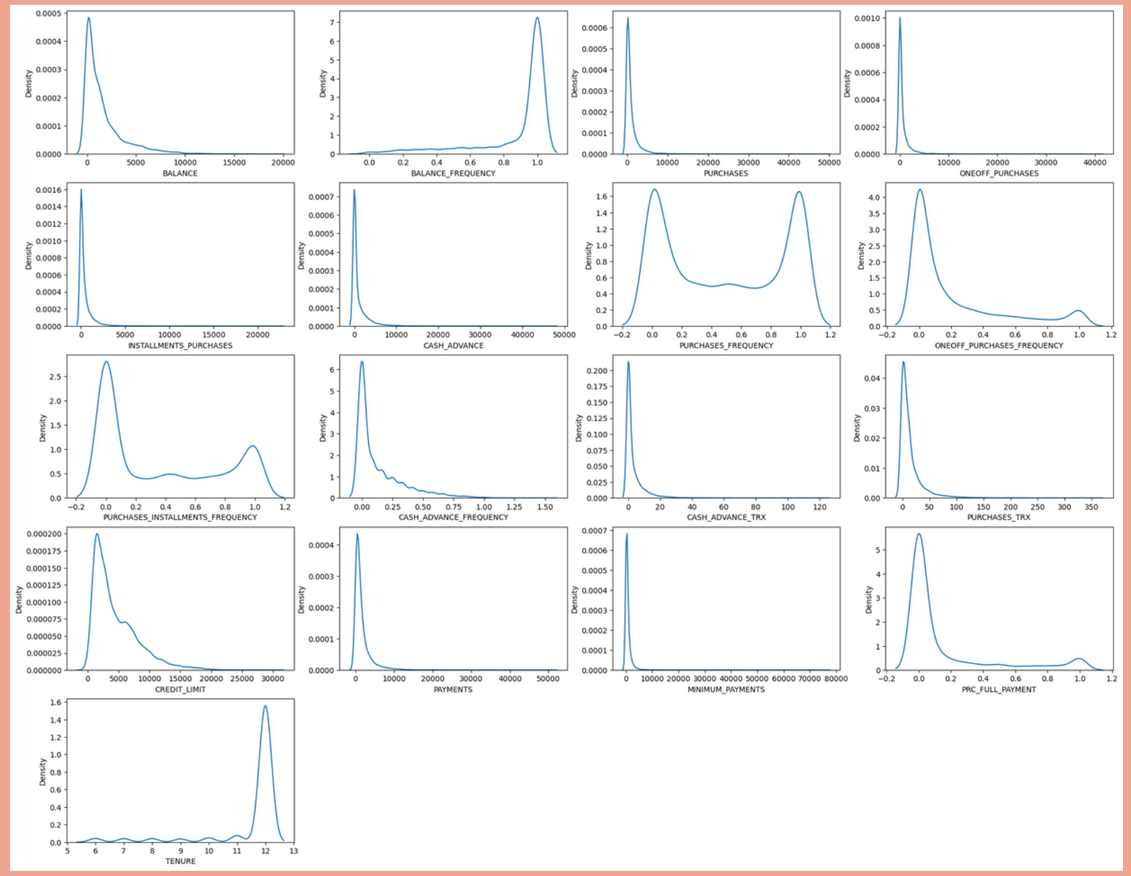
```
df['MINIMUM_PAYMENTS'].fillna(df['MINIMUM_PAYMENTS'].median(), inplace=True)
```

Pada feature MINIMUM_PAYMENTS, karena jumlah data NULL cukup besar (313 baris), maka akan dilakukan pengisian data **menggunakan nilai median.**

Hal ini dikarenakan tidak ada feature yang berelasi dengan feature ini untuk estimasi value yang hilang, dan juga data tersebut memiliki **skewness**, sehingga lebih baik untuk mereplace NaN dengan median. Median akan memberikan estimasi central tendency yang lebih baik.

NORMALISASI

Menentukan langkah normalisasi yang tepat



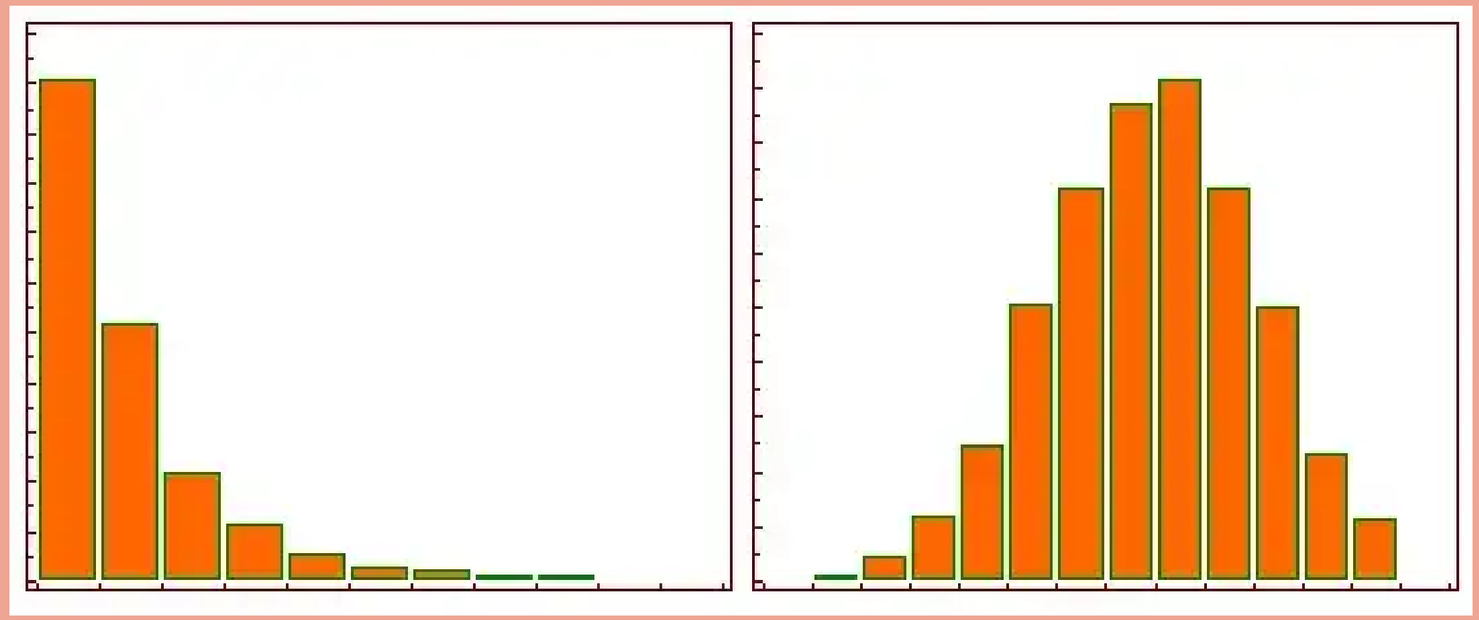
Setiap variabel memiliki **skewness yang bervariasi.** Untuk menghasilkan visualisasi cluster yang baik, kita dapat melakukan transformasi data menggunakan rumus logaritma (**Logarithmic transformation**).

Transformasi ini cocok digunakan untuk data dengan **heavily skewed feature** dan diubah menjadi lebih dekat dengan normal distribution.

Transformasi ini akan **menyebar** data sehingga model dapat membedakan cluster satu dengan cluster lain, dengan tetap mempertahankan value outliers. Analisis statistikal pada data skew yang sudah mengikuti distribusi normal akan menjadi lebih akurat.

NORMALISASI

Logarithmic Transformation



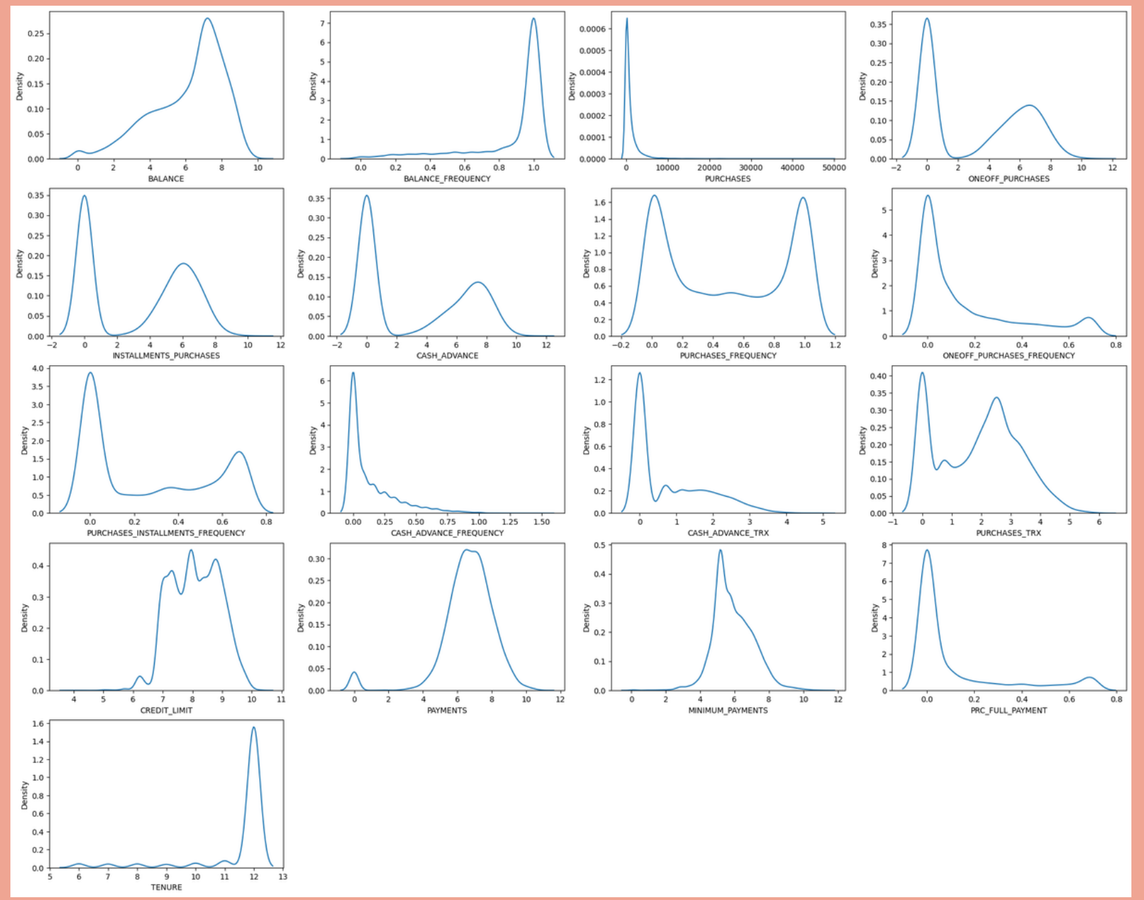
Log transformation mengurangi atau menghilangkan skewness data asli. Namun data asli harus mengikuti atau mendekati distribusi log-normal (positively skewed). Jika tidak, transformasi log tidak akan berfungsi.

NORMALISASI

Logarithmic Transformation

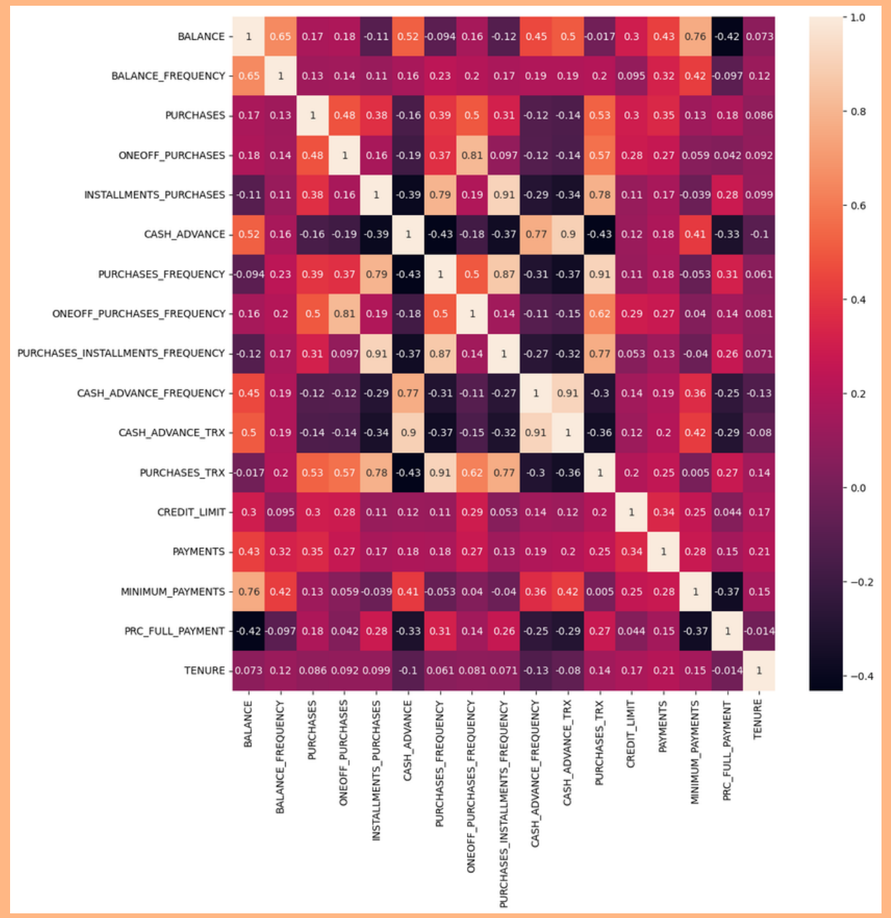
```
cols = ['BALANCE', 'ONEOFF_PURCHASES', 'INSTALLMENTS_PURCHASES',
        'CASH_ADVANCE', 'ONEOFF_PURCHASES_FREQUENCY',
        'PURCHASES_INSTALLMENTS_FREQUENCY', 'CASH_ADVANCE_TRX',
        'PURCHASES_TRX', 'CREDIT_LIMIT', 'PAYMENTS',
        'MINIMUM_PAYMENTS', 'PRC_FULL_PAYMENT']

for col in cols:
    df[col] = np.log(1 + df[col])
```

Kami menggunakan **koefisien 1** dalam log untuk menghindari log(0), dikarenakan nilai minimum pada dataset adalah 0.



DIMENSIONALITY REDUCTION

PCA (Principal Component Analysis)



Dapat dilihat bahwa cukup banyak feature yang berkorelasi, sehingga salah satu feature tersebut kurang lebih merepresentasikan informasi yang sama pada dataset. Oleh karena itu, dapat kita lakukan dimentionality reduction, dan kami akan menggunakan PCA.

```
from sklearn.decomposition import PCA

pca = PCA(n_components=0.95)
reduced_df = pca.fit_transform(df)

pd.DataFrame(reduced_df)
```

DIMENSIONALITY REDUCTION

PCA (Principal Component Analysis)

Secara singkat, PCA akan membuat axis baru untuk menjelaskan maximum variance pada dataset, dan menjadi principal component. Setelah itu, algoritma PCA akan memilih komponen lain yang tegak lurus dengan principal component dimensi pertama, untuk menjelaskan maximum variancenya.

Setelah kita mendapatkan principal components, maka kita dapat memilih berapa banyak komponen yang ingin kita ambil dan representasikan dalam data. Karena hanya mengambil principal component tersebut, maka dimensi data akan berkurang.

Pada kasus ini, kita akan mengambil n_components sebesar 95%, sehingga hasil data PCA akan merepresentasikan 95% varians dari data original sebelum direduksi menjadi satu fitur.

	0
0	-907.918
1	-1003.322
...	...
8947	-1003.323
8948	89.933

BAGIAN 3: PEMBUATAN MODEL

K-Means Merupakan salah satu algoritma clustering.
K-Means adalah algoritma unsupervised learning yang digunakan untuk clustering.
Ini mempartisi dataset menjadi sejumlah cluster (k) tertentu berdasarkan kesamaan titik data dalam sebuah cluster.

TAHAP 1

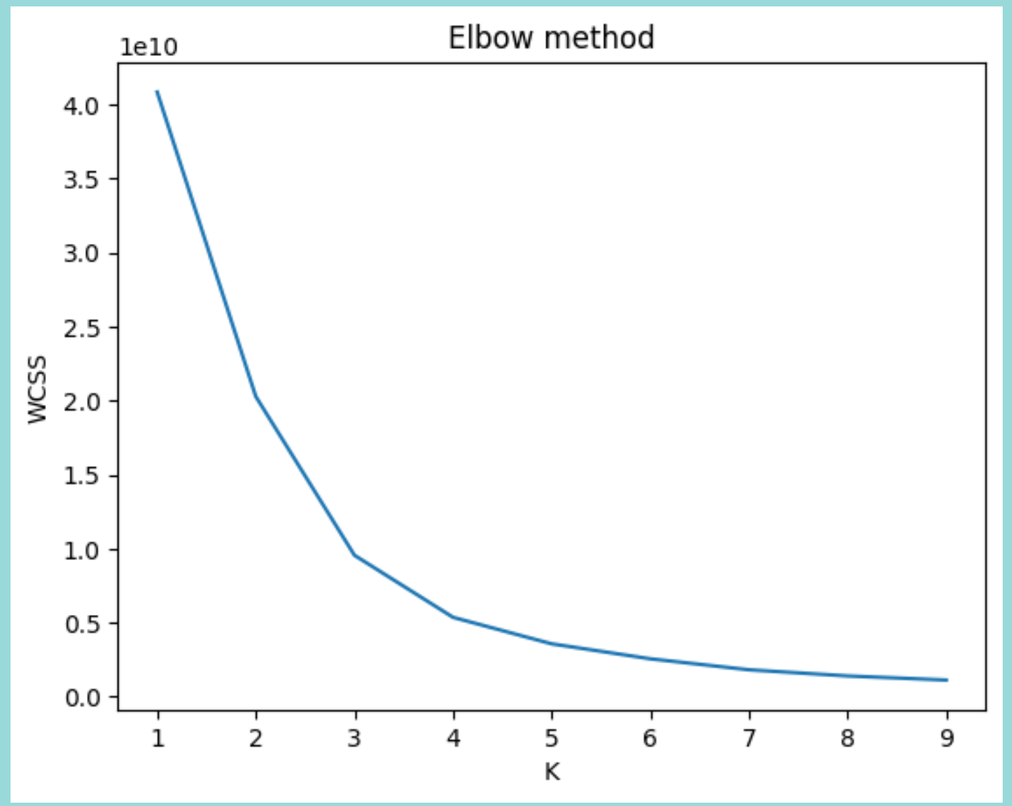
Menentukan Nilai K menggunakan elbow method

Elbow method akan memberi informasi nilai K mana yang terbaik dengan menemukan WCSS, yaitu jumlah jarak kuadrat antara titik-titik dalam sebuah cluster dan centroid clusternya. Semakin tinggi nilai WCSS, maka cluster tersebut belum cukup baik, dimana ada dataset yang sangat jauh dengan nilai centroid clusternya. K terbaik akan dipilih, dimana nilai WCSS akan mulai turun dan membuat sebuah elbow.

```
from sklearn.cluster import KMeans

kmeans_models = [KMeans(n_clusters=k, random_state=23).fit(reduced_df) for k in range (1, 10)]
innertia = [model.inertia_ for model in kmeans_models]
```

TAHAP 1

Menentukan Nilai K menggunakan elbow method



Elbow point yang dihasilkan tidak dapat ditentukan dengan jelas, antara K = 3 atau K = 4. Oleh karena itu, kita dapat menggunakan Silhouette score.

TAHAP 2

Menentukan Nilai K menggunakan Silhouette Score

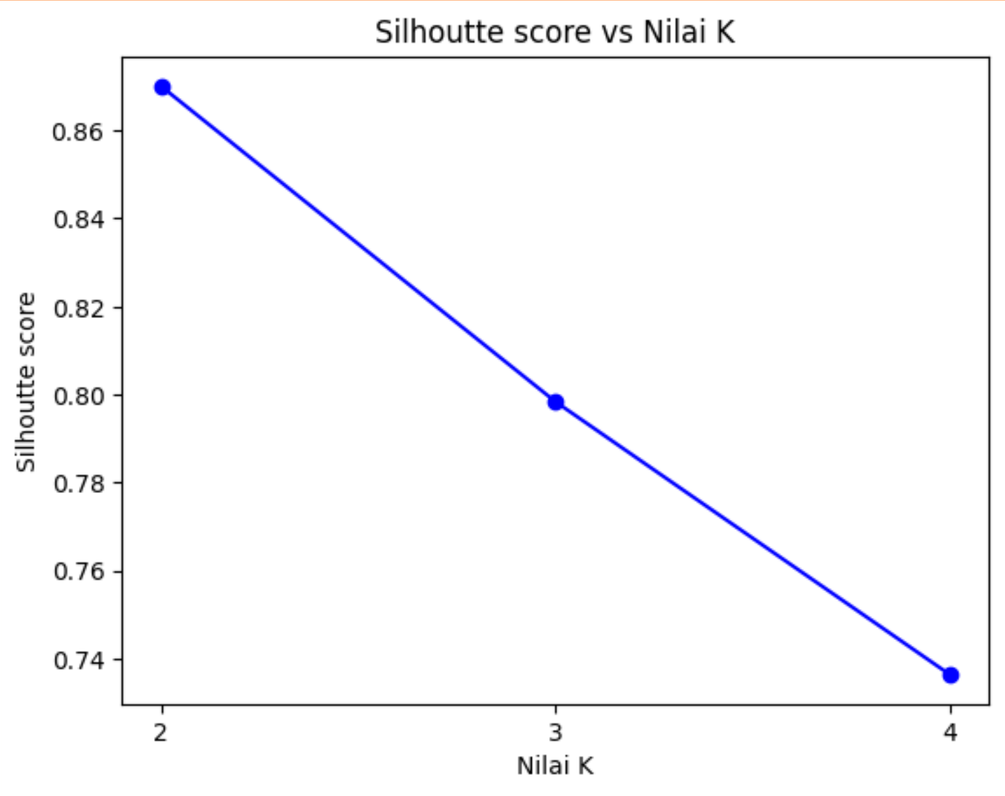
Rumus dari silhouette score adalah (b-a)/max(a,b); dimana a adalah rata-rata jarak diantara tiap poin dalam satu kluster, dan b adalah rata-rata jarak diantara semua cluster.

Silhouette score akan jatuh di nilai -1 sampai dengan 1, yang berarti:
* 1: Poin sudah berada pada cluster yang sesuai dan tiap cluster dapat dibedakan
* 0: Cluster overlapping
* -1: Poin berada pada cluster yang salah

```
from sklearn.metrics import silhouette_score

silhoutte_scores = [silhouette_score(reduced_df, model.labels_) for model in kmeans_models[1:4]]
plt.plot(range(2,5), silhoutte_scores, "bo-")
plt.xticks([2, 3, 4])
```

TAHAP 2

Menentukan Nilai K menggunakan Silhouette Score



nilai silhouette score paling besar jatuh pada nilai K=2, dimana ketepatan poin berada pada cluster yang benar dan lebih tinggi dari yang lain. Hal ini menandakan bahwa hipotesis awal elbow point jatuh di titik 3 dan 4 salah, dan nilai K = 2 lebih baik untuk dataset ini.

TAHAP 3

Evaluasi Model

```
print('Silhoutte score dari model: ' + str(silhouette_score(reduced_df, kmeans.labels_)))

Silhoutte score dari model: 0.870045599956l4
```

Nilai hasil silhouette coefficient terletak pada kisaran nilai -1 hingga 1. Semakin nilai silhouette coefficient mendekati nilai 1, maka semakin baik pengelompokan data dalam suatu cluster.

Dengan menggunakan nilai k = 2, terlihat bahwa proses clustering dapat bekerja dengan efektif dilihat dari silhoute score sebesar 0.87.

BAGIAN 4: Result

Visualisasi persebaran klustering dengan memilih dua feature untuk diwakilkan dalam sumbu x dan y pada plot.

fitur ONEOFF_PURCHASES dengan PURCHASES untuk memvisualisasikan hasil cluster dari dataset, dengan menggunakan scatter plot.

K = 2

Distribution of clusters based on One off purchases and total purchases

cluster 0 (biru): pengeluaran sedikit
cluster 1 (oranye): pengeluaran banyak.

untuk cluster 0, perusahaan dapat menargetkan iklan promosi, testimoni produk untuk meningkatkan kepercayaan brand, sehingga customer akan melakukan transaksi lebih frequent

untuk cluster 1, perusahaan dapat menawarkan produk terbaru, limited-edition product maupun additional service untuk meningkatkan daya transaksi customer

K = 3

Distribution of clusters based on One off purchases and total purchases

cluster 0 : pengeluaran sedang
cluster 1 : pengeluaran sedikit.
cluster 2 : pengeluaran banyak.

Alasan nilai k = 2 lebih efektif:

- Jumlah titik data pada cluster 2 sedikit, dan akan memakan biaya yang lebih dari perusahaan jika ingin membedakan keputusan terhadap cluster tersebut.
- Cluster 2 sangat tersebar dan tidak seperti membentuk cluster dikarenakan silhouette score juga lebih rendah.
- Tidak terdapat pemisah signifikan/yang terlihat dengan jelas pada cluster 0 dan 2.


New Tab
TERIMA
KASIH!