Rakamin Academy

Homework Unsupervised Learning

Team 2 phpMyFeeling:

- Arya Octavian
- Athalla Dewanto
- Bagoes Fikri
- Intan Denovita
- Jose Christian
- Jovian Aditya
- Muhammad Hazim
- Nur Almar
- Syafwan Giffari

EDA + PREPROCESSING

Rakamin Academy

Data Frame Information :

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62988 entries, 0 to 62987
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   MEMBER_NO          62988 non-null  int64
 1   FFP_DATE           62988 non-null  object
 2   FIRST_FLIGHT_DATE  62988 non-null  object
 3   GENDER             62985 non-null  object
 4   FFP_TIER           62988 non-null  int64
 5   WORK_CITY          60719 non-null  object
 6   WORK_PROVINCE      59740 non-null  object
 7   WORK_COUNTRY       62962 non-null  object
 8   AGE                62568 non-null  float64
 9   LOAD_TIME          62988 non-null  object
 10  FLIGHT_COUNT       62988 non-null  int64
 11  BP_SUM             62988 non-null  int64
 12  SUM_YR_1           62437 non-null  float64
 13  SUM_YR_2           62850 non-null  float64
 14  SEG_KM_SUM         62988 non-null  int64
 15  LAST_FLIGHT_DATE   62988 non-null  object
 16  LAST_TO_END        62988 non-null  int64
 17  AVG_INTERVAL       62988 non-null  float64
 18  MAX_INTERVAL       62988 non-null  int64
 19  EXCHANGE_COUNT     62988 non-null  int64
 20  avg_discount       62988 non-null  float64
 21  Points_Sum         62988 non-null  int64
 22  Point_NotFlight    62988 non-null  int64
dtypes: float64(5), int64(10), object(8)
memory usage: 11.1+ MB
```

Handling Missing Values (Categorical Column):

Dapat dilihat bahwa missing values terdapat di kolom yang bertipe data numerik maupun kategorikal. Untuk kolom-kolom data kategorikal diputuskan membuang kolom tersebut karena tidak akan digunakan ketika membuat clustering.

CHECK & HANDLING MISSING VALUES

```
df.isna().sum()
```

```
MEMBER_NO             0
FFP_DATE              0
FIRST_FLIGHT_DATE     0
GENDER                3
FFP_TIER              0
WORK_CITY          2269
WORK_PROVINCE      3248
WORK_COUNTRY         26
AGE                 420
LOAD_TIME             0
FLIGHT_COUNT          0
BP_SUM                0
SUM_YR_1            551
SUM_YR_2            138
SEG_KM_SUM            0
LAST_FLIGHT_DATE      0
LAST_TO_END           0
AVG_INTERVAL          0
MAX_INTERVAL          0
EXCHANGE_COUNT        0
avg_discount          0
Points_Sum            0
Point_NotFlight       0
dtype: int64
```

Untuk kolom-kolom dengan tipe data kategorikal seperti `WORK_CITY`, `WORK_PROVINCE`, `WORK_COUNTRY`, dan `GENDER` diputuskan akan membuang kolom tersebut karena tidak akan digunakan dalam algoritma clustering.

```
df = df.drop(['WORK_CITY','WORK_PROVINCE','WORK_COUNTRY','GENDER'],axis=1)
```

EDA + PREPROCESSING

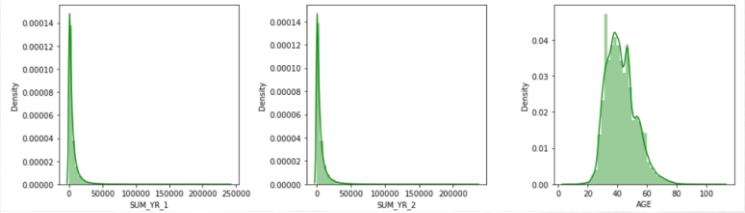
Rakamin Academy

Handling Missing Values (Numerical Column):

Sebelum memutuskan untuk mengganti missing value dengan mean/median, cek terlebih dahulu bentuk distribusinya:

Untuk kolom `SUM_YR_1`, `SUM_YR_2`, `AGE` akan dicek terlebih dahulu bentuk distribusinya sebelum memutuskan mengganti nilai NULL dengan mean/median

```
cek_null = ['SUM_YR_1','SUM_YR_2', 'AGE']
plt.figure(figsize=(18, 4))
for i in range(len(cek_null)):
    plt.subplot(1,4,i+1)
    sns.distplot(df[cek_null[i]],color='g')
    plt.tight_layout()
```




```
#Bentuk Distribusi untuk kolom SUM_YR_1 & SUM_YR_2 --> Right Skewness
#Bentuk Distribusi kolom AGE --> Multimodal
#Diputuskan untuk mengganti semua nilai NULL pada ketiga kolom tersebut dengan Median
df['AGE'] = df['AGE'].fillna(df['AGE'].median())
df['SUM_YR_1'] = df['SUM_YR_1'].fillna(df['SUM_YR_1'].median())
df['SUM_YR_2'] = df['SUM_YR_2'].fillna(df['SUM_YR_2'].median())
```

```
df.isna().any()
```

```
MEMBER_NO           False
FFP_DATE            False
FIRST_FLIGHT_DATE   False
FFP_TIER            False
AGE                 False
LOAD_TIME           False
FLIGHT_COUNT        False
BP_SUM              False
SUM_YR_1            False
SUM_YR_2            False
SEG_KM_SUM          False
LAST_FLIGHT_DATE    False
LAST_TO_END         False
AVG_INTERVAL        False
MAX_INTERVAL        False
EXCHANGE_COUNT      False
avg_discount        False
Points_Sum          False
Point_NotFlight     False
dtype: bool
```

Bentuk Distribusi ada yang right skewness, ada yg multimodal. Akhirnya diputuskan mengganti semuanya dengan median

Check Duplicated Values:

```
df.duplicated().any()
```

```
False
```

Descriptive Statistics:

```
cats = ['FFP_DATE', 'FIRST_FLIGHT_DATE', 'LOAD_TIME', 'LAST_FLIGHT_DATE']
nums = ['MEMBER_NO', 'FFP_TIER', 'AGE', 'FLIGHT_COUNT', 'BP_SUM', 'SUM_YR_1', 'SUM_YR_2',
        'SEG_KM_SUM', 'LAST_TO_END', 'AVG_INTERVAL', 'MAX_INTERVAL',
        'EXCHANGE_COUNT', 'avg_discount', 'Points_Sum', 'Point_NotFlight']
```

```
df[nums].describe()
```

	MEMBER_NO	FFP_TIER	AGE	FLIGHT_COUNT	BP_SUM	SUM_YR_1	SUM_YR_2	SEG_KM_SUM	LAST_TO_END	AVG_INTERVAL
count	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000
mean	31494.500000	4.102162	42.466502	11.839414	10925.081254	5333.022406	5597.823538	17123.878691	176.120102	67.74978
std	18183.213715	0.373856	9.853632	14.049471	16339.486151	8077.407958	8694.832417	20960.844623	183.822223	77.51786
min	1.000000	4.000000	6.000000	2.000000	0.000000	0.000000	0.000000	368.000000	1.000000	0.00000
25%	15747.750000	4.000000	35.000000	3.000000	2518.000000	1020.000000	785.000000	4747.000000	29.000000	23.37037
50%	31494.500000	4.000000	41.000000	7.000000	5700.000000	2800.000000	2773.000000	9994.000000	108.000000	44.66666
75%	47241.250000	4.000000	48.000000	15.000000	12831.000000	6524.250000	6826.250000	21271.250000	268.000000	82.00000
max	62988.000000	6.000000	110.000000	213.000000	505308.000000	239560.000000	234188.000000	580717.000000	731.000000	728.00000

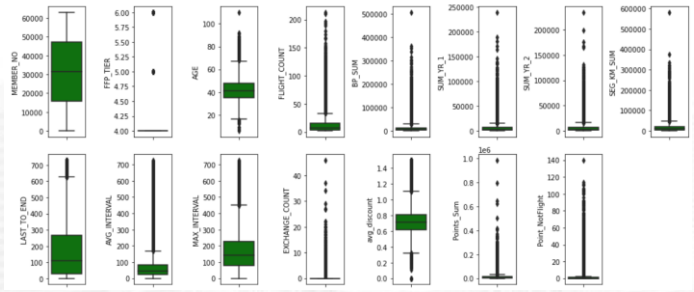
```
df[cats].describe()
```

	FFP_DATE	FIRST_FLIGHT_DATE	LOAD_TIME	LAST_FLIGHT_DATE
count	62988	62988	62988	62988
unique	3068	3406	1	731
top	1/13/2011	2/16/2013	3/31/2014	3/31/2014
freq	184	96	62988	959

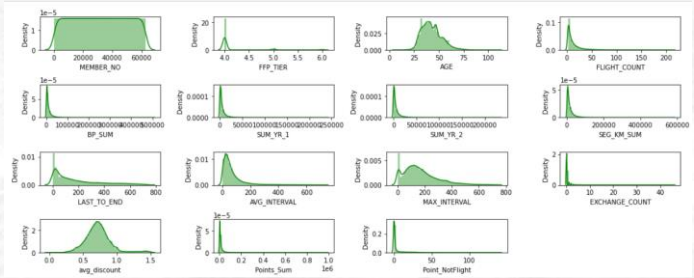
EDA + PREPROCESSING

Univariate Analysis (1/3):

```
plt.figure(figsize = (14,6))
for i in range(len(nums)):
    plt.subplot(2,8,i+1)
    sns.boxplot(y=df[nums[i]],color='g',orient='v')
    plt.tight_layout()
```



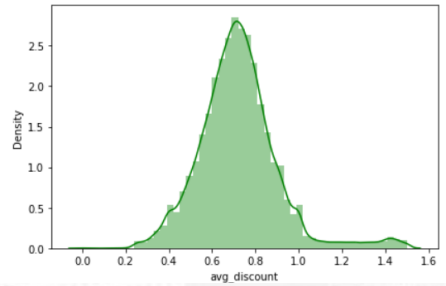

```
plt.figure(figsize = (15,6))
for i in range(len(nums)):
    plt.subplot(4,4,i+1)
    sns.distplot(df[nums[i]],color='g')
    plt.tight_layout()
```



Untuk sementara step handling outlier akan diskip terlebih dahulu., akan dilakukan pada saat telah menentukan fitur apa saja yang akan digunakan agar lebih efisien.

Univariate Analysis (2/3):

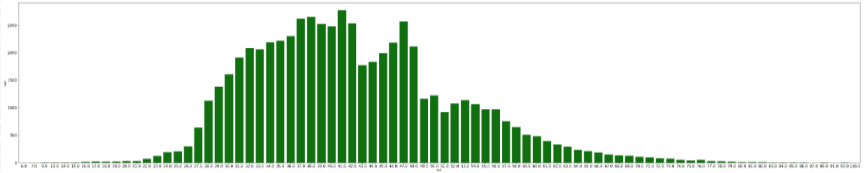
```
sns.distplot(df['avg_discount'],color='g')
plt.tight_layout()
plt.show()
```



Terlihat sedikit janggal karena ada yang mendapat avg_discount > 1. Dimana asumsi avg_discount 1 = 100%

```
#buang data yang diskon > 100%
df = df[df['avg_discount']<=1]
```

```
plt.figure(figsize = (50,10))
sns.countplot(x=df['AGE'],color='g')
plt.tight_layout()
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.show()
```



Dapat dilihat bahwa distribusi umur terbesar pada rentang 27-56 tahun. Pada umur tersebut merupakan usia produktif sehingga banyak perjalanan yang dilakukan oleh kelompok usia tersebut.

EDA + PREPROCESSING

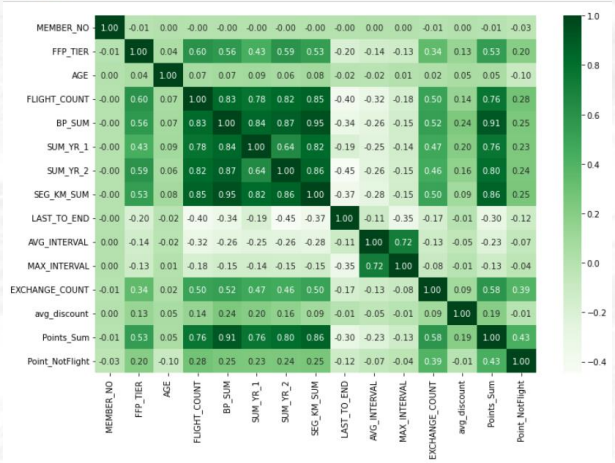
Univariate Analysis (3/3):

```
plt.figure(figsize=(50,10))
sns.countplot(x=df["FLIGHT_COUNT"], color='g')
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.show()
```

Dapat dilihat bahwa setiap pelanggan setidaknya pernah melakukan penerbangan sebanyak 2x.

Multivariate Analysis:

```
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), cmap='Greens', annot=True, fmt='.2f')
plt.show()
```



Setelah melihat dari matriks korelasi dan melihat definisi dari setiap feature diputuskan akan membuang feature berikut ini karena dirasa tidak akan berguna dalam membentuk model clustering : `age`, `MEMBER_NO`, `SUM_YR_1`, `SUM_YR_2`, `Point_NotFlight`, `AVG_INTERVAL`, `MAX_INTERVAL`, `EXCHANGE_COUNT`. Sepertinya `BP_SUM` & `Points_Sum` & `SEG_KM_SUM` merupakan feature yang redundan maka diputuskan hanya akan mengambil salah satunya. Sebenarnya `FLIGHT_COUNT` juga tergolong redundan terhadap `BP_SUM` tetapi feature tersebut merupakan feature yang penting sehingga tidak dibuang.

FEATURE SELECTION & EXTRACTION

Rakamin Academy

Feature Selection :

Feature yang dipilih sementara:

- FLIGHT_COUNT
- SEG_KM_SUM
- LAST_FLIGHT_DATE
- LAST_TO_END
- FIRST_FLIGHT_DATE
- avg_discount
- LOAD_TIME
- FFP_DATE

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60041 entries, 0 to 62987
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   FLIGHT_COUNT       60041 non-null  int64
 1   SEG_KM_SUM         60041 non-null  int64
 2   FIRST_FLIGHT_DATE  60041 non-null  object
 3   LAST_FLIGHT_DATE   60041 non-null  object
 4   LAST_TO_END        60041 non-null  int64
 5   avg_discount       60041 non-null  float64
 6   LOAD_TIME          60041 non-null  object
 7   FFP_DATE           60041 non-null  object
dtypes: float64(1), int64(3), object(4)
memory usage: 4.1+ MB
```

Feature Extraction :

1. Buat Flight / Year Column:

```
# rata-rata penerbangan dalam 1 tahun
df['FLIGHT_PER_YEAR'] =  df['FLIGHT_COUNT'] / ((df['LAST_FLIGHT_DATE'] - df['FIRST_FLIGHT_DATE']) / np.timedelta64(1, 'Y'))
```

2. Buat Join Month Column:

(sudah berapa lama jadi member frequent flyer agar lebih mudah interpretasinya bikin dalam bulan saja. Karena kalau hasilnya misal 2.35 tahun akan jadi sulit diinterpretasi)

```
# Sudah berapa lama join program frequent flyer
df['JOIN_MONTH'] = (df['LOAD_TIME'] - df['FFP_DATE']) / np.timedelta64(1,'M')
```

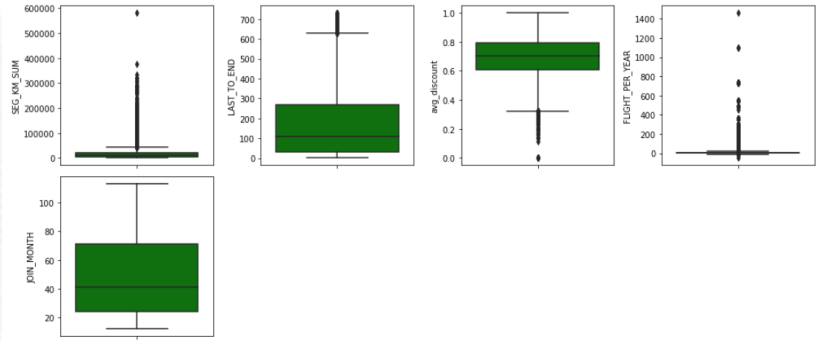
Final Feature for Clustering :

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60041 entries, 0 to 62987
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   SEG_KM_SUM       60041 non-null  int64
 1   LAST_TO_END      60041 non-null  int64
 2   avg_discount     60041 non-null  float64
 3   FLIGHT_PER_YEAR  60041 non-null  float64
 4   JOIN_MONTH       60041 non-null  float64
dtypes: float64(3), int64(2)
memory usage: 2.7 MB
```

HANDLING OUTLIER :

Cek Distribusi Boxplot :

```
# Handling Outlier (dikarenakan saat awal belum dilakukan)
plt.figure(figsize = (14,6))
feats = ['SEG_KM_SUM', 'LAST_TO_END', 'avg_discount', 'FLIGHT_PER_YEAR', 'JOIN_MONTH']
for i in range(len(feats)):
    plt.subplot(2,4,i+1)
    sns.boxplot(y=df[feats[i]],color='g',orient='v')
    plt.tight_layout()
```



Outlier hanya terdapat pada kolom `SEG_KM_SUM`, `LAST_TO_END`, avg_discount & `FLIGHT_PER_YEAR`

Handling Outlier IQR Method:

```
#SEG_KM_SUM (1)
Q1_1 = df['SEG_KM_SUM'].quantile(0.25)
Q3_1 = df['SEG_KM_SUM'].quantile(0.75)
IQR = Q3_1 - Q1_1
batas_bawah_1 = Q1_1 - (IQR * 1.5)
batas_atas_1 = Q3_1 + (IQR * 1.5)

#LAST_TO_END (2)
Q1_2= df['LAST_TO_END'].quantile(0.25)
Q3_2= df['LAST_TO_END'].quantile(0.75)
IQR = Q3_2 - Q1_2
batas_bawah_2 = Q1_2 - (IQR * 1.5)
batas_atas_2 = Q3_2 + (IQR * 1.5)

#avg_discount (3)
Q1_3 = df['avg_discount'].quantile(0.25)
Q3_3 = df['avg_discount'].quantile(0.75)
IQR = Q3_3 - Q1_3
batas_bawah_3 = Q1_3 - (IQR * 1.5)
batas_atas_3 = Q3_3 + (IQR * 1.5)

#AVG_PER_YEAR (4)
Q1_4 = df['FLIGHT_PER_YEAR'].quantile(0.25)
Q3_4 = df['FLIGHT_PER_YEAR'].quantile(0.75)
IQR = Q3_4 - Q1_4
batas_bawah_4 = Q1_4 - (IQR * 1.5)
batas_atas_4 = Q3_4 + (IQR * 1.5)


df_cluster = df[((df['SEG_KM_SUM'] >= batas_bawah_1) & (df['SEG_KM_SUM'] <= batas_atas_1)) &
                ((df['LAST_TO_END'] >= batas_bawah_2) & (df['LAST_TO_END'] <= batas_atas_2)) &
                ((df['avg_discount'] >= batas_bawah_3) & (df['avg_discount'] <= batas_atas_3)) &
                ((df['FLIGHT_PER_YEAR'] >= batas_bawah_4) & (df['FLIGHT_PER_YEAR'] <= batas_atas_4))]
```

STANDARDIZATION:

Rakamin Academy

Final Data Frame for Clustering:

```
df_cluster
```

	SEG_KM_SUM	LAST_TO_END	avg_discount	FLIGHT_PER_YEAR	JOIN_MONTH
0	42885	92	0.998753	14.030381	31.967802
1	44965	38	0.947906	6.265920	62.095731
2	43488	122	0.979314	3.312080	111.049508
3	43823	353	0.964519	2.024305	67.713916
4	44132	23	0.955218	9.838182	49.512310
...	...	...	...	...	...
48877	368	471	0.750000	0.664077	51.582168
48878	368	492	0.750000	1.466837	32.460625
48879	368	252	0.750000	0.232639	111.443767
48880	368	418	0.750000	0.478693	63.804185
48881	368	89	0.710000	0.239346	106.745518

48882 rows × 5 columns

Standardization:

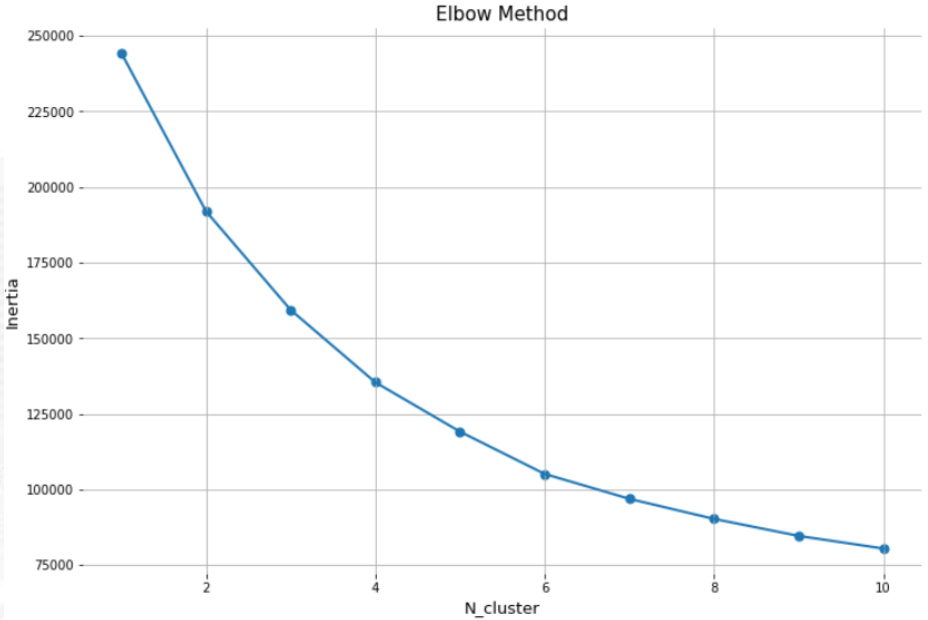
```
#Standardization for all features
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
df_cluster_std = sc.fit_transform(df_cluster)
```

```
df_cluster_std
```

```
array([[ 2.94682306, -0.43341155,  2.18610534,  2.36472813, -0.64082997],
       [ 3.15183584, -0.77659283,  1.81649865,  0.44135765,  0.46366097],
       [ 3.00625705, -0.24275528,  2.04480275, -0.29035166,  2.2583082 ],
       ...,
       [-1.24381562,  0.58342188,  0.37794266, -1.05317437,  2.27276173],
       [-1.24381562,  1.63838657,  0.37794266, -0.99222317,  0.52629296],
       [-1.24381562, -0.45247717,  0.08718682, -1.05151275,  2.10052378]])
```

EXTERNAL EVALUATION:

ELBOW METHOD :



	Inertia Reduction (%)
0	21.436055
1	16.995433
2	15.030147
3	12.051583
4	11.740894
5	7.838495
6	6.854494
7	6.227758
8	4.938600
9	NaN

Dapat dilihat bahwa merubah k = 3 ke k = 4 masih memberikan inertia reduction 15% sedangkan ketika merubah k = 4 menjadi k = 5 inertia reduction berkurang menjadi 12%. Oleh sebab itu digunakan **k = 4**

CLUSTERING & VISUALIZATION:

Rakamin Academy

CLUSTERING USING K-MEANS:

```
kmeans = KMeans(n_clusters=4,random_state=42).fit(df_cluster_std)
kmeans.fit(df_cluster_std)
```

```
KMeans(n_clusters=4, random_state=42)
```

```
df_cluster_std = pd.DataFrame(data=df_cluster_std, columns=['SEG_KM_SUM', 'LAST_TO_END',
                                                            'avg_discount', 'FLIGHT_PER_YEAR', 'JOIN_MONTH'])
df_cluster_std['k_label'] = kmeans.labels_
df_cluster_std
```

	SEG_KM_SUM	LAST_TO_END	avg_discount	FLIGHT_PER_YEAR	JOIN_MONTH	k_label
0	2.946823	-0.433412	2.186105	2.364728	-0.640830	1
1	3.151836	-0.776593	1.816499	0.441358	0.463661	1
2	3.006257	-0.242755	2.044803	-0.290352	2.258308	2
3	3.039276	1.225298	1.937261	-0.609352	0.669624	2
4	3.069732	-0.871921	1.869654	1.326259	0.002352	1
...	...	...	...	...	...	...
48877	-1.243816	1.975213	0.377943	-0.946301	0.078233	0
48878	-1.243816	2.108672	0.377943	-0.747445	-0.622763	0
48879	-1.243816	0.583422	0.377943	-1.053174	2.272762	2
48880	-1.243816	1.638387	0.377943	-0.992223	0.526293	0
48881	-1.243816	-0.452477	0.087187	-1.051513	2.100524	2

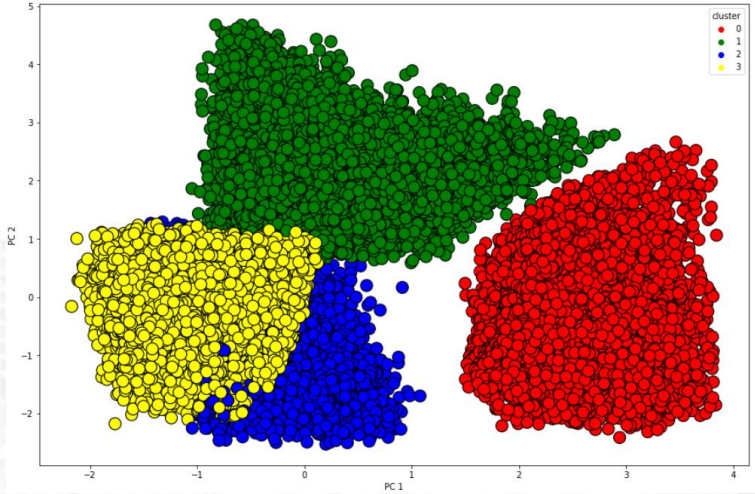
48882 rows × 6 columns

VISUALIZATION WITH PCA:

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(df_cluster_std)
df_pca = pca.transform(df_cluster_std)
```

```
df_pca = pd.DataFrame(data = df_pca, columns = ['PC 1', 'PC 2'])
df_pca['cluster'] = kmeans.labels_
df_pca.head()
```

	PC 1	PC 2	cluster
0	0.055138	3.595867	1
1	-0.292964	1.747790	1
2	-0.540565	0.016185	2
3	0.335833	0.406766	2
4	-0.320742	2.600534	1



DATA FRAME CLUSTERING RESULT & SUMMARY

Rakamin Academy

DATAFRAME CLUSTERING + RESULT

	SEG_KM_SUM	LAST_TO_END	avg_discount	FLIGHT_PER_YEAR	JOIN_MONTH	k_label
0	42885	92	0.998753	14.030381	31.967802	1
1	44965	38	0.947906	6.265920	62.095731	1
2	43488	122	0.979314	3.312080	111.049508	2
3	43823	353	0.964519	2.024305	67.713916	2
4	44132	23	0.955218	9.838182	49.512310	1
...	...	...	...	...	...	...
48877	368	471	0.750000	0.664077	51.582168	0
48878	368	492	0.750000	1.466837	32.460625	0
48879	368	252	0.750000	0.232639	111.443767	2
48880	368	418	0.750000	0.478693	63.804185	0
48881	368	89	0.710000	0.239346	106.745518	2

48882 rows × 6 columns

SUMMARY DATAFRAME FOR INTERPRETATION :

```
df_cluster_interpretation = df_cluster.groupby('k_label').agg({'SEG_KM_SUM' : 'mean',
                                                               'LAST_TO_END':'mean',
                                                               'avg_discount':'mean',
                                                               'FLIGHT_PER_YEAR': 'mean',
                                                               'JOIN_MONTH' : ['mean','count']})
```

```
df_cluster_interpretation
```

	SEG_KM_SUM	LAST_TO_END	avg_discount	FLIGHT_PER_YEAR	JOIN_MONTH	
	mean	mean	mean	mean	mean	count
k_label						
0	5893.962479	424.865803	0.732073	2.899471	52.099532	8875
1	22115.375624	103.976374	0.710630	10.600567	29.370775	9608
2	16713.563620	86.593455	0.710196	2.274904	80.757342	14178
3	8204.876703	113.025091	0.661233	3.659461	32.523818	16221

CLUSTERING RESULTS INTERPRETATION :

	SEG_KM_SUM	LAST_TO_END	avg_discount	FLIGHT_PER_YEAR	JOIN_MONTH	
	mean	mean	mean	mean	mean	count
k_label						
0	5893.962479	424.865803	0.732073	2.899471	52.099532	8875
1	22115.375624	103.976374	0.710630	10.600567	29.370775	9608
2	16713.563620	86.593455	0.710196	2.274904	80.757342	14178
3	8204.876703	113.025091	0.661233	3.659461	32.523818	16221

Cluster 0
Tipe pelanggan ini adalah pelanggan yang telah melakukan total jarak penerbangan dengan rata-rata 5900 km (paling rendah dibandingkan kelompok pelanggan pada cluster lain), dan sudah cukup lama tidak melakukan penerbangan dilihat dari kolom `LAST_TO_END` yang sangat besar dibandingkan dengan cluster lainnya, hal ini juga dapat terlihat bahwa pelanggan ini dalam 1 tahun rata-rata hanya melakukan penerbangan sebanyak hampir 3x. Pelanggan ini juga terlihat sudah lumayan lama menjadi member frequent flyer dengan durasi rata-rata lama menjadi member selama 52 bulan.

Cluster 1
Tipe pelanggan ini adalah pelanggan yang telah melakukan total jarak penerbangan dengan rata-rata 22000km (terbesar dibandingkan kelompok pelanggan pada cluster lain), dan terlihat pelanggan ini memiliki jarak waktu penerbangan terbaru terhadap penerbangan terakhir cukup pendek dimana rata2 kolom `LAST_TO_END` yang cukup pendek dibandingkan kelompok Cluster 0. Hal ini juga dapat dibuktikan bahwa kelompok pelanggan ini dalam 1 tahun rata-rata melakukan penerbangan sebanyak hampir 11x (terbesar dibandingkan cluster lainnya). Kelompok pelanggan ini jika dilihat merupakan member baru dari frequent flyer dimana kelompok pelanggan ini baru bergabung menjadi member selama 29 bulan.

CLUSTERING RESULTS INTERPRETATION :

	SEG_KM_SUM	LAST_TO_END	avg_discount	FLIGHT_PER_YEAR	JOIN_MONTH	
	mean	mean	mean	mean	mean	count
k_label						
0	5893.962479	424.865803	0.732073	2.899471	52.099532	8875
1	22115.375624	103.976374	0.710630	10.600567	29.370775	9608
2	16713.563620	86.593455	0.710196	2.274904	80.757342	14178
3	8204.876703	113.025091	0.661233	3.659461	32.523818	16221

Cluster 2
Tipe pelanggan ini adalah pelanggan yang telah melakukan total jarak penerbangan dengan rata-rata 16000 km (tergolong besar walaupun masih dibawah Cluster 1), pelanggan ini meskipun memiliki jarak waktu penerbangan terbaru terhadap penerbangan terakhir paling pendek dibandingkan cluster lainnya namun kelompok pelanggan ini dalam 1 tahun rata-rata hanya melakukan penerbangan sebanyak 2x. Kelompok pelanggan ini terlihat sudah lama menjadi member frequent flyer dimana terlihat dari durasi rata-rata lama menjadi member sudah selama 81 bulan (terlama dibandingkan cluster lainnya).

Cluster 3 Tipe pelanggan ini adalah pelanggan yang telah melakukan total jarak penerbangan dengan rata-rata 8200 km (meskipun bukan yang terendah, tetapi tergolong rendah dibandingkan cluster 1 & cluster 2). Tipe pelanggan ini memiliki jarak waktu penerbangan terbaru terhadap penerbangan terakhir relatif pendek namun kelompok pelanggan ini dalam 1 tahun rata-rata hanya melakukan penerbangan sebanyak hampir 4x. Kelompok pelanggan ini juga tergolong masih baru menjadi member dari frequent flyer (sama halnya seperti cluster 1) dimana kelompok pelanggan ini telah menjadi member selama 32 bulan.

BUSINESS RECOMMENDATION :

1. Terlihat bahwa masih ada kelompok pelanggan yang memiliki jumlah penerbangan per tahun cukup rendah. Untuk meningkatkan jumlah penerbangan per tahun untuk para kelompok tersebut selain dengan memberikan promo cara yang cukup penting adalah meningkatkan kualitas pelayanan yang dimiliki. Karena dalam industri penerbangan seperti ini kualitas pelayanan merupakan salah satu aspek yang penting.

2. Terlihat bahwa dari ke 4 cluster tersebut ada yg tergolong sebagai member frequent flyer baru dan member frequent flyer lama. Kepada member lama maupun member baru dapat diberikan promo-promo yang sesuai. Misal kepada member yang sudah lama berlanggan diberikan diskon/promo yang lebih besar sebagai bentuk apresiasi, namun kepada member baru juga tetap diberikan diskon/promo meskipun tidak sebesar member lama dengan harapan bahwa member baru akan tetap aktif dan meningkatkan jumlah penerbangan.