

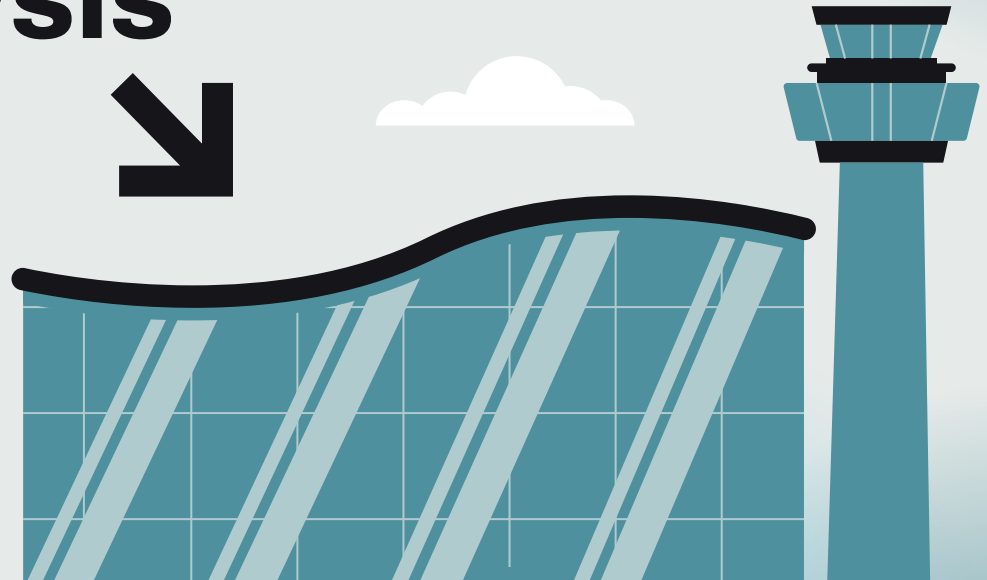
# Homework Unsupervised Learning

# Airline Customer

# Value Analysis

## Group 7 (Car-a-thon):

Bagus Ganjar Lugina  
Bernadetha Stella  
Samuella Magdalena E  
Raihan Kurniasugianto  
M Rifqi Sarosa



# Table of Contents

**01**



**EDA**

Descriptive statistics, univariate and multivariate analysis

**02**



**Data Pre-Processing**

Data cleansing, handling data and feature engineering

**03**



**Modelling**

Clustering and evaluation

**04**



**Recommendation**

Clustering description and business recommendation



# *01*

## **Exploratory Data Analysis**

# Descriptive Statistic

```
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
```

```
df.isna().sum().sort_values(ascending=False) / len(df)*100
```

WORK_PROVINCE	5.156538
WORK_CITY	3.602273
SUM_YR_1	0.874770
AGE	0.666794
SUM_YR_2	0.219089
WORK_COUNTRY	0.041278
GENDER	0.004763
MEMBER_NO	0.000000
LAST_FLIGHT_DATE	0.000000
Points_Sum	0.000000
avg_discount	0.000000
EXCHANGE_COUNT	0.000000
MAX_INTERVAL	0.000000
AVG_INTERVAL	0.000000
LAST_TO_END	0.000000
BP_SUM	0.000000
SEG_KM_SUM	0.000000
FFP_DATE	0.000000
FLIGHT_COUNT	0.000000
LOAD_TIME	0.000000
FFP_TIER	0.000000
FIRST_FLIGHT_DATE	0.000000
Point_NotFlight	0.000000

dtype: float64

```
print(df[df.duplicated()].shape)
df.duplicated().any()
```

```
(0, 23)
```

Berdasarkan data tersebut didapatkan info sebagai berikut :

- Data terdiri dari **62988** kolom
- Nama dan tipe data tiap kolom sudah sesuai.
- Masih terdapat missing value pada beberapa kolom yaitu **WORK\_PROVINCE**, **WORK\_CITY**, **SUM\_YR\_1**, **AGE**, **SUM\_YR\_2**, **WORK\_COUNTRY**, dan **GENDER**.
- Berdasarkan missing value tersebut, terlihat jumlah missing value < 5% sehingga akan didrop.
- Dalam data tersebut tidak terdapat duplicate value.

# Descriptive Statistic

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	MAX_INTERVAL	EXCHANGE_COUNT	avg_discount	Points_Sum	Point_NotFlight
count	62988.000000	62988.000000	62568.000000	62988.000000	62988.000000	62437.000000	62850.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.0000	62988.000000
mean	31494.500000	4.102162	42.476346	11.839414	10925.081254	5355.376064	5604.026014	17123.878691	176.120102	67.749788	166.033895	0.319775	0.721558	12545.7771	2.728155
std	18183.213715	0.373856	9.885915	14.049471	16339.486151	8109.450147	8703.364247	20960.844623	183.822223	77.517866	123.397180	1.136004	0.185427	20507.8167	7.364164
min	1.000000	4.000000	6.000000	2.000000	0.000000	0.000000	0.000000	368.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000
25%	15747.750000	4.000000	35.000000	3.000000	2518.000000	1003.000000	780.000000	4747.000000	29.000000	23.370370	79.000000	0.000000	0.611997	2775.0000	0.000000
50%	31494.500000	4.000000	41.000000	7.000000	5700.000000	2800.000000	2773.000000	9994.000000	108.000000	44.666667	143.000000	0.000000	0.711856	6328.5000	0.000000
75%	47241.250000	4.000000	48.000000	15.000000	12831.000000	6574.000000	6845.750000	21271.250000	268.000000	82.000000	228.000000	0.000000	0.809476	14302.5000	1.000000
max	62988.000000	6.000000	110.000000	213.000000	505308.000000	239560.000000	234188.000000	580717.000000	731.000000	728.000000	728.000000	46.000000	1.500000	985572.0000	140.000000

df[cats].describe()								
	FFP_DATE	FIRST_FLIGHT_DATE	GENDER	WORK_CITY	WORK_PROVINCE	WORK_COUNTRY	LOAD_TIME	LAST_FLIGHT_DATE
count	62988	62988	62985	60719	59740	62962	62988	62988
unique	3068	3406	2	3234	1165	118	1	731
top	1/13/2011	2/16/2013	Male	guangzhou	guangdong	CN	3/31/2014	3/31/2014
freq	184	96	48134	9386	17509	57748	62988	959

Berdasarkan data tersebut didapatkan info dari feature categorical sebagai berikut :

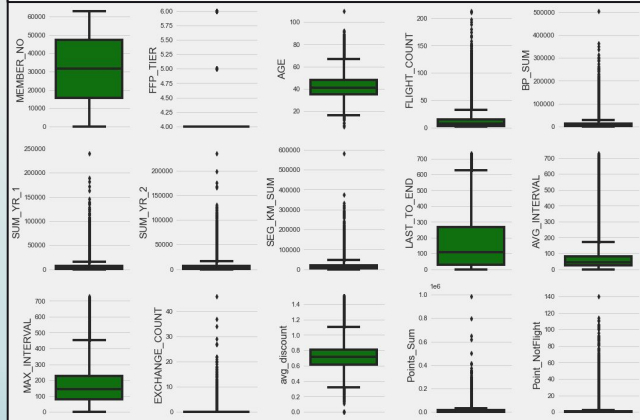
- FFP\_DATE, FIRST\_FLIGHT\_DATE, WORK\_CITY, WORK\_PROVINCE, WORK\_COUNTRY, dan LAST\_FLIGHT\_DATE memiliki unique value yang besar (>100).
- Gender Male mendominasi flight customer yaitu sebesar 48134.

Berdasarkan data tersebut didapatkan info dari feature **numeric** sebagai berikut :

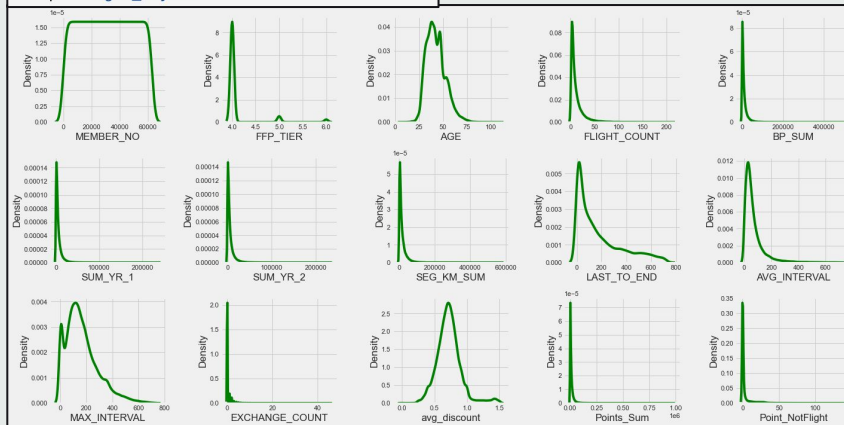
- Tidak ada issue pada nilai minimal. Semua data bernilai positif.
- Kolom FFP\_TIER dan avg\_discount distribusinya tampak simetrik.
- Banyak kolom yang cukup skew karena terdapat perbedaan nilai mean dan median yang cukup besar.

# Univariate Analysis

```
plt.figure(figsize=(15,10))
for i in range(0, len(nums)):
    plt.subplot(3, 5, i+1)
    sns.boxplot(y=df[nums[i]], color='green', orient='v')
    plt.tight_layout()
```



```
plt.figure(figsize=(20,10))
for i in range(0, len(nums)):
    plt.subplot(3, 5, i+1)
    sns.kdeplot(x=df[nums[i]], color='green')
    plt.xlabel(nums[i])
    plt.tight_layout()
```

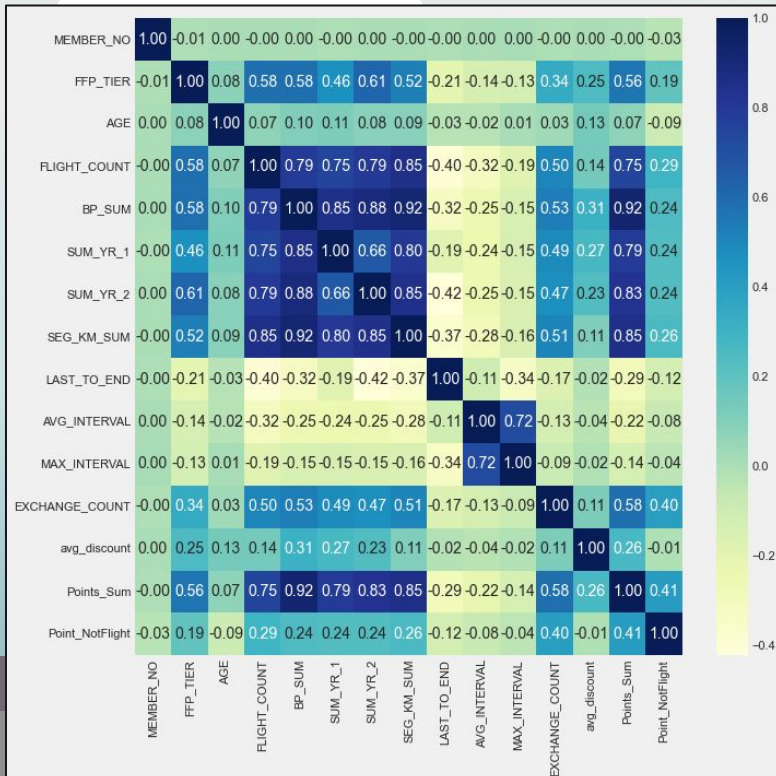


Berdasarkan box plot dan lineplot tersebut didapatkan info sebagai berikut :

- Outlier terlihat paling besar ada pada FLIGHT\_COUNT, BP\_SUM, SUM\_YR1, SUM\_YR\_2, SEG\_KM\_SUM, EXCHANGE\_COUNT, POINT\_SUM, dan POINT\_NoFlight.
- Hampir semua fitur skew positif.
- Avg\_discount memiliki distribusi yang paling simetrik.

# Multivariate Analysis

```
df.corr()  
plt.figure(figsize=(10, 10))  
sns.heatmap(df.corr(), cmap='YlGnBu', annot=True, fmt='.2f');
```

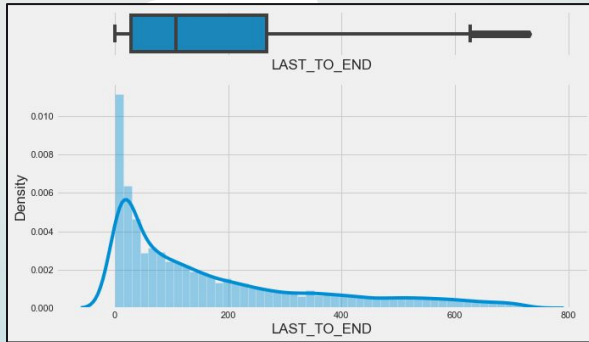


Pengamatan korelasi antar fitur yg memiliki high correlative:

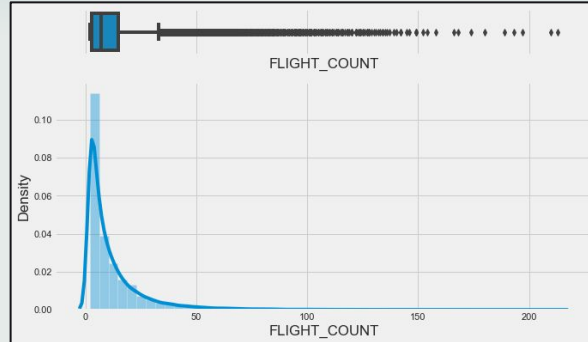
1. FLIGHT\_COUNT - Point\_Sum (0.75)
2. FLIGHT\_COUNT - SEG\_KM\_SUM (0.85)
3. FLIGHT\_COUNT - SUM\_YR\_1 (0.75)
4. FLIGHT\_COUNT - SUM\_YR\_2 (0.79)
5. FLIGHT\_COUNT - BP\_SUM (0.79)
6. BP\_SUM - SEG\_KM\_SUM (0.90)
7. BP\_SUM - SUM\_YR\_1 (0.85)
8. BP\_SUM - SUM\_YR\_2 (0.88)
9. Point\_Sum - BP\_SUM (0.92)
10. Point\_Sum - SUM\_YR\_1 (0.79)
11. Point\_Sum - SUM\_YR\_2 (0.83)
12. Point\_Sum - SEG\_KM\_SUM (0.85)
13. SUM\_YR\_1 - SEG\_KM\_SUM (0.80)
14. SUM\_YR\_1 - SUM\_YR\_2 (0.66)
15. SUM\_YR\_2 - SEG\_KM\_SUM (0.85)
16. SUM\_YR\_2 - FFP\_TIER (0.61)
17. AVG\_INTERVAL - MAX\_INTERVAL (0.72)

# RFM

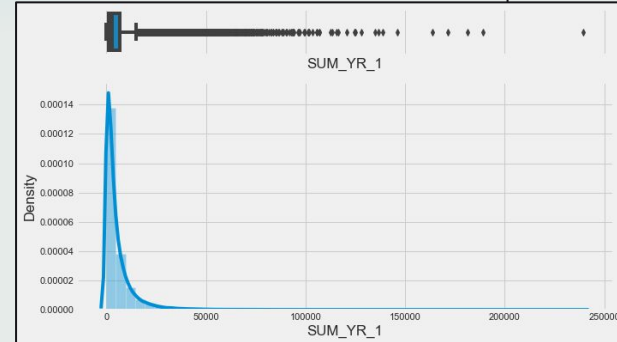
## Recency



## Frequency



## Monetary



RFM digunakan untuk menentukan segmentasi.

1. Untuk recency, diambil fitur LAST\_TO\_END, untuk melihat transaksi terakhir customer melakukan penerbangan. Dari plot diatas, dapat dilihat datanya berdistribusi skew kanan.
2. Untuk frequency, diambil fitur FLIGHT\_COUNT, untuk melihat jumlah penerbangan. Terlihat distribusi data flight\_count juga skew kanan dan memiliki outlier yang cukup banyak. Rata-rata jumlah penerbangan adalah 7.
3. Untuk monetary, diambil fitur SUM\_YR\_1, untuk melihat jumlah biaya yang dikeluarkan customer dalam melakukan perjalanan. Cukup banyak outlier pada data tersebut.





# 02

## **Data Pre-Processing**

# Data Pre-Processing

Fitur yang digunakan dibagi berdasarkan segmentasi:

- **Recency** LAST\_TO\_END: Kebaruan, kapan terakhir kali customer membeli tiket penerbangan.
- **Frequency** FLIGHT\_COUNT: melihat frekuensi berapa kali customer melakukan penerbangan.
- **Monetary** SUM\_YR\_1: nilai monetary, berapa banyak uang yang sudah dihabiskan customer dalam perjalanan penerbangannya.

```
[38]: df = df.copy()
```

```
[39]: df_rfm = df[['LAST_TO_END', 'FLIGHT_COUNT', 'SUM_YR_1']]
df_rfm.columns = ['R', 'F', 'M']
df_rfm_drop = df_rfm[['R', 'F', 'M']]
```

```
[40]: df_rfm_drop.head()
```

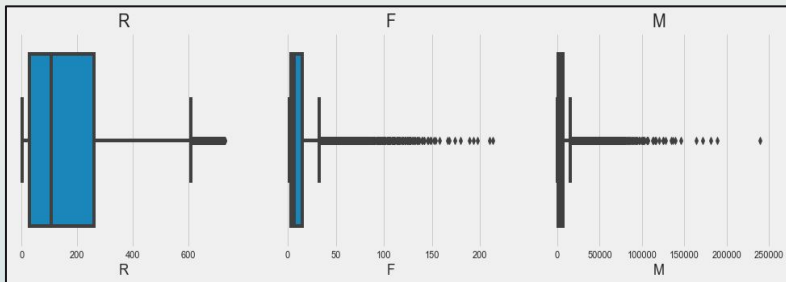
```
[40]:
```

	R	F	M
0	1	210	239560.0
1	7	140	171483.0
2	11	135	163618.0
3	97	23	116350.0
4	5	152	124560.0

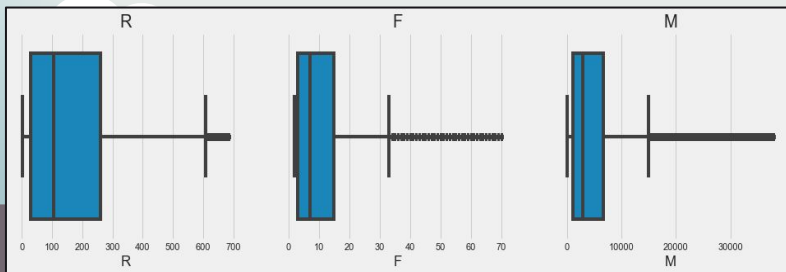
# Data Pre-Processing

## Handling Outliers

Before Handled:



After Handled:



## Scaling

```
[45]: from sklearn.preprocessing import MinMaxScaler
```

```
[46]: df_std = rfm_log
scale = MinMaxScaler()
kolom_all = [x for x in df_std.columns]
for kolom in kolom_all:
    df_std[kolom] = scale.fit_transform(np.array(df_std[kolom]).reshape(-1,1))
```

```
[47]: df_std.head()
```


```
[47]:
```

	R	F	M
0	0.000000	1.000000	1.0
1	0.008772	1.000000	1.0
2	0.014620	1.000000	1.0
3	0.140351	0.308824	1.0
4	0.005848	1.000000	1.0

```
[48]: df_std.describe()
```

```
[48]:
```


	R	F	M
count	62437.000000	62437.000000	62437.000000
mean	0.251019	0.142880	0.136426
std	0.263967	0.189467	0.179398
min	0.000000	0.000000	0.000000
25%	0.040936	0.014706	0.026428
50%	0.153509	0.073529	0.073778
75%	0.380117	0.191176	0.173220
max	1.000000	1.000000	1.000000



## Feature Engineering:

Feature Selection: Semua fitur dilakukan **drop** kecuali kolom **LAST\_TO\_END**, **FLIGHT\_COUNT**, **SUM\_YR\_1**.

Feature Extraction: Setelah melakukan drop kolom, fitur yang baru diubah penamaannya menjadi per segmentasi (*Recency, Frequency, Monetary*):

- **LAST\_TO\_END** -> R
  - **FLIGHT\_COUNT** -> F
  - **SUM\_YR\_1** -> M
- 

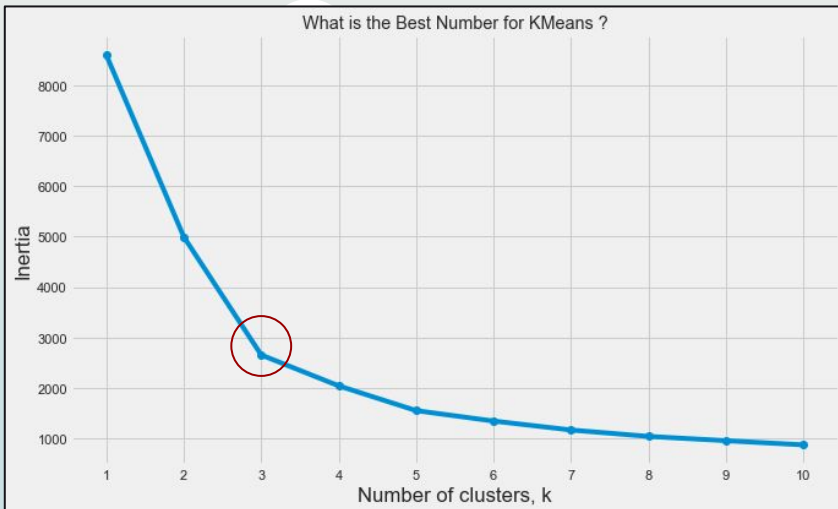


*03*

**Modelling**

# Clustering Model

## Elbow Method



## Silhouette Score

For n\_clusters = 2. The average silhouette\_score is : 0.32698526676597534  
For n\_clusters = 3. The average silhouette\_score is : 0.41966905511633124  
For n\_clusters = 4. The average silhouette\_score is : 0.3570707177653772  
For n\_clusters = 5. The average silhouette\_score is : 0.29315516266694125



Jumlah cluster optimal: **3 cluster**

## Hasil K-Means Clustering

	R	F	M	count
K_Cluster				
0	0.662442	0.031420	0.087912	14796
1	0.135849	0.107956	0.084946	40074
2	0.056474	0.545775	0.503915	7567



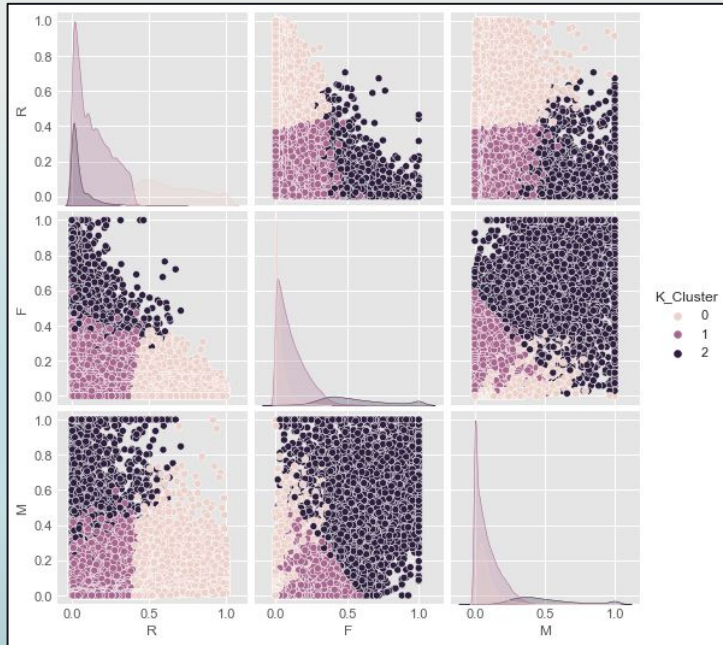
Hasil clustering menunjukkan bahwa cluster terbesar adalah K-Cluster =1



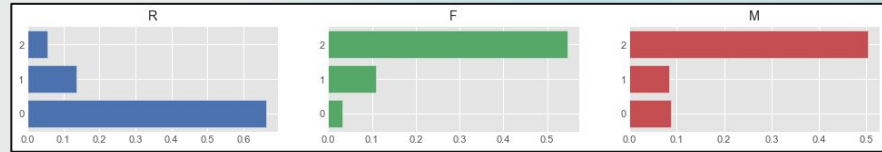
# Evaluasi Clustering



## Visualisasi Cluster



## Pemetaan Cluster



➡ Berdasarkan visualisasi pemetaan tersebut, customer dapat dikelompokkan sebagai berikut:

**K-Cluster = 0 → Low Value**

**K-Cluster = 1 → General**

**K-Cluster = 2 → Loyal**



➡ Dimensi cluster sudah terpetakan dengan jelas sehingga metode PCA sudah tidak diperlukan



# 04

**Recommendation**



# Cluster Labelling

```
cluster = [] #membuat list kosong
for i, k in data_labeling.iterrows(): #iterasi setiap row
    if k['K_Cluster'] == 0:
        cluster_name = 'Low Value'
    elif k['K_Cluster'] == 1:
        cluster_name = 'General'
    else:
        cluster_name = 'Loyal'
    cluster.append(cluster_name)

data_labeling['cluster'] = cluster #membuat kolom dari list
data_label['cluster'] = cluster #membuat kolom dari list
```

	R	F	M	K_Cluster	count
cluster					
General	93.920921	9.341019	3223.860109	1.0	40074
Low Value	454.936199	4.136591	3336.404926	0.0	14796
Loyal	39.627990	40.798467	20591.409806	2.0	7567

- **K-Cluster = 0** sebagai cluster `Low Value` karena customer pada cluster ini tidak banyak melakukan perjalanan, mengeluarkan sedikit uang untuk perjalanan, serta sudah lama tidak melakukan perjalanan dengan perusahaan penerbangan tersebut,
- **K-Cluster = 1** sebagai cluster `General` karena customer pada cluster ini memiliki tingkat perjalanan, tingkat pengeluaran uang untuk perjalanan, serta tingkat kebaruan perjalanan yang cenderung menengah/standar,
- **K-Cluster = 2** sebagai cluster `Loyal` karena customer pada cluster ini memiliki tingkat perjalanan dan tingkat pengeluaran uang yang cenderung tinggi, serta melakukan perjalanan terakhir yang terhitung baru.

# Business Recommendation



## Frequent Flyer



Membuat program frequent flyer yang ditawarkan kepada customer di **cluster Loyal** untuk menjaga kesetiaan customer dalam menggunakan layanan airline tersebut.



## Voucher



Memberikan voucher/diskon kepada customer di **cluster General** untuk meningkatkan ketertarikan customer dalam menggunakan layanan airline tersebut.



## Campaign



Membuat campaign marketing pada customer di **cluster Low Value** agar mendapatkan lebih banyak customer di berbagai target pasar.

# Thanks

**Group 7 - DS Batch 20**  
**Rakamin Academy**

