

cluster

July 13, 2025

```
[5]: import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, LabelEncoder
import seaborn as sns
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
from sklearn.metrics import silhouette_score
```

```
[6]: df = pd.read_csv('/var/www/html/Portfolio-Magang-Indosat/data_pelanggan_indosat.
    csv')
df.head()
```

```
[6]:
```

	Nama	No_HP	Kota	Paket	Tgl_beli	\
0	Balidin Dongoran, S.T.	857	Surabaya	Freedom Combo	2024-10-08	
1	Okto Jailani	814	Jakarta	Freedom Combo	2025-03-15	
2	R. Lantar Anggraini	814	Medan	Freedom Combo	2024-07-22	
3	Darimin Pradipta	814	Semarang	Unlimited 2GB	2025-01-11	
4	Kanda Napitupulu	856	Surabaya	Yellow	2025-03-30	

	Durasi_Bulan	Frekuensi_Topup	Kuota_Bulan_GB
0	9	5	16
1	4	1	7
2	12	2	20
3	2	3	5
4	12	1	1

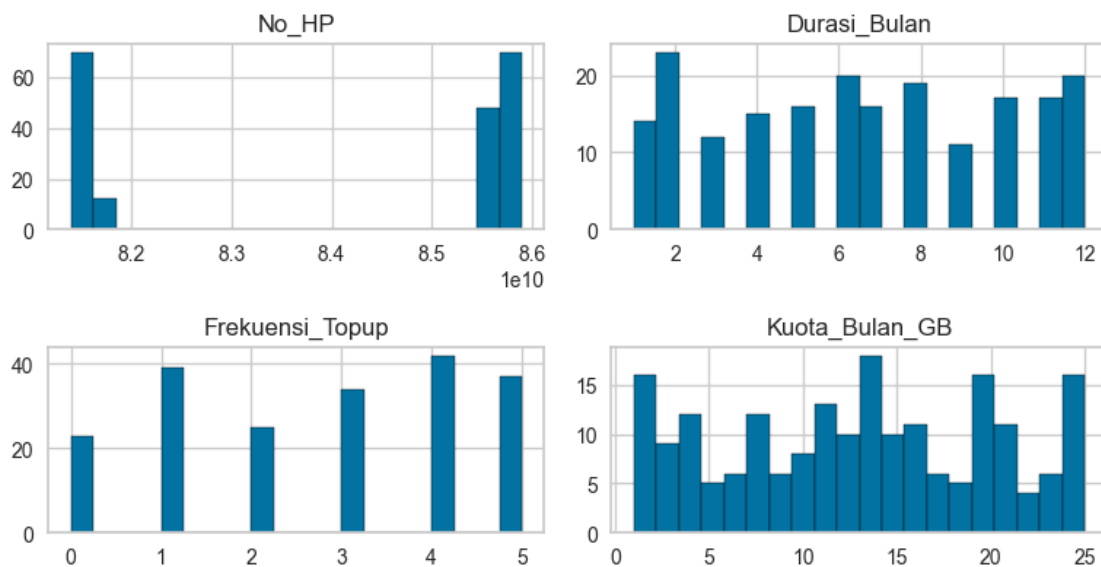
```
[7]: numeric_features = df.select_dtypes(include='number').columns

fig, axes = plt.subplots(3, 2, figsize=(8, 6))
axes = axes.flatten()

for i, column in enumerate(df[numeric_features].columns):
    df[numeric_features][column].hist(ax=axes[i], bins=20, edgecolor='black')
    axes[i].set_title(column)

for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])
```

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



```
[8]: # scaling using standardization, karena data terdistribusi normal
std_scale = StandardScaler()
df[numeric_features] = std_scale.fit_transform(df[numeric_features])

df.head()
```

```
[8]:
```

	Nama	No_HP	Kota	Paket	Tgl_beli	\
0	Balidin Dongoran, S.T.		Surabaya	Freedom Combo	2024-10-08	
1	Okto Jailani		Jakarta	Freedom Combo	2025-03-15	
2	R. Lantar Anggraini		Medan	Freedom Combo	2024-07-22	
3	Darimin Pradipta - ...		Semarang	Unlimited 2GB	2025-01-11	
4	Kanda Napitupulu		Surabaya	Yellow	2025-03-30	

	Durasi_Bulan	Frekuensi_Topup	Kuota_Bulan_GB
0	0.692458	1.359747	0.448111
1	-0.738240	-1.025774	-0.830177
2	1.550877	-0.429394	1.016239
3	-1.310520	0.166986	-1.114241
4	1.550877	-1.025774	-1.682369

```
[9]: # ubah fitur kategori menjadi numerik dengan LabelEncoder
kategori_features = df.select_dtypes(include='object').columns
encoders = {}
df_temp = df.copy()
```

```

for feature in kategori_features:
    le = LabelEncoder()
    df_temp[feature] = le.fit_transform(df[feature])
    encoders[feature] = le

```

```

df = df_temp
df.head()

```

```

[9]:
Nama      No_HP Kota  Paket  Tgl_beli  Durasi_Bulan  Frekuensi_Topup \
0      11          5      0      44      0.692458      1.359747
1     115          1      0     108     -0.738240     -1.025774
2     138          3      0      2      1.550877     -0.429394
3      23          4      2      86     -1.310520      0.166986
4      88          5      4     114      1.550877     -1.025774

```

```

Kuota_Bulan_GB
0      0.448111
1     -0.830177
2      1.016239
3     -1.114241
4     -1.682369

```

[10]: # Melakukan Handling Outlier Data berdasarkan jumlah outlier, apakah menggunakan metode drop atau mengisi nilai tersebut.

```

for feature in df[numeric_features].columns:
    Q1 = df[feature].quantile(0.25)
    Q3 = df[feature].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR

    # tangani outlier dengan
    median = df[feature].median()
    df.loc[:, feature] = df[feature].apply(lambda x: median if x < lower or x > upper else x)

```

visualisasikan

```
cols = 4
```

```
fig, axes = plt.subplots(2, cols, figsize=(10, 8))
```

```

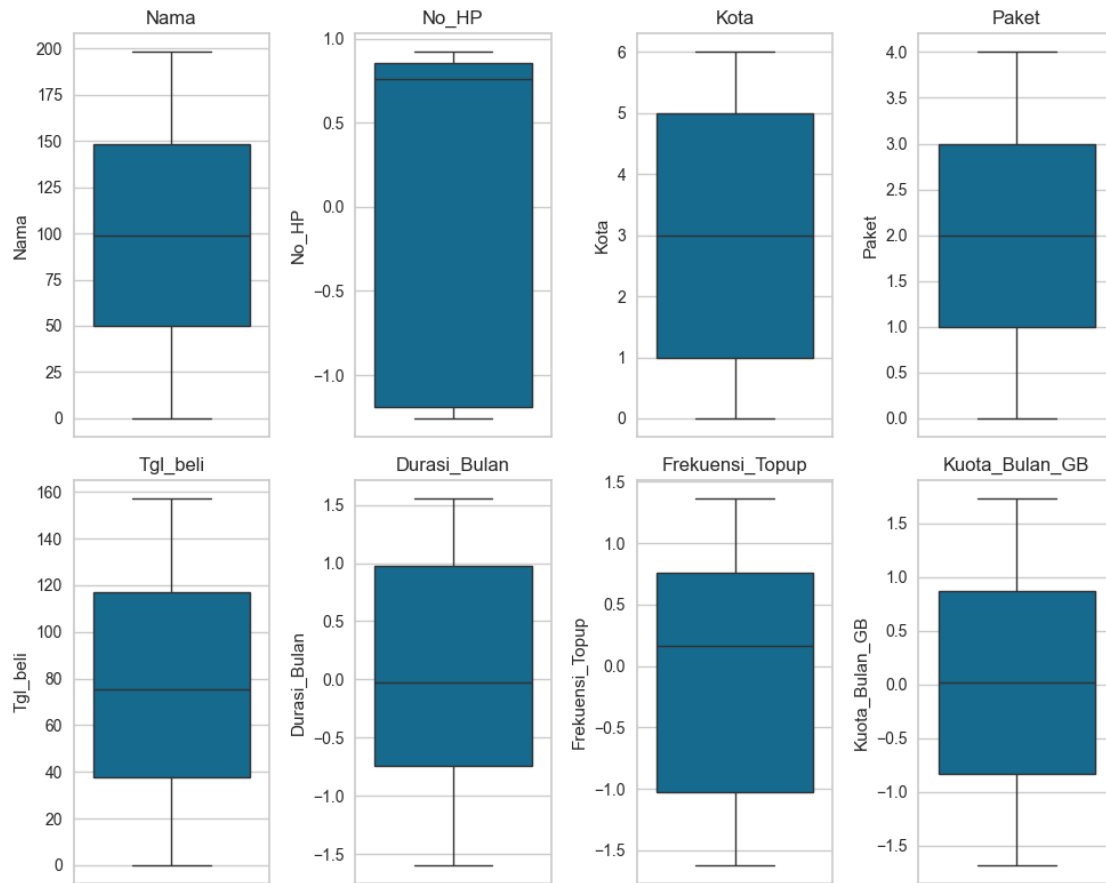
for i, feature in enumerate(df.columns):
    baris, kolom = divmod(i, cols)
    sns.boxplot(y=df[feature], ax=axes[baris, kolom])
    axes[baris, kolom].set_title(f'{feature}')

```

```

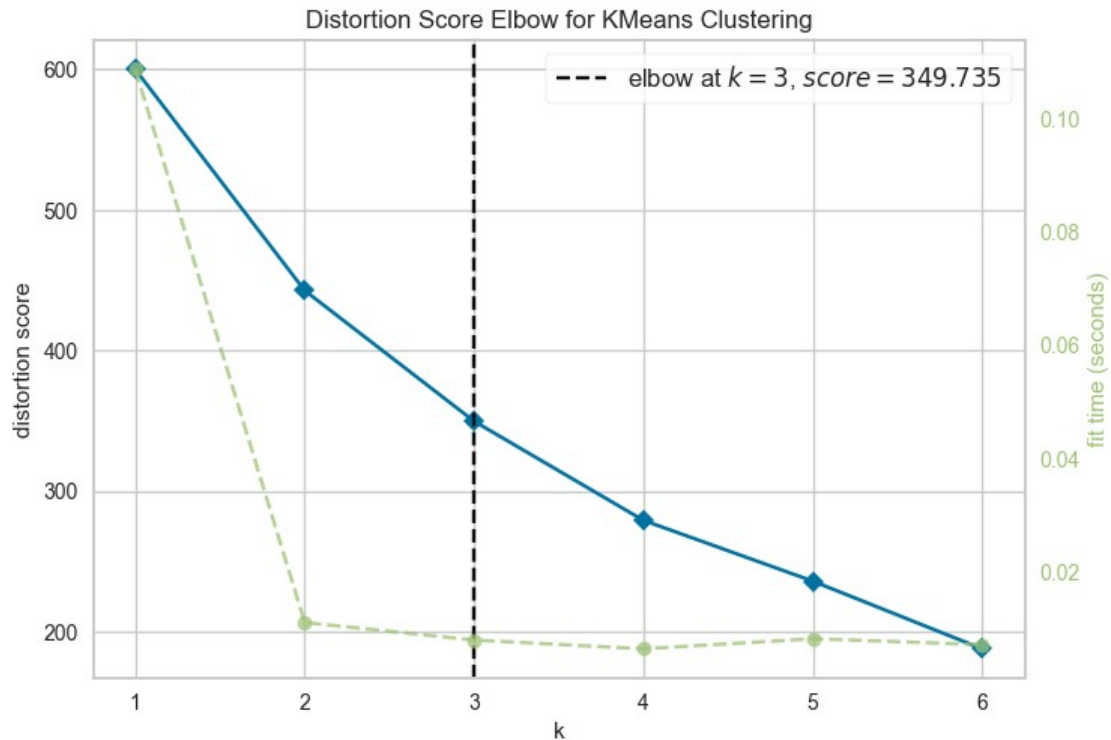
plt.tight_layout()
plt.show()

```



```
[11]: # Melakukan visualisasi Elbow Method menggunakan KElbowVisualizer()
cluster_features = df[['Durasi_Bulan', 'Kuota_Bulan_GB', 'Frekuensi_Topup']]
kmeans_elbow = KMeans(random_state=42)
kelbow_visual = KElbowVisualizer(kmeans_elbow, k=(1, 7))
kelbow_visual.fit(cluster_features)

kelbow_visual.show()
```



[11]: <Axes: title={'center': 'Distortion Score Elbow for KMeans Clustering'},
xlabel='k', ylabel='distortion score'>

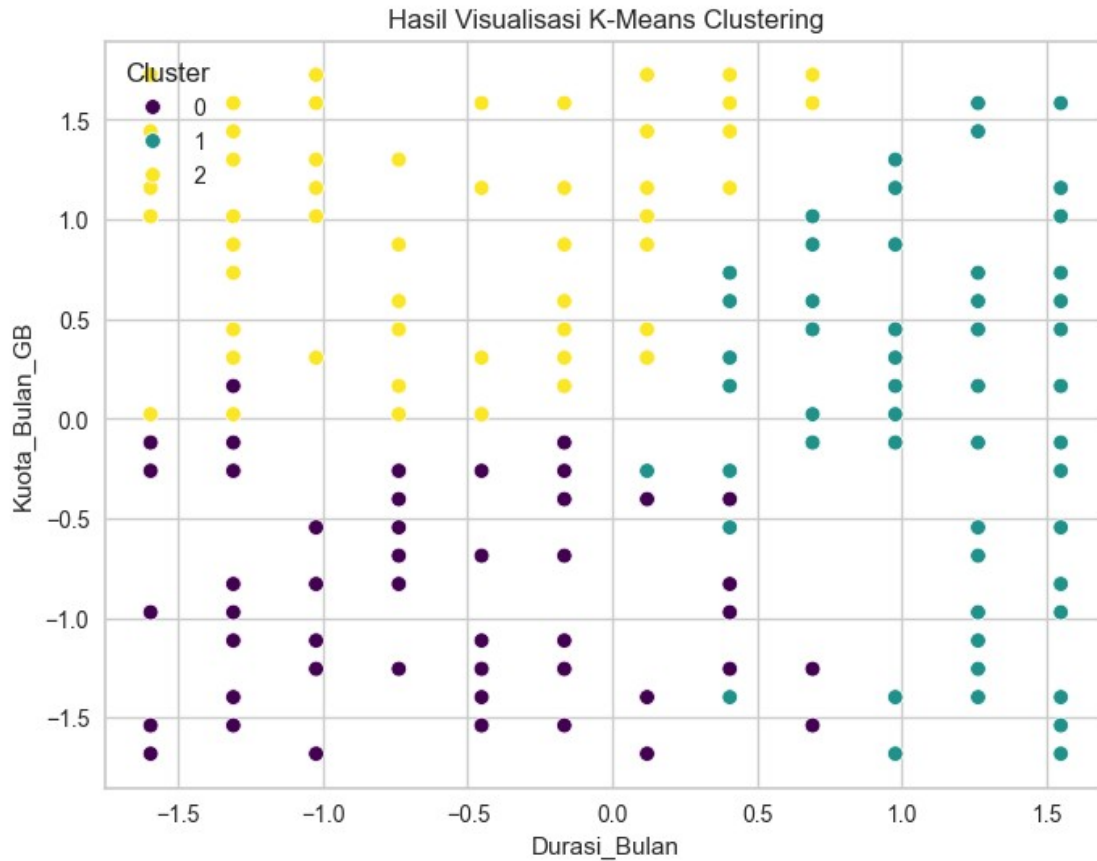
```
[12]: # Menggunakan algoritma K-Means Clustering
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(cluster_features)
```

[12]: KMeans(n_clusters=3, random_state=42)

```
[13]: # Menghitung dan menampilkan nilai Silhouette Score.
silhouette_score(cluster_features, kmeans.fit_predict(cluster_features))
```

[13]: 0.2413538505054932

```
[14]: # Membuat visualisasi hasil clustering
df['Cluster'] = kmeans.labels_
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Durasi_Bulan', y='Kuota_Bulan_GB', hue='Cluster',
               palette='viridis')
plt.title('Hasil Visualisasi K-Means Clustering')
plt.xlabel('Durasi_Bulan')
plt.ylabel('Kuota_Bulan_GB')
plt.show()
```



[15]: # *inverse dataset*

```
df[numeric_features] = std_scale.inverse_transform(df[numeric_features])

for feature in kategori_features:
    df[feature] = encoders[feature].inverse_transform(df[feature].astype(int))

df.head()
```

[15]:

	Nama	No HP	Kota	Paket	Tgl_beli \
0	Balidin Dongoran, S.T.		Surabaya	Freedom Combo	2024-10-08
1	Okto Jailani		Jakarta	Freedom Combo	2025-03-15
2	R. Lantar Anggraini		Medan	Freedom Combo	2024-07-22
3	Darimin Pradipta		Semarang	Unlimited 2GB	2025-01-11
4	Kanda Napitupulu		Surabaya	Yellow	2025-03-30

	Durasi_Bulan	Frekuensi_Topup	Kuota_Bulan_GB	Cluster
0	9.0	5.0	16.0	1
1	4.0	1.0	7.0	0
2	12.0	2.0	20.0	1

3	2.0	3.0	5.0	0
4	12.0	1.0	1.0	1

[16]: # *descriptive features*

```
descriptive_features = ['Durasi_Bulan', 'Frekuensi_Topup', 'Kuota_Bulan_GB']
descriptive_features_categorical = ['Kota', 'Paket']

agg_result = df.groupby('Cluster')[descriptive_features].agg(['mean', 'min', 'max', 'count'])
agg_result_categorical = df.groupby('Cluster')[descriptive_features_categorical].agg(pd.Series.mode)

display(agg_result)
display(agg_result_categorical)
```

Cluster	Durasi_Bulan				Frekuensi_Topup			
	mean	min	max	count	mean	min	max	count
0	4.441176	1.0	10.0	68	2.176471	0.0	5.0	68
1	10.333333	7.0	12.0	72	2.902778	0.0	5.0	72
2	4.500000	1.0	9.0	60	3.116667	0.0	5.0	60

Cluster	Kuota_Bulan_GB			
	mean	min	max	count
0	6.705882	1.0	15.0	68
1	12.902778	1.0	24.0	72
2	19.733333	13.0	25.0	60

Cluster	Kota	Paket
	0	Yogyakarta
1	Surabaya	Freedom Internet
2	[Bandung, Semarang]	Freedom Internet