Clustering K-Means

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Membaca data
pariwisata = pd.read_excel("pariwisata.xlsx")
```

```
#Menampilkan sampel data
pariwisata.tail()
```

	Provinsi	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	hun
30	MALUKU	7.57	130.06	4173.0	416005.9	38.10	
31	MALUKU UTARA	0.66	93.65	516.0	308029.4	50.44	
32	PAPUA BARAT	19.25	243.80	1772.0	237191.0	51.16	
33	PAPUA	12.58	449.56	2557.0	448181.0	50.34	
34	INDONESIA	11307.43	74066.92	3283275.0	57370362.0	54.81	

```
#Menghapus baris ke-34 INDONESIA
pariwisata = pariwisata.drop(34,axis=0)
```

```
pariwisata
```

	Provinsi	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	h
0	ACEH	9.21	312.73	10803.0	1030136.0	43.18	
1	SUMATERA UTARA	201.88	2736.09	21104.0	4850322.0	48.86	
2	SUMATERA BARAT	51.83	1470.99	3786.0	1167212.0	56.00	
3	RIAU	35.33	4646.98	65845.0	1413345.0	48.69	
4	JAMBI	6.02	502.86	2391.0	429609.7	45.49	
5	SUMATERA SELATAN	14.05	2013.76	1157.0	1159986.0	53.99	
6	BENGKULU	0.56	171.82	559.0	446140.5	64.06	
7	LAMPUNG	3.01	785.58	10710.0	1207142.0	60.31	
8	KEP. BANGKA BELITUNG	9.68	467.16	175.0	212880.4	36.07	
9	KEP. RIAU	1595.59	1504.26	97611.0	1056443.0	52.31	
10	DKI JAKARTA	1529.76	10262.67	5969.0	4310569.0	59.71	
11	JAWA BARAT	502.72	12850.51	37134.0	7483742.0	54.47	
12	JAWA TENGAH	122.15	7247.54	7861.0	5539721.0	47.46	
13	DI YOGYAKARTA	211.50	5025.09	57837.0	3711716.0	58.91	
14	JAWA TIMUR	267.97	7526.00	59205.0	9094596.0	57.20	
15	BANTEN	464.21	2821.56	1429.0	943900.3	51.57	
16	BALI	5687.80	3186.16	2462937.0	1858640.0	61.13	
17	NUSA TENGGARA BARAT	195.80	557.37	237746.0	932996.5	42.23	
18	NUSA TENGGARA TIMUR	58.59	449.67	112806.0	456460.6	52.17	
19	KALIMANTAN BARAT	33.00	977.22	7301.0	1207524.0	47.74	
20	KALIMANTAN TENGAH	6.68	396.28	1454.0	997713.2	56.71	
21	KALIMANTAN SELATAN	9.41	1073.82	505.0	942969.8	50.72	
22	KALIMANTAN TIMUR	30.66	1910.00	3151.0	1008884.0	57.70	
23	KALIMANTAN UTARA	2.57	75.14	2424.0	362328.3	46.10	

24	SULAWESI UTARA	164.71	829.37	42208.0	632282.6	64.40
25	SULAWESI TENGAH	2.00	154.09	5952.0	685989.3	50.13
26	SULAWESI	46.91	2681.40	11228.0	1819646.0	51.03

```
pariwisata.info()
```

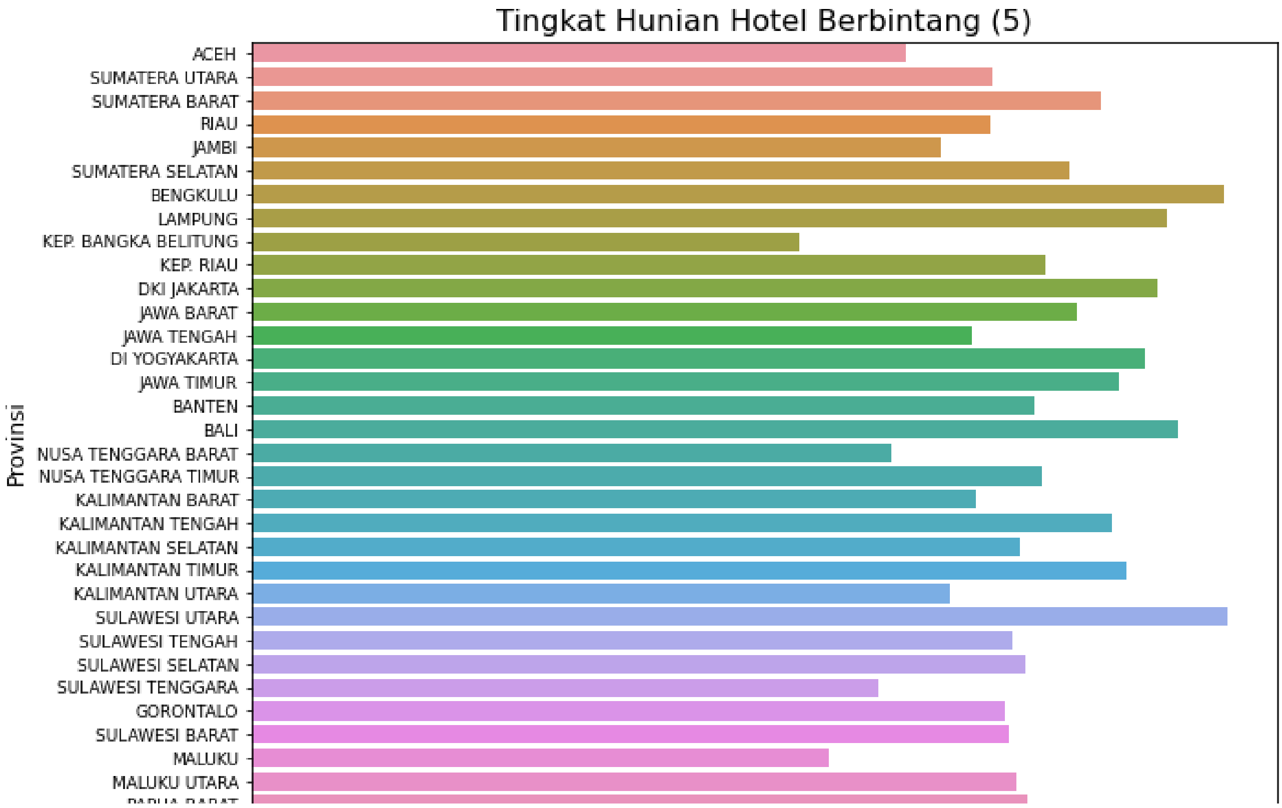
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 34 entries, 0 to 33
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Provinsi      34 non-null     object
 1   jum_asing     34 non-null     float64
 2   jum_lokal     34 non-null     float64
 3   jum_nb_asing  34 non-null     float64
 4   jum_nb_lokal  34 non-null     float64
 5   hunian_b      34 non-null     float64
 6   hunian_nb     34 non-null     float64
dtypes: float64(6), object(1)
memory usage: 2.1+ KB
```

```
pariwisata.isnull().any()
```

```
Provinsi        False
jum_asing       False
jum_lokal       False
jum_nb_asing    False
jum_nb_lokal    False
hunian_b        False
hunian_nb       False
dtype: bool
```

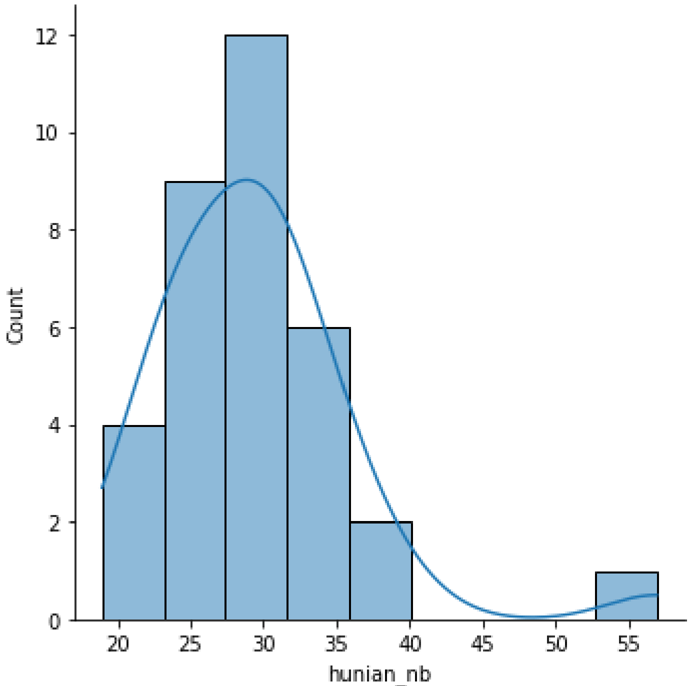
VISUALISASI

```
fig, ax = plt.subplots()
fig.set_size_inches(10,8)

a = sns.barplot(x="hunian_b", y="Provinsi", data = pariwisata)
a.axes.set_title("Tingkat Hunian Hotel Berbintang (5)", fontsize = 16)
a.set_xlabel("Tingkat Hunian ($)", fontsize = 12)
a.set_ylabel("Provinsi",fontsize = 12)
a.tick_params(labelsize = 9)
plt.show()

#simpan foto
fig.savefig("tingkat hunian.png")
```




```
sns.displot(pariwisata.hunian_nb, kde = True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fa6adf29d90>
```



Clustering

```
pariwisata.shape
```

```
(34, 7)
```

```
# Menentukan variabel yang akan di clustering
```

```
x = pariwisata.iloc[:, 1:7]
x.head()
```

	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	hunian_nb
0	9.21	312.73	10803.0	1030136.0	43.18	28.48
1	201.88	2736.09	21104.0	4850322.0	48.86	36.54
2	51.83	1470.99	3786.0	1167212.0	56.00	24.75
3	35.33	4646.98	65845.0	1413345.0	48.69	30.71
4	6.02	502.86	2391.0	429609.7	45.49	23.13

```
#Mengubah data menjadi menjadi array
x_array = np.array(x)
x_array
```

```
[1.405000e+01, 2.013760e+03, 1.157000e+03, 1.159986e+06,
 5.399000e+01, 3.251000e+01],
[5.600000e-01, 1.718200e+02, 5.590000e+02, 4.461405e+05,
 6.406000e+01, 2.904000e+01],
[3.010000e+00, 7.855800e+02, 1.071000e+04, 1.207142e+06,
 6.031000e+01, 3.487000e+01],
[9.680000e+00, 4.671600e+02, 1.750000e+02, 2.128804e+05,
 3.607000e+01, 2.078000e+01],
[1.595590e+03, 1.504260e+03, 9.761100e+04, 1.056443e+06,
 5.231000e+01, 3.573000e+01],
[1.529760e+03, 1.026267e+04, 5.969000e+03, 4.310569e+06,
 5.971000e+01, 5.695000e+01],
[5.027200e+02, 1.285051e+04, 3.713400e+04, 7.483742e+06,
 5.447000e+01, 2.980000e+01],
[1.221500e+02, 7.247540e+03, 7.861000e+03, 5.539721e+06,
 4.746000e+01, 2.767000e+01],
[2.115000e+02, 5.025090e+03, 5.783700e+04, 3.711716e+06,
 5.891000e+01, 3.221000e+01],
[2.679700e+02, 7.526000e+03, 5.920500e+04, 9.094596e+06,
 5.720000e+01, 3.095000e+01],
[4.642100e+02, 2.821560e+03, 1.429000e+03, 9.439003e+05,
 5.157000e+01, 2.657000e+01],
[5.687800e+03, 3.186160e+03, 2.462937e+06, 1.858640e+06,
 6.113000e+01, 3.476000e+01],
[1.958000e+02, 5.573700e+02, 2.377460e+05, 9.329965e+05,
 4.223000e+01, 2.739000e+01],
[5.859000e+01, 4.496700e+02, 1.128060e+05, 4.564606e+05,
 5.217000e+01, 2.352000e+01],
[3.300000e+01, 9.772200e+02, 7.301000e+03, 1.207524e+06,
 4.774000e+01, 2.982000e+01],

[6.680000e+00, 3.962800e+02, 1.454000e+03, 9.977132e+05,
 5.671000e+01, 2.498000e+01],
[9.410000e+00, 1.073820e+03, 5.050000e+02, 9.429698e+05,
 5.072000e+01, 3.095000e+01],
[3.066000e+01, 1.910000e+03, 3.151000e+03, 1.008884e+06,
 5.770000e+01, 3.101000e+01],
[2.570000e+00, 7.514000e+01, 2.424000e+03, 3.623283e+05,
 4.610000e+01, 2.730000e+01],
[1.647100e+02, 8.293700e+02, 4.220800e+04, 6.322826e+05,
 6.440000e+01, 3.970000e+01],
```

```
       [2.000000e+00, 1.540900e+02, 5.952000e+03, 6.859893e+05,
        5.013000e+01, 2.375000e+01],
       [4.691000e+01, 2.681400e+03, 1.122800e+04, 1.819646e+06,
        5.103000e+01, 3.065000e+01],
       [1.400000e+00, 2.922100e+02, 1.011000e+03, 6.374603e+05,
        4.134000e+01, 2.401000e+01],
       [1.990000e+00, 1.283000e+02, 1.945000e+03, 1.623239e+05,
        4.974000e+01, 2.213000e+01],
       [3.800000e-01, 9.322000e+01, 1.600000e+01, 1.982748e+05,
        4.991000e+01, 1.892000e+01],
       [7.570000e+00, 1.300600e+02, 4.173000e+03, 4.160059e+05,
        3.810000e+01, 2.446000e+01],
       [6.600000e-01, 9.365000e+01, 5.160000e+02, 3.080294e+05,
        5.044000e+01, 2.337000e+01],
       [1.925000e+01, 2.438000e+02, 1.772000e+03, 2.371910e+05,
        5.116000e+01, 2.942000e+01],
       [1.258000e+01, 4.495600e+02, 2.557000e+03, 4.481810e+05,
        5.034000e+01, 3.258000e+01]])
```

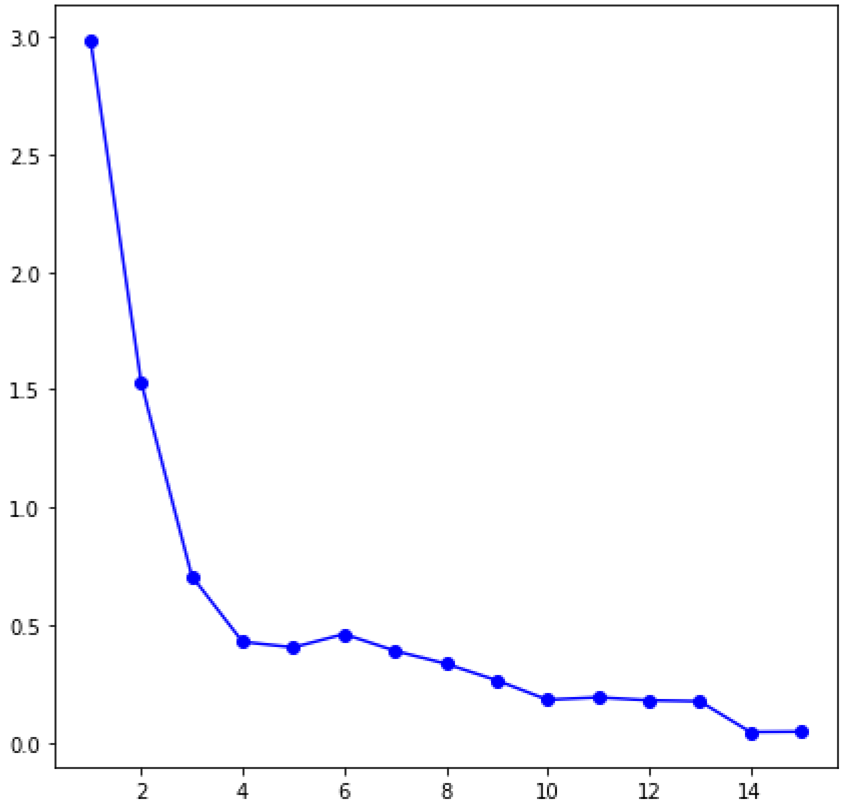
```
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler


# Menstandarkan ukuran (scaling)
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x_array)


# function returns WSS score for k values from 1 to kmax
def calculate_WSS(points, kmax):
  sse = []
  for k in range(1, kmax+1):
    kmeans = KMeans(n_clusters = k).fit(points)
    centroids = kmeans.cluster_centers_
    pred_clusters = kmeans.predict(points)
    curr_sse = 0

    # calculate square of Euclidean distance of each point from its cluster center
    for i in range(len(points)):
      curr_center = centroids[pred_clusters[i]]
      curr_sse += (points[i, 0] - curr_center[0]) ** 2 + (points[i, 1] - curr_cente

    sse.append(curr_sse)
  return sse
```

```
wss = calculate_WSS(x_scaled, 15)
wss
```

```
[2.9880854102444747,
 1.5313390318196365,
 0.7079544303628379,
 0.4286123405740351,
 0.4048822371786375,
 0.4616831809942087,
 0.3906615216091572,
 0.33517300723853044,
```

```
     0.26478441973721045,
     0.1821231656802822,
     0.19248796514677444,
     0.1796206590797465,
     0.1758057456702484 5,
     0.044913270769797765,
     0.046486995383640324]
```

```
xx=np.arange(1,16,1)
plt.figure(figsize=[7,7])
plt.plot(xx, wss, "b-o")
plt.show()
```



xx = np.arange(1,16,1) plt.figure(figsize=[7,7]) plt.plot(xx,wss, "b-o") plt.show()

Jumlah cluster: silhouette

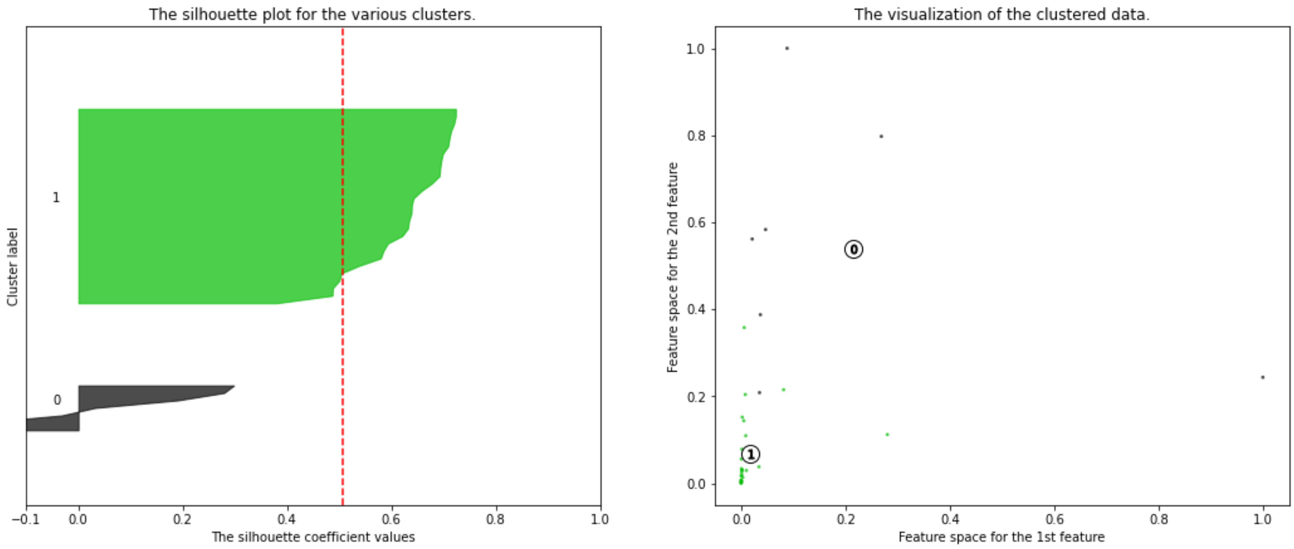
```
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import numpy as np

range_n_clusters = [2, 3, 4, 5, 6]

def silh(X):
    for n_clusters in range_n_clusters:
```

```
    # Create a subplot with 1 row and 2 columns
    fig, (ax1, ax2) = plt.subplots(1, 2)
    fig.set_size_inches(18, 7)

    # The 1st subplot is the silhouette plot
    # The silhouette coefficient can range from -1, 1 but in this example all
    # lie within [-0.1, 1]
    ax1.set_xlim([-0.1, 1])
    # The (n_clusters+1)*10 is for inserting blank space between silhouette
    # plots of individual clusters, to demarcate them clearly.
    ax1.set_ylim([0, len(X) + (n_clusters + 1) * 10])

    # Initialize the clusterer with n_clusters value and a random generator
    # seed of 10 for reproducibility.
    clusterer = KMeans(n_clusters=n_clusters, random_state=10)
    cluster_labels = clusterer.fit_predict(X)

    # The silhouette_score gives the average value for all the samples.
    # This gives a perspective into the density and separation of the formed
    # clusters
    silhouette_avg = silhouette_score(X, cluster_labels)
    print(
        "For n_clusters =",
        n_clusters,
        "The average silhouette_score is :",
        silhouette_avg,
    )

    # Compute the silhouette scores for each sample
    sample_silhouette_values = silhouette_samples(X, cluster_labels)

    y_lower = 10
    for i in range(n_clusters):
        # Aggregate the silhouette scores for samples belonging to
        # cluster i, and sort them
        ith_cluster_silhouette_values = sample_silhouette_values[cluster_labels

        ith_cluster_silhouette_values.sort()

        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i

        color = cm.nipy_spectral(float(i) / n_clusters)
        ax1.fill_betweenx(
            np.arange(y_lower, y_upper),
            0,
            ith_cluster_silhouette_values,
            facecolor=color,
            edgecolor=color,
            alpha=0.7,
        )

        # Label the silhouette plots with their cluster numbers at the middle
        ax1.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))
```

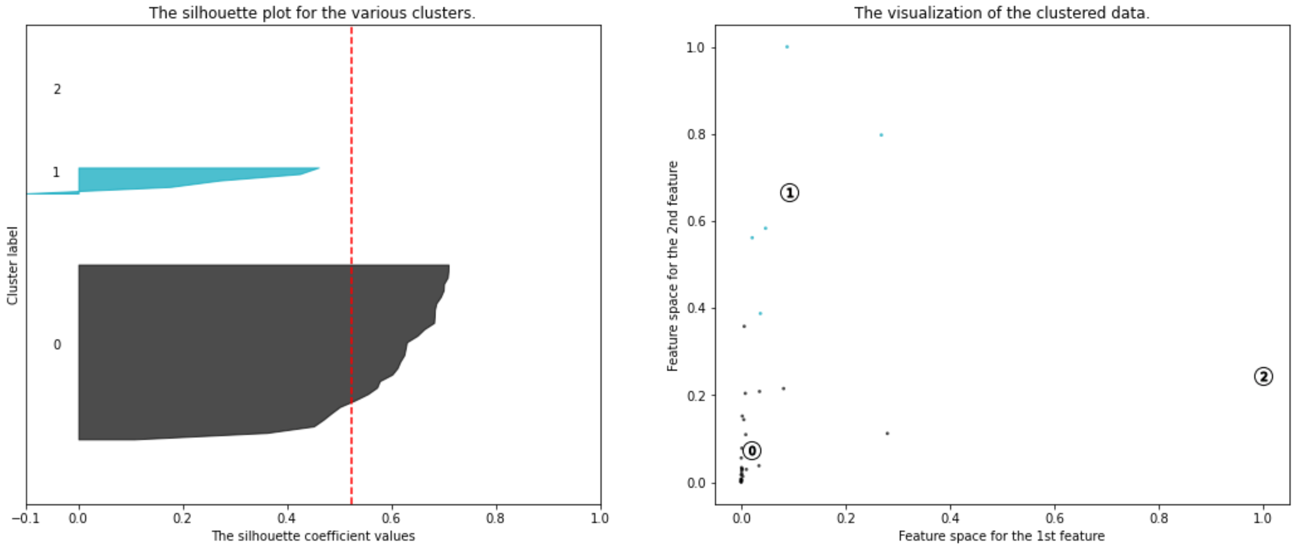
```
            # Compute the new y_lower for next plot
            y_lower = y_upper + 10  # 10 for the 0 samples

        ax1.set_title("The silhouette plot for the various clusters.")
        ax1.set_xlabel("The silhouette coefficient values")
        ax1.set_ylabel("Cluster label")

        # The vertical line for average silhouette score of all the values
        ax1.axvline(x=silhouette_avg, color="red", linestyle="--")

        ax1.set_yticks([])  # Clear the yaxis labels / ticks
        ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])

        # 2nd Plot showing the actual clusters formed
        colors = cm.nipy_spectral(cluster_labels.astype(float) / n_clusters)
        ax2.scatter(
            X[:, 0], X[:, 1], marker=".", s=30, lw=0, alpha=0.7, c=colors, edgecol
        )

        # Labeling the clusters
        centers = clusterer.cluster_centers_
        # Draw white circles at cluster centers
        ax2.scatter(
            centers[:, 0],
            centers[:, 1],
            marker="o",
            c="white",
            alpha=1,
            s=200,
            edgecolor="k",
        )

        for i, c in enumerate(centers):
            ax2.scatter(c[0], c[1], marker="$%d$" % i, alpha=1, s=50, edgecolor="k'

        ax2.set_title("The visualization of the clustered data.")
        ax2.set_xlabel("Feature space for the 1st feature")
        ax2.set_ylabel("Feature space for the 2nd feature")

        plt.suptitle(
            "Silhouette analysis for KMeans clustering on sample data with n_cluste
            % n_clusters,
            fontsize=14,
            fontweight="bold",
        )

    plt.show()

silh(x_scaled)
```

```
For n_clusters = 4 The average silhouette_score is : 0.2917577906968881
For n_clusters = 5 The average silhouette_score is : 0.2940985767639586
For n_clusters = 6 The average silhouette_score is : 0.27633665389663226
```

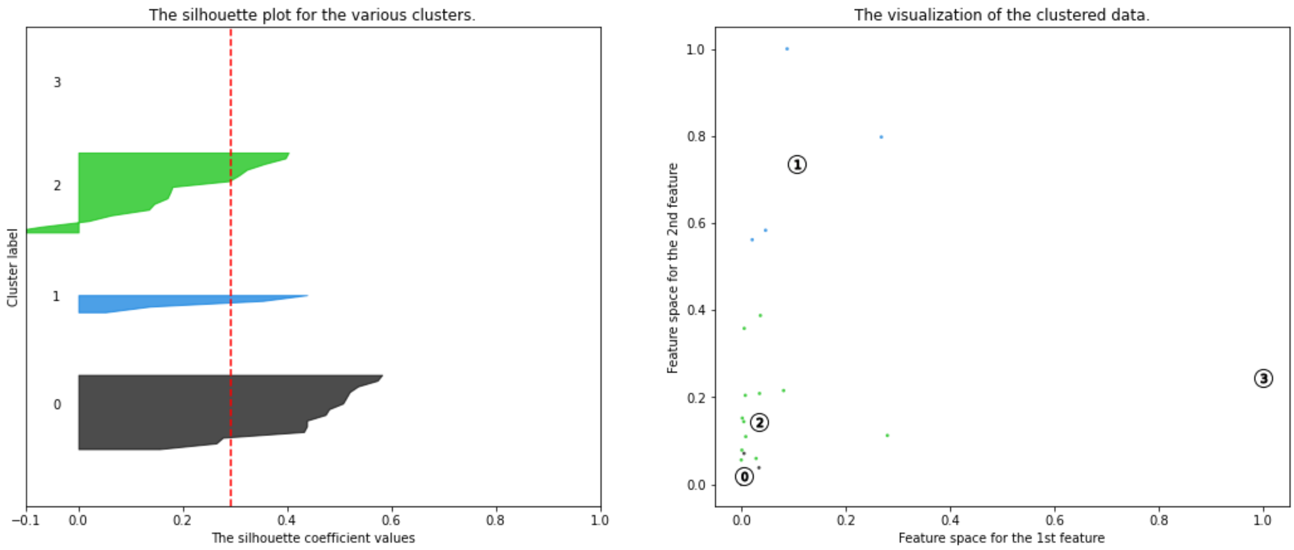
Silhouette analysis for KMeans clustering on sample data with n_clusters = 2



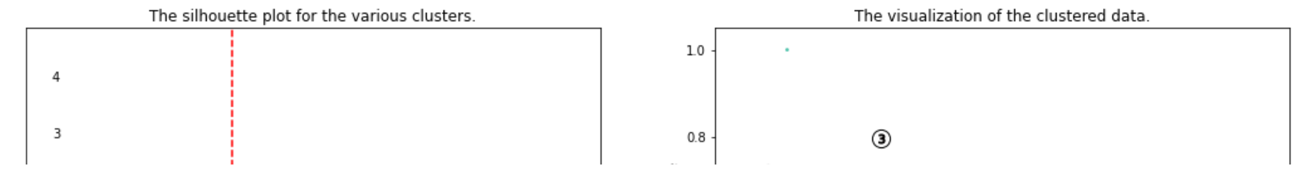
Silhouette analysis for KMeans clustering on sample data with n_clusters = 3

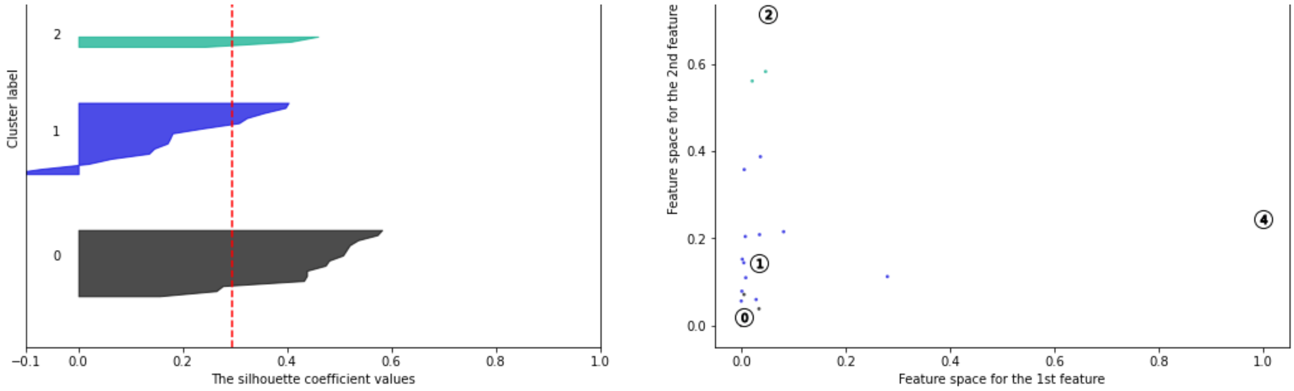


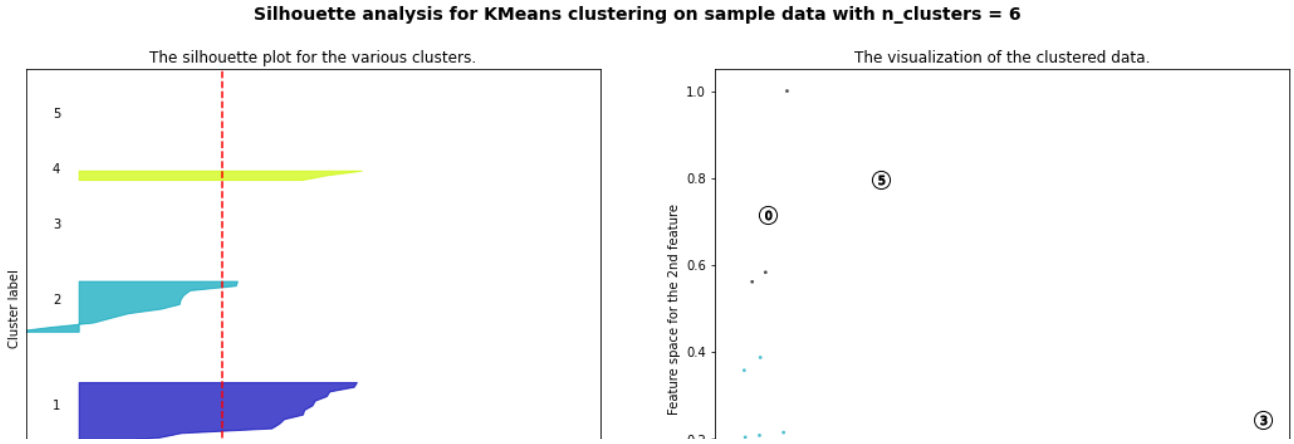
Silhouette analysis for KMeans clustering on sample data with n_clusters = 4



Silhouette analysis for KMeans clustering on sample data with n_clusters = 5








```
# Menentukan model kmeans
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters = 3, random_state = 123)
```

```
x_scaled
```

```
[2.40355029e-03, 1.51746681e-01, 4.63271051e-04, 1.11691862e-01,
 6.32545005e-01, 3.57349461e-01],
[3.16487968e-05, 7.56768688e-03, 2.20469922e-04, 3.17742895e-02,
 9.87998588e-01, 2.66105706e-01],
[4.62424087e-04, 5.56101311e-02, 4.34199879e-03, 1.16971146e-01,
 8.55630074e-01, 4.19405732e-01],
[1.63518784e-03, 3.06856083e-02, 6.45574909e-05, 5.65998208e-03,
 0.00000000e+00, 4.89087562e-02],
[2.80480429e-01, 1.11865253e-01, 3.96257127e-02, 1.00099850e-01,
 5.73243911e-01, 4.42019458e-01],
[2.68905760e-01, 7.97435221e-01, 2.41704870e-03, 4.64410964e-01,
 8.34451112e-01, 1.00000000e+00],
[8.83247589e-02, 1.00000000e+00, 1.50707229e-02, 8.19659099e-01,
 6.49488175e-01, 2.86089929e-01],
[2.14104110e-02, 5.61424053e-01, 3.18524224e-03, 6.02018953e-01,
 4.02047300e-01, 2.30081515e-01],
[3.71205221e-02, 3.87460402e-01, 2.34765955e-02, 3.97367216e-01,
 8.06212496e-01, 3.49460952e-01],

[4.70494530e-02, 5.83220682e-01, 2.40320335e-02, 1.00000000e+00,
 7.45852453e-01, 3.16329214e-01],
[8.15536746e-02, 2.14977727e-01, 5.73709023e-04, 8.75002901e-02,
 5.47123191e-01, 2.01156981e-01],
[1.00000000e+00, 2.43517018e-01, 1.00000000e+00, 1.89908691e-01,
 8.84574656e-01, 4.16513279e-01],
[3.43600437e-02, 3.77468520e-02, 9.65235994e-02, 8.62795705e-02,
 2.17437346e-01, 2.22718906e-01],
[1.02348692e-02, 2.93165677e-02, 4.57952163e-02, 3.29296619e-02,
 5.68302153e-01, 1.20957139e-01],
```

```
       [5.73546529e-03, 7.06108708e-02, 2.95786994e-03, 1.17013912e-01,
        4.11930815e-01, 2.86615830e-01],
       [1.10770789e-03, 2.51374324e-02, 5.83859572e-04, 9.35248379e-02,
        7.28556301e-01, 1.59347883e-01],
       [1.58771464e-03, 7.81722956e-02, 1.98544736e-04, 8.73961173e-02,
        5.17119661e-01, 3.16329214e-01],
       [5.32403093e-03, 1.43624803e-01, 1.27287883e-03, 9.47754491e-02,
        7.63501588e-01, 3.17906916e-01],
       [3.85060361e-04, 0.00000000e+00, 9.77700868e-04, 2.23912122e-02,
        3.54041652e-01, 2.20352353e-01],
       [2.88935932e-02, 5.90378204e-02, 1.71308783e-02, 5.26135674e-02,
        1.00000000e+00, 5.46410728e-01],
       [2.84839171e-04, 6.17986015e-03, 2.41014633e-03, 5.86262257e-02,
        4.96293682e-01, 1.27004996e-01],
       [8.18121398e-03, 2.04006616e-01, 4.55231816e-03, 1.85543172e-01,
        5.28062125e-01, 3.08440705e-01],
       [1.79343182e-04, 1.69912887e-02, 4.03991845e-04, 5.31932295e-02,
        1.86021885e-01, 1.33841704e-01],
       [2.83080905e-04, 4.16113193e-03, 7.83216352e-04, 0.00000000e+00,
        4.82527356e-01, 8.44070471e-02],
       [0.00000000e+00, 1.41522320e-03, 0.00000000e+00, 4.02483261e-03,
        4.88528062e-01, 0.00000000e+00],
       [1.26419361e-03, 4.29889702e-03, 1.68783327e-03, 2.84006127e-02,
        7.16554889e-02, 1.45674468e-01],
       [4.92314617e-05, 1.44888172e-03, 2.03010978e-04, 1.63122550e-02,
        5.07236145e-01, 1.17012885e-01],
       [3.31784887e-03, 1.32019660e-02, 7.12974553e-04, 8.38164122e-03,
        5.32650900e-01, 2.76097818e-01],
       [2.14508512e-03, 2.93079574e-02, 1.03170179e-03, 3.20027309e-02,
        5.03706318e-01, 3.59190113e-01]])
```

```
# Menginputkan data
kmeans.fit(x_scaled)
```

```
KMeans(n_clusters=3, random_state=123)
```

```
kmeans.get_params
```

```
<bound method BaseEstimator.get_params of KMeans(n_clusters=3, random_state=1
```

```
#Menampilkan Label Cluster
kmeans.labels_
```

```
array([2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 2, 1, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], dtype=int32)
```

```
# Menampilkan pusat cluster
kmeans.cluster_centers_
```

```
array([[0.08304   , 0.58963798, 0.01279064, 0.63471575, 0.64825274,
        0.44088001],
       [1.        , 0.24351702, 1.        , 0.18990869, 0.88457466,
        0.41651328],
       [0.01806059, 0.06727088, 0.00948586, 0.06691637, 0.50690931,
        0.2334317 ]])
```

```
# Menambahkan kolom cluster ke data frame pariwisata
pariwisata['cluster'] = kmeans.labels_
pariwisata.head()
```

	Provinsi	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	huni
0	ACEH	9.21	312.73	10803.0	1030136.0	43.18	
1	SUMATERA UTARA	201.88	2736.09	21104.0	4850322.0	48.86	
2	SUMATERA BARAT	51.83	1470.99	3786.0	1167212.0	56.00	
3	RIAU	35.33	4646.98	65845.0	1413345.0	48.69	
4	JAMBI	6.02	502.86	2391.0	429609.7	45.49	

```
# Membentuk grup berdasarkan kolom cluster
grup = pariwisata.groupby('cluster')
```

```
profil = pd.DataFrame(grup.mean())
profil
```

cluster	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	hunian_r
0	472.663333	7607.983333	3.151833e+04	5.831778e+06	54.435000	35.6866
1	5687.800000	3186.160000	2.462937e+06	1.858640e+06	61.130000	34.7600
2	103.098148	934.550370	2.337893e+04	7.600391e+05	50.430741	27.7974

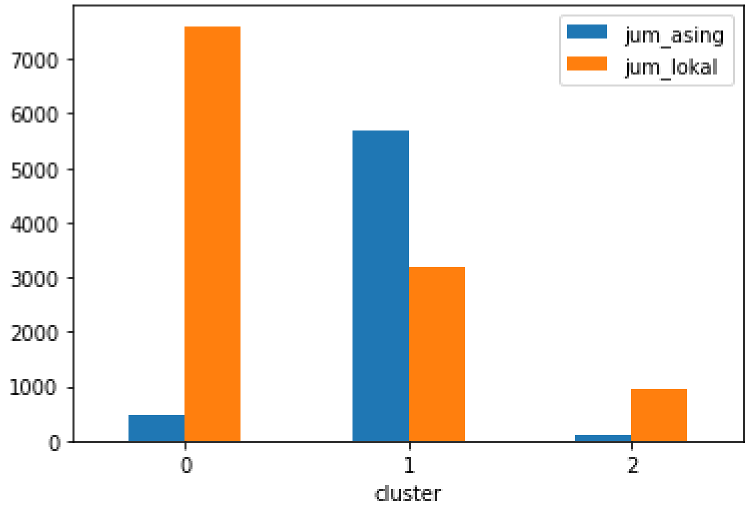
```
profil['cluster'] = profil.index
```

```
profil
```

cluster	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	hunian_r
0	472.663333	7607.983333	3.151833e+04	5.831778e+06	54.435000	35.6866
1	5687.800000	3186.160000	2.462937e+06	1.858640e+06	61.130000	34.7600
2	103.098148	934.550370	2.337893e+04	7.600391e+05	50.430741	27.7974

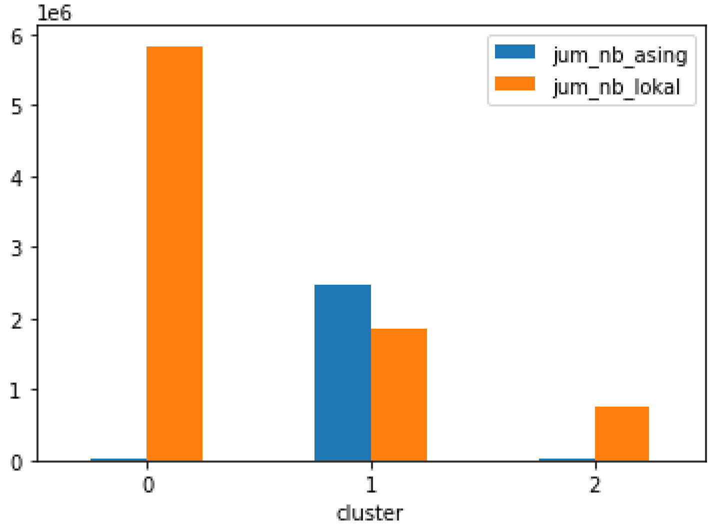
```
profil.plot(x = 'cluster', y = ['jum_asing', 'jum_lokal'], kind = 'bar')
plt.xticks(rotation = 0)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```



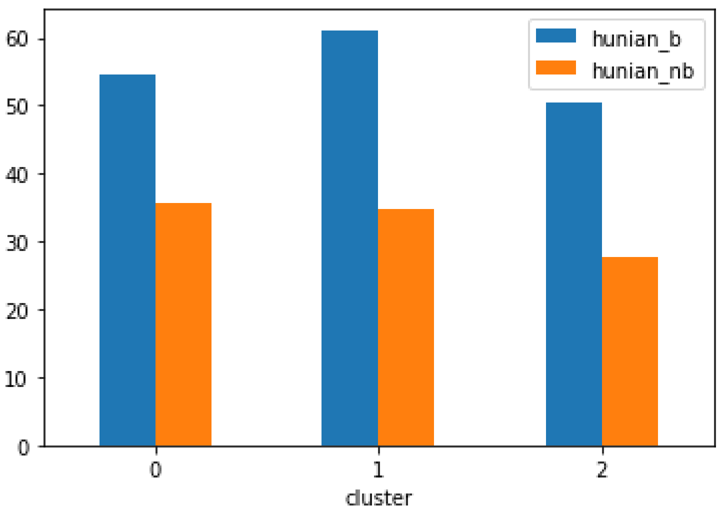

```
profil.plot(x = 'cluster', y = ['jum_nb_asing', 'jum_nb_lokal'], kind = 'b
plt.xticks(rotation = 0)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```




```
profil.plot(x = 'cluster', y = ['hunian_b', 'hunian_nb'], kind = 'bar')
plt.xticks(rotation = 0)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```



```
# Anggota cluster 0
pariwisata[pariwisata.cluster == 0]
```

	Provinsi	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	h
1	SUMATERA UTARA	201.88	2736.09	21104.0	4850322.0	48.86	
10	DKI JAKARTA	1529.76	10262.67	5969.0	4310569.0	59.71	
11	JAWA BARAT	502.72	12850.51	37134.0	7483742.0	54.47	
12	JAWA TENGAH	122.15	7247.54	7861.0	5539721.0	47.46	
13	DI YOGYAKARTA	211.50	5025.09	57837.0	3711716.0	58.91	
14	JAWA TIMUR	267.97	7526.00	59205.0	9094596.0	57.20	

```
# Anggota cluster 1
pariwisata[pariwisata.cluster == 1]
```

	Provinsi	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	hunia
16	BALI	5687.8	3186.16	2462937.0	1858640.0	61.13	

```
# Anggota cluster 2
pariwisata[pariwisata.cluster == 2]
```

	Provinsi	jum_asing	jum_lokal	jum_nb_asing	jum_nb_lokal	hunian_b	hu
0	ACEH	9.21	312.73	10803.0	1030136.0	43.18	
2	SUMATERA BARAT	51.83	1470.99	3786.0	1167212.0	56.00	
3	RIAU	35.33	4646.98	65845.0	1413345.0	48.69	
4	JAMBI	6.02	502.86	2391.0	429609.7	45.49	
5	SUMATERA SELATAN	14.05	2013.76	1157.0	1159986.0	53.99	
6	BENGKULU	0.56	171.82	559.0	446140.5	64.06	
7	LAMPUNG	3.01	785.58	10710.0	1207142.0	60.31	
8	KEP. BANGKA BELITUNG	9.68	467.16	175.0	212880.4	36.07	
9	KEP. RIAU	1595.59	1504.26	97611.0	1056443.0	52.31	
15	BANTEN	464.21	2821.56	1429.0	943900.3	51.57	
17	NUSA TENGGARA BARAT	195.80	557.37	237746.0	932996.5	42.23	
18	NUSA TENGGARA TIMUR	58.59	449.67	112806.0	456460.6	52.17	
19	KALIMANTAN BARAT	33.00	977.22	7301.0	1207524.0	47.74	
20	KALIMANTAN TENGAH	6.68	396.28	1454.0	997713.2	56.71	
21	KALIMANTAN SELATAN	9.41	1073.82	505.0	942969.8	50.72	
22	KALIMANTAN TIMUR	30.66	1910.00	3151.0	1008884.0	57.70	
23	KALIMANTAN UTARA	2.57	75.14	2424.0	362328.3	46.10	
24	SULAWESI UTARA	164.71	829.37	42208.0	632282.6	64.40	

Profilisasi cluster:

1. Kluster 0: tinggi, jumlah wisatawan lokal yang menginap di hotel non bintang dan bintang tinggi
2. Kluster 1: sedang, jumlah wisatawan lokal yang menginap di hotel non bintang dan bintang sedang
3. Kluster 2: rendah, jumlah wisatawan lokal yang menginap di hotel non bintang dan bintang rendah

29	SULAWESI	0.38	93.22	16.0	198274.8	40.91	