PENGGALIAN DATA – IS184943

TUGAS GROUP PROJECT #3

Analisis Klaster

SULIS AVANDHY PUTRA	-	05211940000084
MUHAMMAD ZUHDI AFI ABIYYI	-	05211940000135
AFLAH ADITYA	-	05211942000001

Program Studi Sarjana
Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya

Tahun 2022

Ringkasan Progress Tugas Group Project

B. Tugas

1. Setiap kelompok diharuskan mengumpulkan dua laporan berbeda untuk tugas analisis klaster dan tugas analisis asosiasi.
2. Lakukan ekplorasi data secara umum.
3. **Untuk tugas analisis klaster:**
   a. Lakukan praproses data yang diperlukan untuk keperluan analisis klaster
   b. Lakukan proses *clustering* menggunakan tiga metode yang berbeda: **partisional (K-means), metode hirarki (MIN, MAX, dan AVERAGE), dan metode berbasis densitas (DBScan).** Gunakan library python yang sesuai untuk masing-masing metode.
   c. Untuk metode DBScan, lakukan eskperimen untuk beberapa nilai *minimum points* dan *epsilon* yang berbeda untuk mendapatkan hasil terbaik.
   d. Jumlah klaster dari tugas ini sesuai dengan jumlah nilai atribut *"Kingdom"*. Untuk itu lakukan proses evaluasi menggunakan *entropy* dan *purity* baik untuk megukur kualitas masing-masing klaster yang dihasilkan maupun kualitas keseluruhan hasil klaster.
4. **Untuk tugas analisis asosiasi:**
   a. Lakukan praproses data yang diperlukan untuk keperluan analisis asosiasi, terutama praproses untuk mentrasformasikan data agar dapat diperlakukan sebagai item, sehingga dapat dilakukan analisis asosiasi. **Dalam hal ini setiap pasangan atribut dan nilainya dapat dianggap sebagai sebuah item.**
   b. Lakukan proses pembangkitan *frequent itemsets* dengan menggunakan algoritma *FP-growth*. Lakukan uji coba untuk berbagai nilai ambang batas *support* dan tentukan nilai ambang batas *support* yang pas menurut hasil uji coba. Kemudian lakukan perbandingan yang diperoleh menggunakan kedua algoritma tersebut berdasarkan waktu komputasi yang dibutuhkan oleh masing-masing algoritma.
   c. Bangkitkan sejumlah aturan asosiasi (*association rules*) yang menarik dari satu set *frequent itemsets* yang diperoleh. Salah satu aturan asosiasi yang harus dibangkiktkan adalah menjadikan atribut *"Ncodons"* sebagai target beserta data statistik berupa nilai rataan dari Ncodons dan juga juga simpangan baku dari aturan asosiasi yang dibangktkan. Lakukan analisis kemenarikan (*interestingness*) dari aturan yang dihasilkan menggunakan berbagai ukuran kemenarikan (selain hanya menggunakan *confidence*). Lakukan uji coba untuk berbagai nilai ambang batas ukuran kemenarikan dan buat kesimpulan dari hasil uji coba tersebut.

A. Pendahuluan

1. Dataset

Data yang digunakan pada tugas ini adalah data terkait DNA Codon yang diperoleh dari UCI Machine Learning Repository. Kodon adalah urutan trinukleotida DNA atau RNA yang sesuai dengan asam amino tertentu. Kode genetik menggambarkan hubungan antara urutan basa DNA (A, C, G, dan T) dalam gen dan urutan protein yang sesuai yang dikodekannya. Sel membaca urutan gen dalam kelompok tiga basa. Ada 64 kodon yang berbeda: 61 pertama menyatakan urutan sinyal asam amino sedangkan tiga sisanya menyatakan sinyal akhir dari urutan asam amino. Data ini terdiri dari terdiri 13.028 baris dan 69 kolom/atribut.

Berikut merupakan beberapa variabel/atribut dalam dataset yang dapat dikelompokkan menjadi empat kategori sebagai berikut:

1) Kingdom, 'Kingdom' adalah kode 3 huruf yang sesuai dengan `xxx' dalam nama basis data CUTG: 'arc'(archaea), 'bct'(bacteria), 'phg'(bacteriophage), 'plm' (plasmid), ' entri urutan pln' (tanaman), 'inv' (invertebrata), 'vrt' (vertebrata), 'mam' (mamalia), 'rod' (tikus), 'pri' (primata), dan 'vrl'(virus) . Perhatikan bahwa basis data CUTG tidak mengandung 'arc' dan 'plm' (ini telah dikuratori sendiri secara manual).
2) DNAtype. 'DNAType' dilambangkan sebagai bilangan bulat untuk komposisi genom dalam spesies: 0-genomic, 1-mitochondrial, 2-chloroplast, 3-cyanelle, 4-plastid, 5-nucleomorph,

6-secondary_endosymbiont, 7-chromoplast, 8-leucoplast, 9-NA, 10-proplastid, 11-apicoplast, and 12-kinetoplast.

3) SpeciesID, 'SpeciesID' adalah bilangan bulat, yang secara unik menunjukkan entri suatu organisme. Ini adalah pengidentifikasi aksesi untuk setiap spesies berbeda dalam basis data CUTG asli, diikuti oleh item pertama yang tercantum dalam setiap genom.
4) Ncodons, jumlah kodon (`Ncodons') adalah jumlah aljabar dari angka yang terdaftar untuk kodon yang berbeda dalam entri CUTG. Frekuensi kodon dinormalisasi ke jumlah total kodon, maka jumlah kejadian dibagi dengan 'Ncodons' adalah frekuensi kodon yang tercantum dalam file data.
5) SpeciesName, nama spesies ('SpeciesName') diwakili dalam string yang dibersihkan dari `koma' (yang sekarang diganti dengan `spasi'). Ini adalah label deskriptif nama spesies untuk interpretasi data.
6) Codon, frekuensi kodon ('Codon') termasuk 'UUU', 'UUA', 'UUG', 'CUU', dll., dicatat sebagai float (dengan desimal dalam 5 digit).

B. Pra Proses Data

1. Missing Value

Missing value merupakan suatu kondisi dimana suatu atribut memiliki nilai null. Kondisi ini perlu ditangani dengan tepat agar model yang dibangun dapat melakukan klasifikasi dengan baik. Ada beberapa cara penanganannya diantaranya yaitu menghapus kolom atau baris yang memiliki missing value atau mengganti nilainya dengan nilai lain seperti median, mean, atau mode.

```
[12] # cek nilai null
     data.isnull().values.any()

     False
```

Berdasarkan gambar di atas, dataset data tidak memiliki nilai null pada setiap atribut sehingga tidak diperlukan penanganan missing value lebih lanjut.

2. Categorical Encoding

Categorical Encoding merupakan proses mengubah atribut kategorikal menjadi integer. Pada tugas ini, dilakukan encoding terhadap atribut ’Kingdom’ secara manual dengan mendaftar seluruh nilai unik dari atribut ’Kingdom’ dan memasangkannya pada angka tertentu.

```
# mengubah kingdom menjadi angka
data.replace({'Kingdom' : {'vrl': 0 , 'arc': 1, 'bct': 2, 'phg': 3, 'plm': 4,
                           'pln': 5, 'inv': 6, 'vrt': 7, 'mam': 8, 'rod': 9, 'pri':10}}, inplace=True)
```

3. Menghapus Atribut Redundan

Tahapan ini dilakukan untuk memastikan tidak ada atribut yang menyimpan informasi yang sama sehingga bersifat redundan. Dari dataset yang ada, diketahui bahwa ‘SpeciesID’ dan ‘SpeciesName’ memiliki informasi yang sama sehingga salah satunya dapat dihapus.

```
[9]  # menghapus atribut redundan
     data.drop('SpeciesName', axis=1, inplace=True)
```

4. Menghapus Data Kodon yang Berisi String

Tahapan ini dilakukan untuk menghilangkan nilai pada atribut ‘Codon’ yang berisi string. Hal ini diketahui dari info dataset bahwa atribut ini hanya berisi data bertipe float namun ditemukan ada dua atribut yang belum bertipe float.

```
[4] dna.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13028 entries, 0 to 13027
Data columns (total 69 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Kingdom      13028 non-null  object
 1   DNAtype      13028 non-null  int64
 2   SpeciesID    13028 non-null  int64
 3   Ncodons      13028 non-null  int64
 4   SpeciesName  13028 non-null  object
 5   UUU          13028 non-null  object
 6   UUC          13028 non-null  object
```

```
[13] data = data.drop(data.index[data['UUC'] == '-'])
     data['UUC'].astype('float64')

0        0.01203
1        0.01357
2        0.02180
3        0.02245
4        0.01371
           ...
13023    0.03555
13024    0.03193
13025    0.03321
13026    0.02028
13027    0.03724
Name: UUC, Length: 13027, dtype: float64
```

```
[14] data = data.drop(data.index[data['UUU'] == 'non-B hepatitis virus'])
     data['UUU'].astype('float64')

0        0.01654
1        0.02714
2        0.01974
3        0.01775
4        0.02816
           ...
13023    0.02552
13024    0.01258
13025    0.01423
13026    0.01757
13027    0.01778
Name: UUU, Length: 13026, dtype: float64
```

5. Seleksi Fitur

   Pada proses ini akan dipilih fitur-fitur yang akan digunakan dalam proses klasterisasi. Fitur yang digunakan dalam clustering adalah seluruh atribut kodon dan yang tidak digunakan adalah fitur 'Kingdom', 'DNAtype', 'SpeciesID', dan 'Ncodons'. Penghapusan ini dilakukan karena fitur 'Kingdom', 'DNAtype', 'SpeciesID' merupakan fitur kategorik yang tidak tepat jika nantinya direduksi pada proses selanjutnya dan penghapusan 'Ncodons' karena perbedaan skala yang terlalu besar dengan fitur-fitur kodon.

```
# mengambil fitur yang digunakan
X = data.drop(columns=['Kingdom','DNAtype','SpeciesID','Ncodons'], axis=1)
y = data.Kingdom
```

6. Reduksi Dimensi

Pada tahap ini, atribut akan direduksi menjadi hanya dua komponen untuk agar mudah dalam visualisasi serta membantu dalam klasterisasi. Proses ini menggunakan principal component analysis dengan bantuan library dari sklearn.

```
[15] # dimentionality reduction
     from sklearn.decomposition import PCA

[17] # Create a PCA instance: pca
     pca = PCA(n_components = 2)
     pca_component = pca.fit_transform(X)

     # Plot the explained variances
     features = range(pca.n_components_)
     pca_variance = pca.explained_variance_ratio_
     pca_result = pd.DataFrame({'PCA Feature': features,'Variance (%)': pca_variance})
     fig = px.bar(pca_result, x= pca_result.columns[0], y= pca_result.columns[1], text= pca_result.columns[1])
     fig.update_layout(title_text= 'Variation per Principal Component')
     fig.show()
```

C. Implementasi Model Klasterisasi

1. Metode Partitional (K-Means)

Proses implementasi diawali dengan melakukan import beberapa library yang dibutuhkan sebelum melakukan proses klasterisasi. Pada tabel berikut merupakan list beberapa library yang dibutuhkan dalam menjalankan metode *k-means clustering*.

Library	**Fungsi**
`from sklearn.cluster import KMeans`	Library untuk penerapan metode K-means clustering
`import plotly.express as px`	Library ini berfungsi untuk memvisualisasikan data
`import matplotlib.pyplot as plt`	Library ini berfungsi untuk memvisualisasikan data
`from sklearn import metrics`	Library ini berfungsi untuk membantu dalam mengevaluasi performa klasterisasi
`import seaborn as sb`	Library ini berfungsi untuk memvisualisasikan data

Proses implementasi dilanjutkan dengan memanggil fungsi "`KMeans`" dari library sklearn. Kemudian didefinisikan parameter yang dibutuhkan yaitu n_clusters. Pada tugas ini, dilakukan penentuan nilai parameter n_clusters berdasarkan jumlah data kingdom,

```
n_cluster = 11
kmean = KMeans(n_clusters= n_cluster)

# fitting model
kmean.fit(X_pca)

# mengambil label hasil klaster
result_kmeans = pd.DataFrame(X_pca.copy())
result_kmeans['Cluster'] = kmean.labels_
result_kmeans
```

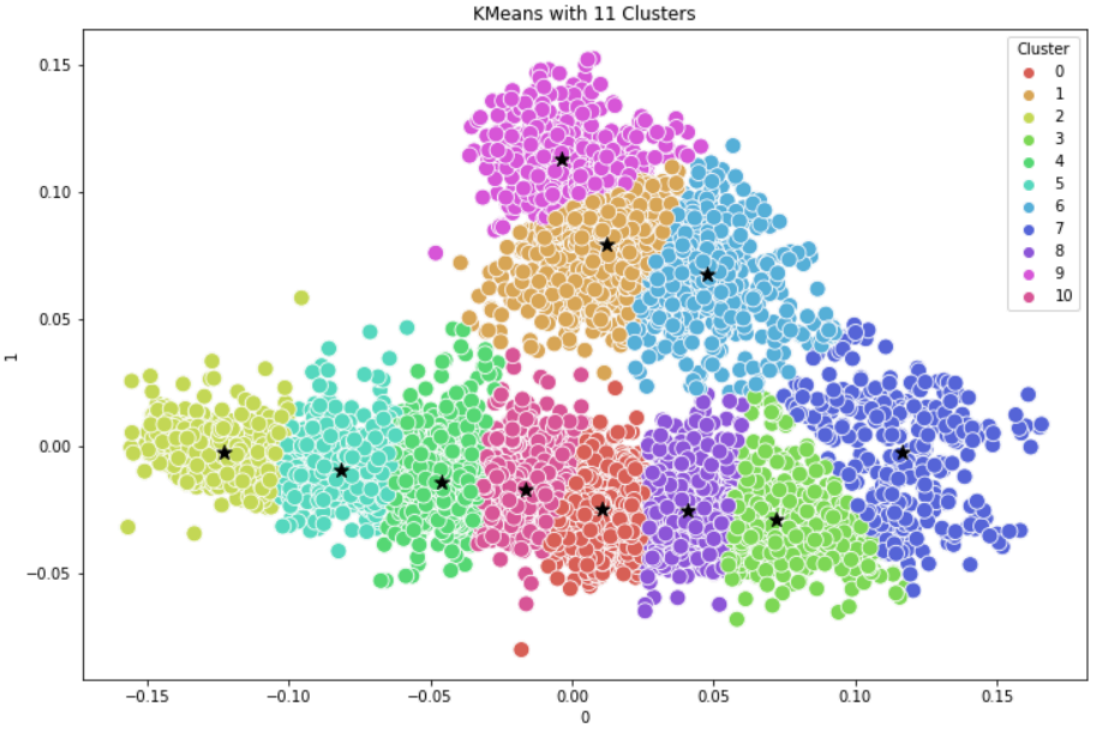
```
# ambil koordinat centroid
centers = kmean.cluster_centers_
print("Koordinat Centroid {}".format(centers))
```

```
Koordinat Centroid [[  -72148.72929302   -45140.63047661]
 [32534627.93298414   272917.24444554]
 [ 1380332.99627797  -124598.5943132 ]
 [ 6160099.43654649   179945.76943667]
 [23157821.22736607   335399.92067532]
 [  583094.11503641  -118136.00124044]
 [  -67754.5265929   -204082.54379029]
 [40580123.30221549   495157.54535435]
 [12996266.68664579    83199.42102235]
 [ 2460957.10286562   -66809.60181733]
 [  -67927.76428012    94808.53529377]]
```

Setelah membangun model, langkah berikutnya yang dilakukan yaitu melakukan plot dari hasil pengelompokkan sebelumnya dan melakukan plot nilai centroids. Untuk titik yang berbentuk bulat merepresentasikan data dari hasil pengelompokkan, sedangkan untuk titik koordinat yang berbentuk bintang merepresentasikan centroid.

```
# plot clustering result
plt.figure(figsize=(12, 8))
sb.scatterplot(
    result_kmeans.iloc[:,0], result_kmeans.iloc[:,1],
    s = 120,
    hue= result_kmeans.Cluster,
    # palette= ['g', 'r', 'b', 'y'])
    palette= sb.color_palette('hls', len(result_kmeans.Cluster.unique())))

# plot the centroids
centroids = kmean.cluster_centers_
plt.scatter(centroids[:, 0], centroids[:, 1], marker = "*", s = 100, color = 'black')
# define plot title
plt.title('KMeans with ' + str(n_cluster) + ' Clusters')
plt.show()
```



2. Metode Hirarki (MIN, MAX, AVERAGE)

Proses implementasi diawali dengan melakukan import beberapa library yang dibutuhkan sebelum melakukan proses klasterisasi . Pada tabel berikut merupakan list beberapa library yang dibutuhkan dalam menjalankan metode *agglomerative hierarchical clustering*.

Library	**Fungsi**
`from sklearn.cluster import AgglomerativeClustering`	Library untuk penerapan metode Agglomerative hierarchical clustering
`import plotly.express as px`	Library ini berfungsi untuk memvisualisasikan data
`import matplotlib.pyplot as plt`	Library ini berfungsi untuk memvisualisasikan data
`from sklearn import metrics`	Library ini berfungsi untuk membantu dalam mengevaluasi performa klasterisasi
`import seaborn as sb`	Library ini berfungsi untuk memvisualisasikan data

Proses implementasi dilanjutkan dengan memanggil fungsi "`AgglomerativeClustering`" dari library sklearn. Kemudian didefinisikan parameter yang dibutuhkan yaitu n_clusters dan linkage. Pada tugas ini, dilakukan eksperimen terhadap parameter linkage atau metode pencari kesamaan, beberapa value parameter yang digunakan yaitu "single" (MIN), complete" (MAX), dan "average". Berikut merupakan cara dan hasil implementasinya:

- Min

Langkah awal yang dilakukan yaitu membangun model menggunakan metode Agglomerative hierarchical clustering dengan menerapkan "single" (MIN) sebagai nilai pada parameter linkage.

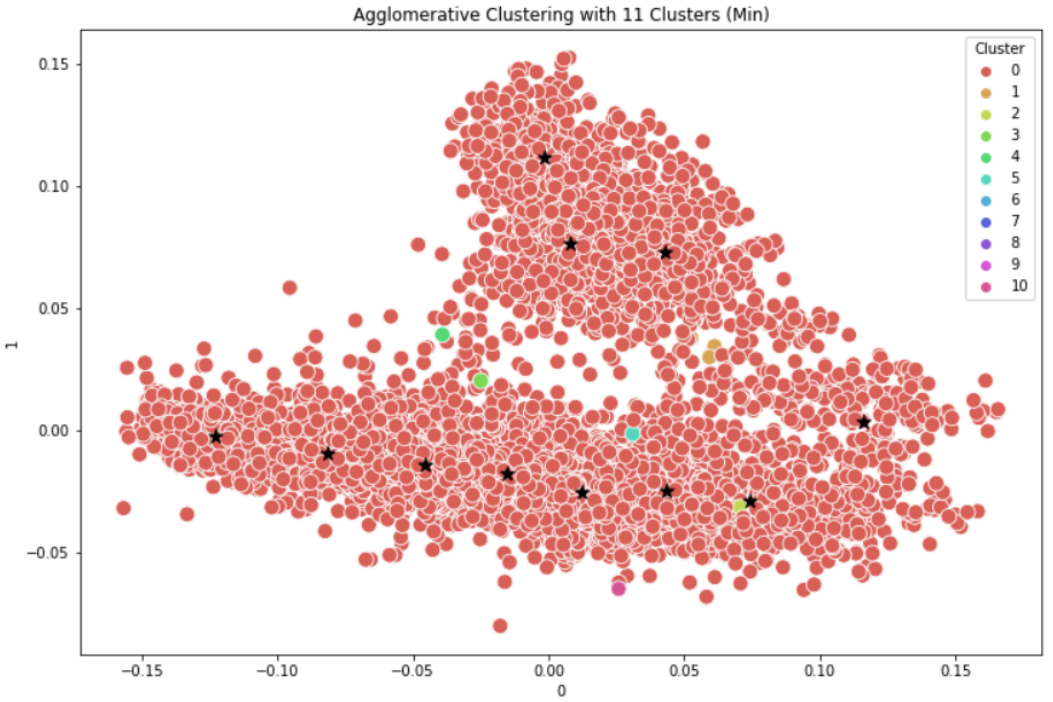
```
[52] # membuat model
     n_cluster= 11
     agglo_model_min = AgglomerativeClustering(n_clusters= n_cluster, linkage="single")
     agglo_model_min.fit(X)

     result_agglo_min = pd.DataFrame(X_pca.copy())
     result_agglo_min['Cluster'] = agglo_model_min.labels_
     result_agglo_min
```

Setelah membangun model, langkah berikutnya yang dilakukan yaitu melakukan plot dari hasil pengelompokkan sebelumnya dan melakukan plot nilai centroids. Untuk titik yang berbentuk bulat merepresentasikan data dari hasil pengelompokkan, sedangkan untuk titik koordinat yang berbentuk bintang merepresentasikan centroid.

```
# plot clustering result
plt.figure(figsize=(12, 8))
sb.scatterplot(
    result_agglo_min.iloc[:,0], result_agglo_min.iloc[:,1],
    s = 120,
    hue= result_agglo_min.Cluster,
    # palette= ['g', 'r', 'b', 'y'])
    palette= sb.color_palette('hls', n_cluster))

# plot the centroids
centroids_vq = centroids
plt.scatter(centroids_vq[:, 0], centroids_vq[:, 1], marker = "*", s = 100, color = 'black')
# define plot title
plt.title('Agglomerative Clustering with ' + str(n_cluster) + ' Clusters (Min)')
plt.show()
```



- Max

  Langkah awal yang dilakukan yaitu membangun model menggunakan metode Agglomerative hierarchical clustering dengan menerapkan “complete" (MAX) sebagai nilai pada parameter linkage.

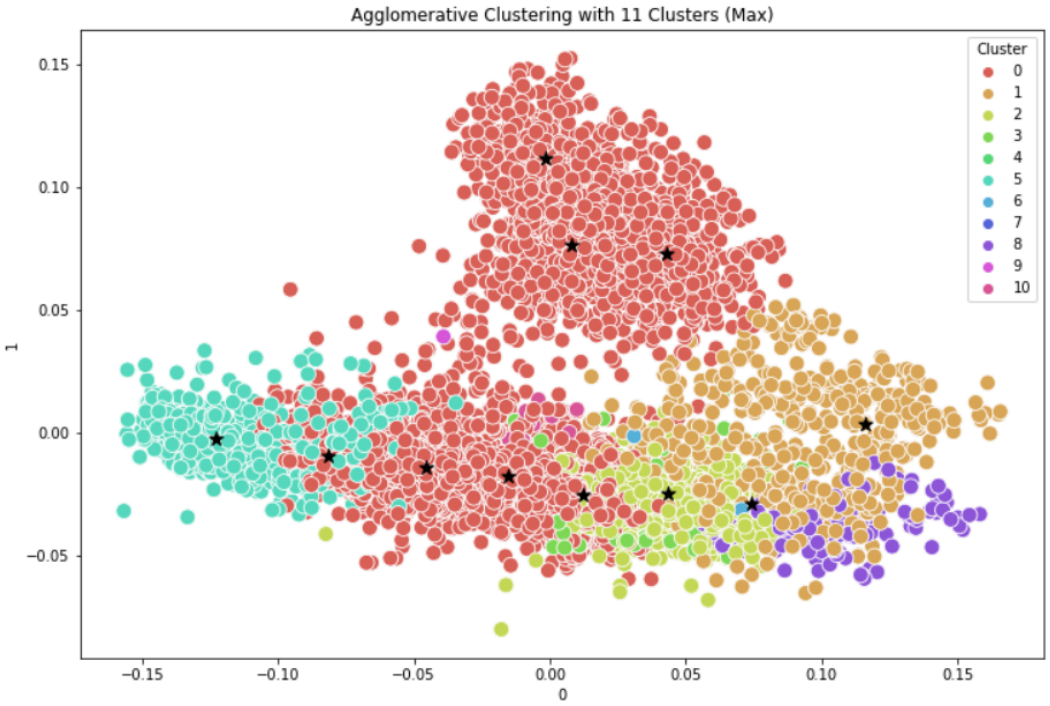
```
[53] # membuat model
     n_cluster= 11
     agglo_model_max = AgglomerativeClustering(n_clusters= n_cluster, linkage="complete")
     agglo_model_max.fit(X)

     result_agglo_max = pd.DataFrame(X_pca.copy())
     result_agglo_max['Cluster'] = agglo_model_max.labels_
     result_agglo_max
```

Setelah membangun model, langkah berikutnya yang dilakukan yaitu melakukan plot dari hasil pengelompokkan sebelumnya dan melakukan plot nilai centroids. Untuk titik yang berbentuk bulat merepresentasikan data dari hasil pengelompokkan, sedangkan untuk titik koordinat yang berbentuk bintang merepresentasikan centroid.

```
# plot clustering result
plt.figure(figsize=(12, 8))
sb.scatterplot(
    result_agglo_max.iloc[:,0], result_agglo_max.iloc[:,1],
    s = 120,
    hue= result_agglo_max.Cluster,
    # palette= ['g', 'r', 'b', 'y'])
    palette= sb.color_palette('hls', n_cluster))

# plot the centroids
centroids_vq = centroids
plt.scatter(centroids_vq[:, 0], centroids_vq[:, 1], marker = "*", s = 100, color = 'black')
# define plot title
plt.title('Agglomerative Clustering with ' + str(n_cluster) + ' Clusters (Max)')
plt.show()
```



- Average

  Langkah awal yang dilakukan yaitu membangun model menggunakan metode Agglomerative hierarchical clustering dengan menerapkan “average" (AVG) sebagai nilai pada parameter linkage.

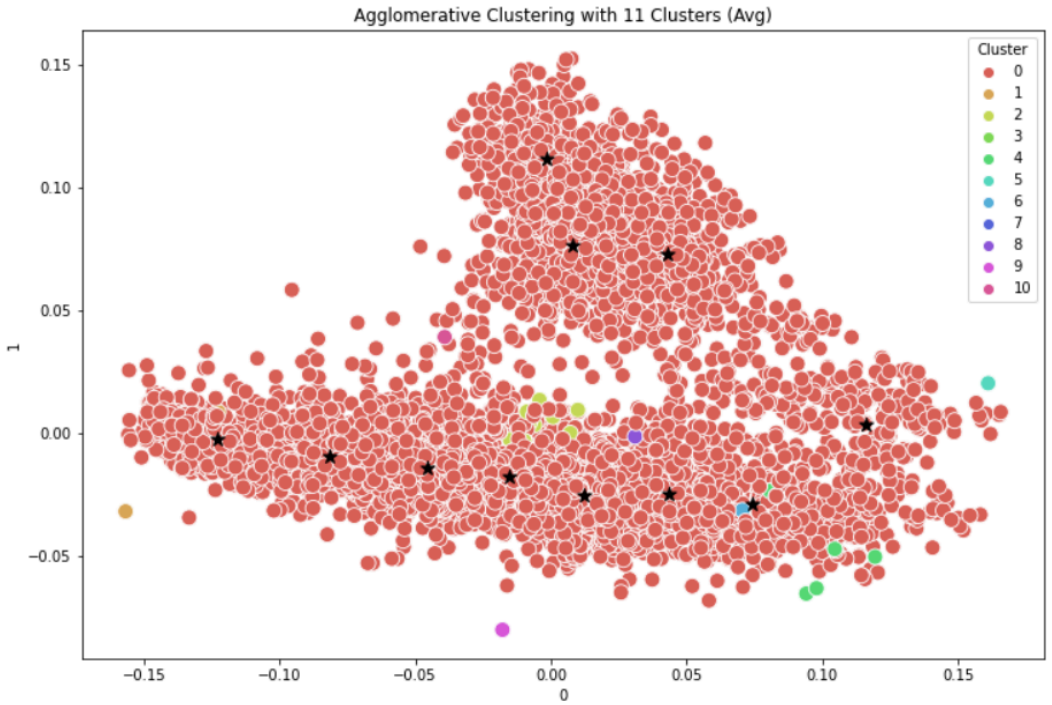
```
# membuat model
n_cluster= 11
agglo_model_avg = AgglomerativeClustering(n_clusters= n_cluster, linkage="average")
agglo_model_avg.fit(X)

result_agglo_avg = pd.DataFrame(X_pca.copy())
result_agglo_avg['Cluster'] = agglo_model_avg.labels_
result_agglo_avg
```

Setelah membangun model, langkah berikutnya yang dilakukan yaitu melakukan plot dari hasil pengelompokkan sebelumnya dan melakukan plot nilai centroids. Untuk titik yang berbentuk bulat merepresentasikan data dari hasil pengelompokkan, sedangkan untuk titik koordinat yang berbentuk bintang merepresentasikan centroid.

```
# plot clustering result
plt.figure(figsize=(12, 8))
sb.scatterplot(
    result_agglo_avg.iloc[:,0], result_agglo_avg.iloc[:,1],
    s = 120,
    hue= result_agglo_avg.Cluster,
    # palette= ['g', 'r', 'b', 'y'])
    palette= sb.color_palette('hls', n_cluster))

# plot the centroids
centroids_vq = centroids
plt.scatter(centroids_vq[:, 0], centroids_vq[:, 1], marker = "*", s = 100, color = 'black')
# define plot title
plt.title('Agglomerative Clustering with ' + str(n_cluster) + ' Clusters (Average)')
plt.show()
```



3. Metode Berbasis Densitas (DBScan)

Proses implementasi diawali dengan melakukan import beberapa library yang dibutuhkan sebelum melakukan proses klasterisasi . Pada tabel berikut merupakan list beberapa library yang dibutuhkan dalam menjalankan metode DBScan.

Library	**Fungsi**

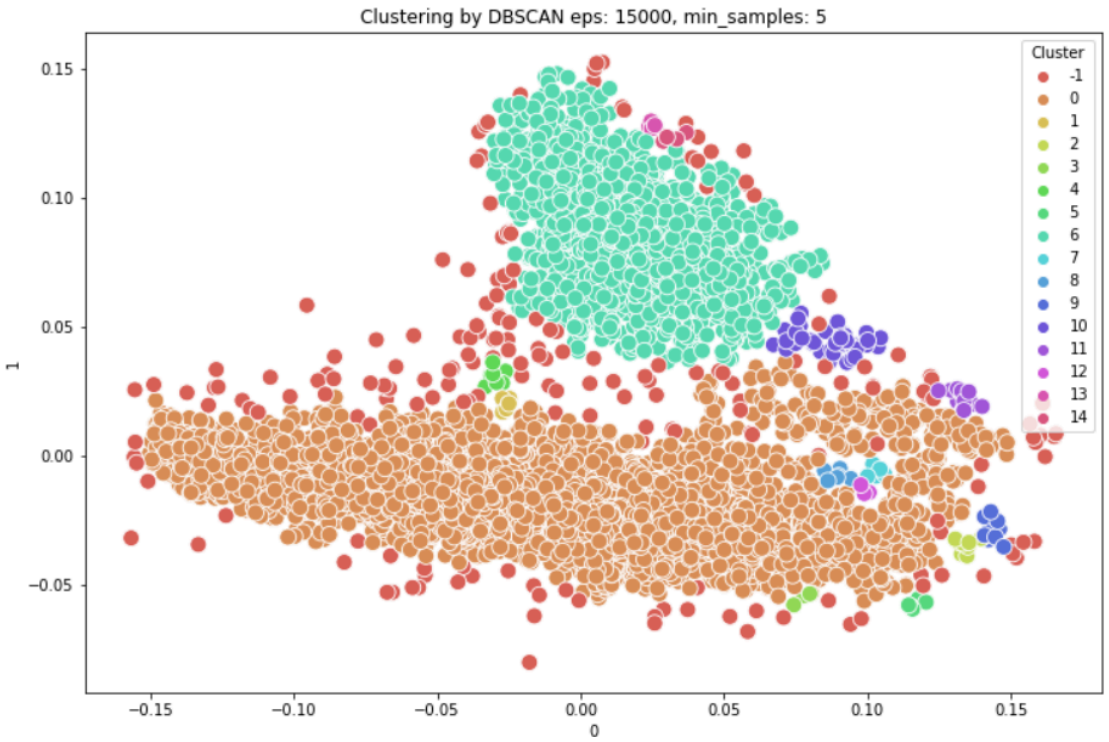
`from sklearn.cluster import DBSCAN`	Library untuk penerapan metode clustering
`import plotly.express as px`	Library ini berfungsi untuk memvisualisasikan data
`import matplotlib.pyplot as plt`	Library ini berfungsi untuk memvisualisasikan data
`from sklearn import metrics`	Library ini berfungsi untuk membantu dalam mengevaluasi performa klasterisasi
`import seaborn as sb`	

Proses implementasi dilanjutkan dengan memanggil fungsi DBScan dari library sklearn. Kemudian didefinisikan parameter yang dibutuhkan yaitu epsilon dan min_sample. Pada tugas ini, dilakukan eksperimen terhadap beberapa kombinasi nilai epsilon dan min_sample. Parameter epsilon yang akan digunakan yaitu 0.005 dan 0.0075 sementara parameter min_sample adalah 5 dan 10.

Selanjutnya adalah proses untuk mengklasterisasi dataset yang telah direduksi menggunakan fungsi yang telah dipanggil sebelumnya. Kemudian hasil klasterisasi dimasukkan ke dalam tabel sehingga dapat divisualisasikan. Berdasarkan hasil eksperimen, dari keempat kombinasi, epsilon 0.005 dan min_sample 5 memberikan hasil klasterisasi yang mendekati jumlah atribut Kingdom.

```
[154] # menggunakan DBSCAN
      dbscan_1 = DBSCAN(eps=0.005, min_samples=5)
      dbscan_1.fit(X_pca)

      result_dbscan_1 = pd.DataFrame(X_pca.copy())
      result_dbscan_1['Cluster'] = dbscan_1.labels_
```



D. Evaluasi Model Klaster

1. Menentukan Fungsi (Define Function)

Pada tahapan ini, dilakukan pendefinisian beberapa fungsi untuk mengevaluasi performa klasterisasi. Ada 4 fungsi yang dibuat yaitu ‘entropy_score’ yang berfungsi untuk mengukur entropi keseluruhan hasil klaster, ‘purity_score’ yang berfungsi untuk mengukur purity keseluruhan hasil klaster, ‘entropy_cluster’ yang berfungsi untuk mengukur entropi masing-masing hasil klaster, dan ‘purity_cluster’ yang berfungsi untuk mengukur purity masing-masing hasil klaster.

- Import library

  Ada beberapa library yang digunakan dalam evaluasi performa. Berikut merupakan library dan fungsi yang dibutuhkan dalam melakukan proses evaluasi model dengan mendapatkan nilai dari ‘entropy dan purity’ pada masing-masing model.

Library	**Fungsi**
`from scipy.stats import entropy as en`	Library untuk menghitung nilai entropy
`from sklearn import metrics`	Library ini berfungsi untuk membantu dalam mengevaluasi performa klasterisasi

- Fungsi ‘entropy_score’

```
[104] def entropy_score(y_true, y_pred):
        # compute contingency matrix (also called confusion matrix)
        entropy_list = entropy_cluster(y_true, y_pred)
        contingency_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
        a= []
        total = np.sum(contingency_matrix)
        for i in range(len(contingency_matrix)):
          total_cluster = np.sum(contingency_matrix[:,i])
          p = total_cluster*entropy_list[i]/total
          a.append(p)
        print('Nilai entropy keseluruhan adalah {0:.4f} \nNilai entropy masing-masing klaster adalah {1}'.format(np.sum(a), entropy_list))
```

- Fungsi ‘purity_score’

```
[113] def purity_score(y_true, y_pred):
        # compute contingency matrix (also called confusion matrix)
        purity_list = purity_cluster(y_true, y_pred)
        contingency_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
        a= []
        total = np.sum(contingency_matrix)
        for i in range(len(contingency_matrix)):
          total_cluster = np.sum(contingency_matrix[:,i])
          p = total_cluster*purity_list[i]/total
          a.append(p)
        print('Nilai purity keseluruhan adalah {0:.4f} \nNilai purity masing-masing klaster adalah {1}'.format(np.sum(a), purity_list))
```

- Fungsi ‘entropy_cluster’

```
def entropy_cluster(y_true, y_pred):
  # Membuat matriks kontingensi
  contingency_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
  # Menghitung nilai entropy
  entropy_list = []
  for i in range(len(contingency_matrix)):
    sum = np.sum(contingency_matrix[:,i])
    a = []
    for j in range(len(contingency_matrix[:,i])):
      a.append(contingency_matrix[:,i][j]/sum)
    entropy_list.append(en(a, base=2))
  # Mengembalikan nilai entropy
  return entropy_list
```

- Fungsi ‘purity_cluster’

```
def purity_cluster(y_true, y_pred):
  # Membuat matriks kontingensi
  contingency_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
  # Menghitung nilai purity
  purity_list = []
  for i in range(len(contingency_matrix)):
    sum = np.sum(contingency_matrix[:,i])
    a = []
    for j in range(len(contingency_matrix[:,i])):
      a.append(contingency_matrix[:,i][j]/sum)
    purity_list.append(max(a))
  # Mengembalikan nilai purity
  return purity_list
```

2. Evaluasi Model K Means

Pengukuran Entropy adalah untuk mengukur kemurnian dari cluster yang dihasilkan dengan memperhatikan pada kategori ada. Nilai Entropy yang lebih kecil menghasilkan cluster yang lebih bagus kualitasnya. Keakuratan hasil Entropy berada pada jangkauan 0 – 1 dimana semakin kecil hasil Entropy-nya, maka kualitas cluster semakin baik. Purity adalah ukuran “kemurnian” suatu cluster, yaitu seberapa murni solusi clustering yang diperoleh.

Nilai kemurnian (purity) dari sebuah cluster berkisaran antara 0 dan 1. Clustering buruk jika nilai kemurnian mendekati 0, dan baik jika nilai kemurnian mendekati 1.

- Mendapatkan nilai 'purity_score' pada keseluruhan klaster dan nilai entropy masing-masing klaster

  Nilai kemurnian (purity) dari sebuah cluster berkisaran antara 0 dan 1. Clustering buruk jika nilai purity mendekati 0, dan baik jika nilai purity mendekati 1. Nilai kemurnian dari keseluruhan cluster K means yang terbentuk dalam percobaan ini adalah 0.4740. Dapat dikatakan hasil dari penelitian ini baik karena nilai purity (kemurnian) mendekati 1.

  Cluster/kelompok 1 sebesar 0.4987, cluster 2 sebesar 0.7911, cluster 3 sebesar 0.2387, cluster 4 sebesar 0.4208, cluster 5 sebesar 0.5031, cluster 6 sebesar 0.7387, cluster 7 sebesar 0.8429, cluster 8 sebesar 0.6875, cluster 9 sebesar 0.4791, cluster 10 sebesar 0.3176, cluster 11 sebesar 0.3721 dengan jumlah keseluruhan sebesar 0.4740. Cluster yang paling baik adalah cluster 7 dengan nilai purity 0.8429, dan cluster yang kurang baik adalah cluster 3 dengan nilai purity 0.2387.

```
[42] purity_score(y,result_kmeans['Cluster'])

Nilai purity keseluruhan adalah 0.4740
Nilai purity masing-masing klaster adalah [0.4987, 0.7911, 0.2387, 0.4208, 0.5031, 0.7387, 0.8429, 0.6875, 0.4791, 0.3176, 0.3721]
```

- Mendapatkan nilai 'entropy_score' pada keseluruhan klaster dan nilai entropy masing-masing klaster

  Berdasarkan nilai Entropy yang dihasilkan maka dapat disimpulkan bahwa cluster K means memiliki kualitas yang kurang baik karena nilai Entropy yang masih jauh dari 0. Dengan demikian, anggota dalam satu cluster memiliki banyak perbedaan.

  Cluster/kelompok 1 sebesar 2.0456, cluster 2 sebesar 0.9195, cluster 3 sebesar 2.8181, cluster 4 sebesar 1.8662, cluster 5 sebesar 2.2967, cluster 6 sebesar 1.2087, cluster 7 sebesar 0.9489, cluster 8 sebesar 1.3405, cluster 9 sebesar 1.9441, cluster 10 sebesar 2.1495, cluster 11 sebesar 2.493 dengan jumlah keseluruhan sebesar 2.0130. Cluster yang paling baik adalah cluster 2 dengan nilai Entropy 0.9195, dan cluster yang paling tidak baik adalah cluster 3 dengan nilai Entropy 2.8181.

```
[41] entropy_score(y,result_kmeans['Cluster'])

Nilai entropy keseluruhan adalah 2.0130
Nilai entropy masing-masing klaster adalah [2.0456, 0.9195, 2.8181, 1.8662, 2.2967, 1.2087, 0.9489, 1.3405, 1.9441, 2.1495, 2.493]
```

3. Evaluasi Model Hirarki (Agglomerative Hierarchical Clustering)
   - Mendapatkan nilai 'entropy_score pada keseluruhan klaster untuk MIN, MAX, AVERAGE

     Nilai entropy dari cluster model hirarki MIN yang terbentuk dalam percobaan ini adalah 2.7365. Nilai entropy dari cluster model hirarki MAX yang terbentuk dalam percobaan ini adalah 2.3901. Nilai entropy dari cluster model hirarki AVERAGE yang terbentuk dalam percobaan ini adalah 2.7336. Berdasarkan nilai Entropy yang dihasilkan maka dapat disimpulkan bahwa model hirarki memiliki kualitas yang kurang baik karena

nilai Entropy masih jauh dari 0. Dengan demikian, anggota dalam satu cluster memiliki banyak perbedaan.

Berikut merupakan *entropy score* setiap cluster dari hirarki MIN, Cluster/kelompok 1 sebesar 2.7394, cluster 2 sebesar 0.0, cluster 3 sebesar 0.0, cluster 4 sebesar 0.0, cluster 5 sebesar 0.0, cluster 6 sebesar 0.0, cluster 7 sebesar 0.0, cluster 8 sebesar 0.0, cluster 9 sebesar 0.0, cluster 10 sebesar 0.0, cluster 11 sebesar 0.0 dengan jumlah keseluruhan sebesar 2.7365. Cluster yang paling baik adalah cluster 2 hingga cluster 11 dengan nilai Entropy 0.0 dan cluster yang paling tidak baik adalah cluster 1 dengan nilai Entropy 2.7394.

Berikut merupakan *entropy score* setiap cluster dari hirarki MAX, Cluster/kelompok 1 sebesar 2.7466, cluster 2 sebesar 1.4822, cluster 3 sebesar 2.0643, cluster 4 sebesar 0.968, cluster 5 sebesar 0.0, cluster 6 sebesar 1.3718, cluster 7 sebesar 0.0, cluster 8 sebesar 0.0, cluster 9 sebesar 1.4641, cluster 10 sebesar 0.0, cluster 11 sebesar 0.0 dengan jumlah keseluruhan sebesar 2.3901. Cluster yang paling baik adalah cluster 7, 8, 10, dan 11 dengan nilai Entropy 0.0 dan cluster yang paling tidak baik adalah cluster 1 dengan nilai Entropy 2.7466.

Berikut merupakan *entropy score* setiap cluster dari hirarki AVG, Cluster/kelompok 1 sebesar 2.7399, cluster 2 sebesar 0.0, cluster 3 sebesar 0.0, cluster 4 sebesar 0.0, cluster 5 sebesar 0.0, cluster 6 sebesar 0.0, cluster 7 sebesar 0.0, cluster 8 sebesar 0.0, cluster 9 sebesar 0.0, cluster 10 sebesar 0.0, cluster 11 sebesar 0.0 dengan jumlah keseluruhan sebesar 2.7336. Cluster yang paling baik adalah cluster 2 hingga cluster 11 dengan nilai Entropy 0.0 dan cluster yang paling tidak baik adalah cluster 1 dengan nilai Entropy 2.7399.

```
[43] print('Clustering Hirarki MIN')
     entropy_score(y,result_agglo_min['Cluster'])
     print("------------------------------------------------------------")
     print('Clustering Hirarki MAX')
     entropy_score(y,result_agglo_max['Cluster'])
     print("------------------------------------------------------------")
     print('Clustering Hirarki AVG')
     entropy_score(y,result_agglo_avg['Cluster'])
```

```
Clustering Hirarki MIN
Nilai entropy keseluruhan adalah 2.7365
Nilai entropy masing-masing klaster adalah [2.7394, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
------------------------------------------------------------
Clustering Hirarki MAX
Nilai entropy keseluruhan adalah 2.3901
Nilai entropy masing-masing klaster adalah [2.7466, 1.4822, 2.0643, 0.968, 0.0, 1.3718, 0.0, 0.0, 1.4641, 0.0, 0.0]
------------------------------------------------------------
Clustering Hirarki AVG
Nilai entropy keseluruhan adalah 2.7336
Nilai entropy masing-masing klaster adalah [2.7399, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

- Mendapatkan nilai ‘purity_score pada keseluruhan klaster untuk MIN, MAX, AVERAGE

  Nilai purity dari cluster model hirarki MIN yang terbentuk dalam percobaan ini adalah 0.2251. Nilai purity dari cluster model hirarki MAX yang terbentuk dalam percobaan ini adalah 0.3611. Nilai purity dari cluster model hirarki AVERAGE yang terbentuk dalam percobaan ini adalah 0.2264. Berdasarkan nilai purity yang dihasilkan maka dapat disimpulkan bahwa model hirarki memiliki kualitas yang kurang baik karena nilai purity mendekati 0. Dengan demikian, anggota dalam satu cluster memiliki banyak kemiripan, tetapi dengan objek pada cluster lain memiliki banyak perbedaan.

Berikut merupakan *purity score* setiap cluster dari hirarki MIN, Cluster/kelompok 1 sebesar 0.2243, cluster 2 sebesar 1.0, cluster 3 sebesar 1.0, cluster 4 sebesar 1.0, cluster 5 sebesar 1.0, cluster 6 sebesar 1.0, cluster 7 sebesar 1.0, cluster 8 sebesar 1.0, cluster 9 sebesar 1.0, cluster 10 sebesar 1.0, cluster 11 sebesar 1.0 dengan jumlah keseluruhan sebesar 0.2251. Cluster yang paling baik adalah cluster 2 hingga cluster 11 dengan nilai purity 1 dan cluster yang paling tidak baik adalah cluster 1 dengan nilai purity 0.2243.

Berikut merupakan *purity score* setiap cluster dari hirarki MAX, Cluster/kelompok 1 sebesar 0.273, cluster 2 sebesar 0.5339, cluster 3 sebesar 0.3405, cluster 4 sebesar 0.8039, cluster 5 sebesar 1.0, cluster 6 sebesar 0.7803, cluster 7 sebesar 1.0, cluster 8 sebesar 1.0, cluster 9 sebesar 0.7048, cluster 10 sebesar 1.0, cluster 11 sebesar 1.0 dengan jumlah keseluruhan sebesar 0.3611. Cluster yang paling baik adalah cluster 5, 7, 8, 10, dan 11 dengan nilai purity 1 dan cluster yang paling tidak baik adalah cluster 1 dengan nilai purity 0.273.

Berikut merupakan *purity score* setiap cluster dari hirarki AVG, Cluster/kelompok 1 sebesar 0.2246, cluster 2 sebesar 1.0, cluster 3 sebesar 1.0, cluster 4 sebesar 1.0, cluster 5 sebesar 1.0, cluster 6 sebesar 1.0, cluster 7 sebesar 1.0, cluster 8 sebesar 1.0, cluster 9 sebesar 1.0, cluster 10 sebesar 1.0, cluster 11 sebesar 1.0 dengan jumlah keseluruhan sebesar 0.2264. Cluster yang paling baik adalah cluster 2 hingga cluster 11 dengan nilai purity 1 dan cluster yang paling tidak baik adalah cluster 1 dengan nilai purity 0.2246.

```
[44] print('Clustering Hirarki MIN')
     purity_score(y,result_agglo_min['Cluster'])
     print("------------------------------------------------------------")
     print('Clustering Hirarki MAX')
     purity_score(y,result_agglo_max['Cluster'])
     print("------------------------------------------------------------")
     print('Clustering Hirarki AVG')
     purity_score(y,result_agglo_avg['Cluster'])
```

```
Clustering Hirarki MIN
Nilai purity keseluruhan adalah 0.2251
Nilai purity masing-masing klaster adalah [0.2243, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
------------------------------------------------------------
Clustering Hirarki MAX
Nilai purity keseluruhan adalah 0.3611
Nilai purity masing-masing klaster adalah [0.273, 0.5339, 0.3405, 0.8039, 1.0, 0.7803, 1.0, 1.0, 0.7048, 1.0, 1.0]
------------------------------------------------------------
Clustering Hirarki AVG
Nilai purity keseluruhan adalah 0.2264
Nilai purity masing-masing klaster adalah [0.2246, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
```

4. Evaluasi Model Berbasis Densitas (DBScan)
   - Mendapatkan nilai ‘entropy_score’ pada keseluruhan klaster

     Berdasarkan nilai Entropy yang dihasilkan maka dapat disimpulkan bahwa semua cluster memiliki kualitas yang kurang baik karena nilai Entropy masih jauh dari 0. Dengan demikian, anggota dalam satu cluster memiliki banyak perbedaan.

     Cluster/kelompok 1 sebesar 2.3199, cluster 2 sebesar 2.4453, cluster 3 sebesar 0.9852, cluster 4 sebesar 1.9056, cluster 5 sebesar 0.9183, cluster 6 sebesar 0.5917, cluster 7 sebesar 0.0, cluster 8 sebesar 1.4242, cluster 9 sebesar 0.971, cluster 10 sebesar 1.0, cluster 11 sebesar 0.9911 dengan jumlah keseluruhan sebesar 2.2490.

Cluster yang paling baik adalah cluster 7 dengan nilai Entropy 0.0, kemudian cluster yang paling tidak baik adalah cluster 2 dengan nilai Entropy 2.4453.

```
[45] entropy_score(y,result_dbscan_1['Cluster'])
     # entropy_score(y,result_dbscan_2['Cluster'])
     # entropy_score(y,result_dbscan_4['Cluster'])

     Nilai entropy keseluruhan adalah 2.2490
     Nilai entropy masing-masing klaster adalah [2.3199, 2.4453, 0.9852, 1.9056, 0.9183, 0.5917, 0.0, 1.4242, 0.971, 1.0, 0.9911]
```

- Mendapatkan nilai 'purity_score pada keseluruhan klaster

  Nilai kemurnian (purity) dari sebuah cluster berkisaran antara 0 dan 1. Clustering buruk jika nilai purity mendekati 0, dan baik jika nilai purity mendekati 1. Nilai kemurnian dari cluster DBscan yang terbentuk dalam percobaan ini adalah 0.3453. Dapat dikatakan hasil dari penelitian ini baik karena nilai purity (kemurnian) mendekati 1.

  Cluster/kelompok 1 sebesar 0.3866, cluster 2 sebesar 0.2766, cluster 3 sebesar 0.5714, cluster 4 sebesar 0.375, cluster 5 sebesar 0.6667, cluster 6 sebesar 0.8571, cluster 7 sebesar 1.0, cluster 8 sebesar 0.6659, cluster 9 sebesar , cluster 10 sebesar 0.5, cluster 11 sebesar 0.5556 dengan jumlah keseluruhan sebesar 0.3453. Cluster yang paling baik adalah cluster 7 dengan nilai purity 1.0 dan cluster yang paling tidak baik adalah cluster 2 dengan nilai purity 0.2766.

```
[46] purity_score(y,result_dbscan_1['Cluster'])

     Nilai purity keseluruhan adalah 0.3453
     Nilai purity masing-masing klaster adalah [0.3866, 0.2766, 0.5714, 0.375, 0.6667, 0.8571, 1.0, 0.6659, 0.6, 0.5, 0.5556]
```

E. Kesimpulan

Berdasarkan hasil evaluasi dari ketiga model clustering yang digunakan yaitu K means, Hirarki (MIN, MAX, AVERAGE) dan DBScan dapat disimpulkan bahwa model K means merupakan model clustering yang paling baik untuk melakukan klasterisasi dari data yang digunakan. Berdasarkan hasil perhitungan nilai entropy dan purity dari setiap klaster pada model K means diperoleh nilai entropy untuk semua klaster sebesar 2,0130 dan nilai purity untuk semua klaster 0,474. Hasil yang klasterisasi yang diperoleh belum merupakan hasil terbaik. Hal ini dikarenakan dataset asli memiliki persebaran yang saling tumpang tindih ketika divisualisasikan berdasarkan atribut 'Kingdom'.

Selain itu hasil skor entropy dan purity pada setiap klaster dalam masing-masing model memiliki hasil yang tidak merata karena ada nilai entropy yang sangat bagus yaitu 0 dan ada yang entropy sangat jelek, begitu pula pada nilai purity. Hal ini terjadi karena beberapa model yang dibangun melakukan klasterisasi secara tidak merata dengan beberapa klaster memiliki lebih banyak titik dari pada klaster lainnya.