CASE BASED - 02

Mata Kuliah Pembelajaran Mesin

KODE DOSEN: GKL

Disusun oleh:

Berlian Muhammad Galin Al Awienoor (1301204378)

KELAS IF-44-10

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
2022/2022

PENDAHULUAN

A. Latar Belakang

Pada era industri 4.0, kemampuan mengolah data dan membuat model sudah seperti kewajiban. Kemajuan teknologi membuat pekerjaan pemrosesan data lebih mudah tetapi dengan kemajuan ini juga muncul banyak istilah baru. Pada perkembangan *Big Data* yang pesat seperti sekarang, kita sudah sering mendengar istilah *Machine Learning* ataupun *Artificial Intelligence*. Secara umum, model *Machine Learning* dapat dibedakan tergantung dari penggunaannya, seperti *Supervised* dan *Unsupervised,* yang merupakan istilah untuk memisahkan model dalam fungsi tertentu. Secara singkat, bisa dikatakan bahwa *Supervised Learning* adalah *machine learning model* yang membutuhkan data target sedangkan *Unsupervised Learning* tidak memerlukan data target.

Unsupervised learning adalah salah satu tipe algoritma *machine learning* yang digunakan untuk menarik kesimpulan dari dataset. Metode ini hanya akan mempelajari suatu data berdasarkan kedekatannya saja atau yang biasa disebut dengan clustering. Metode *unsupervised learning* yang paling umum adalah analisis cluster, yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data.

B. Rumusan Masalah

Pada tugas Case-Based kali ini diberikan file Country-data.csv dengan kriteria seperti dibawah ini:

Clustering the Countries by using Unsupervised Learning for HELP International

Objective:

To categorise the countries using socio-economic and health factors that determine the overall development of the country.

About organization:

HELP International is an international humanitarian NGO that is committed to fighting poverty and providing the people of backward countries with basic amenities and relief during the time of disasters and natural calamities.

Problem Statement:

HELP International have been able to raise around $ 10 million. Now the CEO of the NGO needs to decide how to use this money strategically and effectively. So, CEO has to make decision to choose the countries that are in the direst need of aid. Hence, your Job as a Data scientist is to categorise the countries using some socio-economic and health factors that determine the overall development of the country. Then you need to suggest the countries which the CEO needs to focus on the most.

Dengan data tersebut, kita ditugaskan untuk menyelidiki masalah kualitas data negara di seluruh dunia yang telah diberikan, lalu menjelaskan keputusan mengenai pendekatan pra-pemrosesan data. Menjelajahi kumpulan data dengan meringkas data menggunakan statistik dan mengidentifikasi masalah kualitas data yang relevan. Selain itu, kita juga ditujukan untuk mengkategorikan negara menggunakan faktor sosial-ekonomi dan kesehatan yang menentukan pembangunan negara secara keseluruhan.

C. Tujuan

Tujuan dari disusunnya laporan ini adalah sebagai berikut:

1. Untuk memenuhi tugas mata kuliah Pembelajaran Mesin
2. Untuk menunjukkan dan menjelaskan hasil analisis dengan algoritma *Hierarchical Clustering*
3. Untuk menunjukkan dan menjelaskan hasil akhir yang didapat dari dataset yang diberikan

STRUKTUR LAPORAN

1. Ikhtisar Kumpulan Data

   Data yang saya gunakan adalah Country-data (NIM Genap) dengan kriteria data sebagai berikut:

country	Nama negara sebanyak 167 negara
child-mort	Kematian anak di bawah usia 5 tahun per 1000 kelahiran hidup
exports	Ekspor barang dan jasa per kapita. Diberikan sebagai % dari usia PDB per kapita
health	Total pengeluaran kesehatan per kapita. Diberikan sebagai % dari usia PDB per kapita
imports	Impor barang dan jasa per kapita. Diberikan sebagai % dari usia PDB per kapita
income	Pendapatan bersih per orang
inflation	Ukuran tingkat pertumbuhan tahunan Total PDB
life_expect	Jumlah rata-rata tahun hidup seorang anak yang baru lahir jika pola kematian saat ini tetap sama
total_fer	Jumlah anak yang akan dimiliki setiap wanita jika tingkat kesuburan usia saat ini tetap sama
gdpp	PDB per kapita. Dihitung sebagai Total PDB dibagi dengan total populasi

Baris	167
Kolom	10
Tipe Data	float64 (7), int64 (2), object (1)

Dataset Country-data bertujuan untuk mengelompokkan 167 data (baris) dengan menggunakan beberapa atribut sebanyak 10 data (kolom). Kita diharuskan mengelompokkan negara tersebut dengan mempertimbangkan faktor sosial-ekonomi dan kesehatan yang menentukan pembangunan negara secara keseluruhan. Dengan begitu, kita dapat mengelompokkan beberapa negara tersebut kedalam berbagai Cluster.

2. Ringkasan Pra-Pemrosesan Data

   Pada Pra-Pemrosesan data, saya melakukan berbagai tahap diantaranya eksplorasi data dan visualisasi data, untuk tahapannya sebagai berikut:
   Eksplorasi Data:
   Converting - Display Info - Detecting Missing and Duplicate Value - Indexing - Standardization & Scaling Data
   Visualisasi Data:
   Display Info - Data Distribution - Boxplot Outliers - Correlation

3. Algoritma *Agglomerative Hierarchical Clustering*

   Algoritma yang saya pilih untuk tugas ini adalah algoritma *Agglomerative Hierarchical Clustering.* Karena *Hierarchical Clustering* yang dirasa paling cocok dan mudah untuk kasus ini, dimana *Hierarchical Clustering* itu sendiri merupakan sebuah teknik untuk melakukan pengelompokan data yang dilakukan dengan membuat suatu bagan hirarki (grafik dendrogram) dengan tujuan menunjukkan kemiripan antar data. Setiap data yang mirip akan memiliki hubungan hierarki yang dekat dan membentuk cluster data. Bagan hirarki akan terus terbentuk hingga seluruh data terhubung dalam bagan hirarki tersebut. Cluster dapat dihasilkan dengan memotong bagan hirarki pada level tertentu. Beberapa metode dalam *hierarchical clustering* yaitu *single linkage, complete linkage, average linkage, centroid linkage* dan *ward linkage.*

   Modeling Data:
   Principal Component Analysis - Elbow Method - Cophenet & Silhouette Coefficient - Hierarchical Clustering (Agglomerative)

4. Evaluasi Hasil

   Hasil evaluasi yang didapat adalah dengan keluaran hasil metode Elbow yang menunjukan bahwa cluster terbaik adalah sebanyak 2 cluster. Disamping itu, untuk nilai koefisien Cophenet dan skor Silhouette nya juga menunjukkan bahwa Cluster 2 adalah Cluster terbaik dengan nilai tertinggi dibanding Cluster yang lain. Sedangkan untuk *linkage method* didapat bahwa *Average Linkage* dan *Ward Linkage* menampilkan kualitas dendrogram yang terbaik, jadi saya menggunakan *Ward Linkage*. Terakhir dengan Agglomerative Hierarchical Clustering, hasil analisis menunjukan bahwa terdapat 2 cluster dengan data negara berkembang dan data negara maju. Untuk negara berkembang sebanyak 133 negara yang diperkirakan sebagian membutuhkan bantuan untuk pembangunan negara. Untuk negara maju sebanyak 34 negara yang pembangunan negara sudah baik dan nilai atribut menunjukan hasil positif.

   (Lampiran Screenshot pada bagian IMPLEMENTASI PENGKODEAN)
   (Hasil Analisis dan Evaluasi pada bagian ANALISIS DAN EVALUASI)
   (Source Code https://github.com/rahulacj/Unsupervised-Learning-on-Country-Dataset)

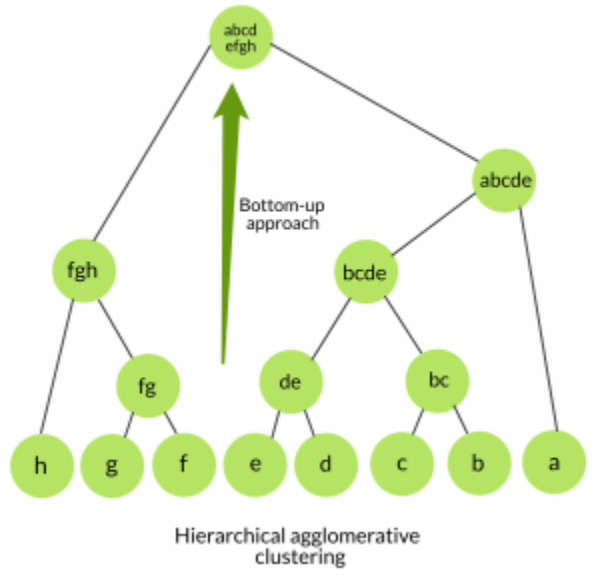
PEMBAHASAN

A. *Agglomerative Hierarchical Clustering*

Metode *Hierarchical Clustering* adalah teknik *clustering* membentuk hirarki atau berdasarkan tingkatan tertentu sehingga menyerupai struktur pohon. Dengan demikian proses pengelompokannya dilakukan secara bertingkat atau bertahap. Biasanya, metode ini digunakan pada data yang jumlahnya tidak terlalu banyak dan jumlah cluster yang akan dibentuk belum diketahui. Di dalam metode hirarki, terdapat dua jenis strategi pengelompokan yaitu *agglomerative* dan *divisive*.

Agglomerative (metode penggabungan) adalah strategi pengelompokan hirarki yang dimulai dengan setiap objek dalam satu cluster yang terpisah kemudian membentuk cluster yang semakin membesar. Jadi, banyaknya cluster awal adalah sama dengan banyaknya objek.

Agglomerative clustering biasa disebut juga sebagai agglomerative nesting dimana cara kerja dalam melakukan pengelompokan data menggunakan ***bottom-up***. Prosesnya dimulai dengan menganggap setiap data sebagai satu cluster kecil (*leaf*) yang hanya memiliki satu anggota saja, lalu pada tahap selanjutnya dua cluster yang memiliki kemiripan akan dikelompokkan menjadi satu cluster yang lebih besar (*nodes*). Proses ini akan dilakukan terus menerus hingga semua data menjadi satu cluster besar (*root*).



Hierarchical agglomerative clustering

1. Algoritma *Agglomerative Hierarchical Clustering*

Dikenal sebagai pendekatan bottom-up atau hirarkis agglomerative clustering (HAC). Sebuah struktur yang lebih informatif daripada kumpulan cluster tidak terstruktur yang dikembalikan oleh clustering datar. Algoritma pengelompokan ini tidak mengharuskan kita untuk menentukan jumlah cluster. Algoritma bottom-up memperlakukan setiap data sebagai cluster tunggal pada awalnya dan kemudian secara berturut-turut menggabungkan pasangan cluster sampai semua cluster digabungkan menjadi satu cluster yang berisi semua data.

Hierarchical Clustering adalah algoritma yang mengelompokkan objek serupa ke dalam kelompok yang disebut cluster. Titik akhir adalah kumpulan cluster, di mana setiap cluster berbeda satu sama lain, dan objek dalam setiap cluster secara umum mirip satu sama lain. Dalam metode ini, setiap titik data dianggap sebagai satu cluster dan cluster ini dikelompokkan untuk membentuk cluster yang lebih besar dan akhirnya cluster tunggal dari semua pengamatan dibuat.

Membangun Model : Untuk menemukan jumlah cluster yang optimal dengan Dendrogram dan Metode Skor Silhouette. Pertama, mencari matriks keterkaitan yang mewakili jarak antara cluster berdasarkan metode keterkaitan yang diberikan.

Memutuskan Metode Linkage Terbaik : Untuk menentukan metode hubungan terbaik dengan menggunakan koefisien cophenet. Koefisien cophenet dengan nilai tertinggi merupakan keterkaitan terbaik.

Jarak Matriks : Saat menggabungkan dua cluster, kita diharuskan memeriksa jarak antara dua setiap pasangan cluster dan menggabungkan pasangan dengan jarak paling sedikit/kemiripan terbanyak. Tetapi pertanyaannya adalah bagaimana jarak itu ditentukan. Ada berbagai cara untuk menentukan jarak/kesamaan Cluster yang disebut Linkage Method , yaitu:

1. *Single Linkage*

Pengukuran jarak antar cluster dilakukan dengan mengukur terlebih dahulu jarak antar tiap observasi dari cluster yang berbeda. Jarak paling kecil (*minimum distance*) akan menjadi ukuran antar cluster. Dendrogram akan terbentuk dari cluster-cluster yang memiliki *(dis)similarity* paling kecil. Hal ini membuat dendrogram yang terbentuk menjadi lebih "*loose*" atau berdekatan antar clusternya.

$$d_{12} = \min_{ij} d(X_i, Y_j)$$

2. *Complete Linkage*

   Pengukuran jarak antar cluster dilakukan dengan mengukur terlebih dahulu jarak antar tiap observasi dari cluster yang berbeda. Jarak paling tinggi (*maximum distance*) akan menjadi ukuran antar cluster. Kemudian, dendrogram akan terbentuk dari cluster-cluster yang memiliki *(dis)similarity* paling kecil. Hal ini membuat dendrogram yang terbentuk menjadi lebih terpisah antar clusternya (terbentuk cluster yang "*compact*").

$$d_{12} = \max_{ij} d(X_i, Y_j)$$

3. *Average Linkage*

   Pengukuran jarak antar cluster dilakukan dengan mengukur terlebih dahulu jarak antar tiap observasi dari cluster yang berbeda. Kemudian, dihitung rata-rata jarak dari pairwise distance tersebut dan nilai tersebut akan menjadi ukuran antar cluster. Dendrogram akan terbentuk dari cluster-cluster yang memiliki *(dis)similarity* paling kecil. Umumnya metode ini akan menghasilkan cluster yang tidak terlalu "*loose*" maupun "*compact*".

$$d_{12} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{l} d(X_i, Y_j)$$

4. Ward Linkage

   Pada metode ini, di setiap iterasinya akan dibentuk cluster-cluster yang kemudian dihitung nilai ***within sum of square*** tiap cluster (WSS). WSS dapat diartikan sebagai jumlah dari jarak tiap observasi ke nilai tengah cluster. Cluster-cluster yang menghasilkan ***within sum of square*** terkecil akan diambil kemudian digabungkan hingga membentuk satu dendrogram utuh.

5. Centroid Linkage

   Perhitungan *(dis)similarity* atau jarak antar cluster dilakukan dengan mengukur jarak antar centroid pada dua cluster. Perhitungan centroid disini menggunakan rata-rata pada suatu variabel x. Dendrogram yang terbentuk akan berdasarkan cluster dengan jarak antar centroid paling kecil.

$$d_{12} = d(\bar{X}, \bar{Y})$$

2. Kelebihan

Terdapat kelebihan dari penggunaan algoritma A*gglomerative Hierarchical Clustering,* berikut adalah kelebihannya.

a) Mampu menggambarkan kedekatan antar data dengan dendrogram.
b) Cukup mudah untuk pembuatannya.
c) Dapat menentukan banyak cluster yang terbentuk setelah dendrogram terbentuk.
d) Tidak memiliki fungsi objektif alami yang sedang dioptimalkan (berbeda dengan K-Means)
e) Monotonisitas:
   - *Dissimilarity* antara sepasang kluster yang digabungkan pada titik mana pun dalam algoritma selalu setidaknya sebesar perbedaan pasangan kluster yang digabungkan pada langkah sebelumnya
   - Hanya untuk *Single-Link, Complete-Link,* dan *Average-Link*

3. Kekurangan

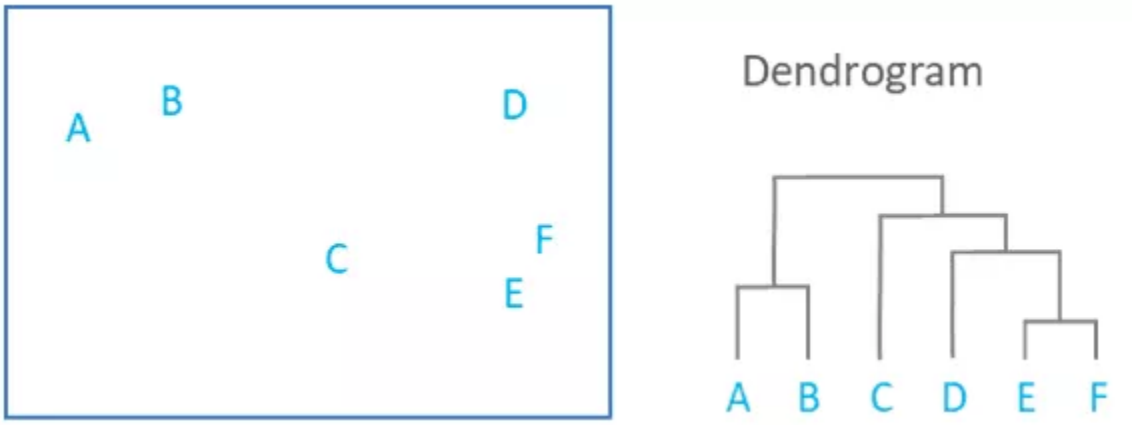
Sementara itu, terdapat juga kekurangan dari algoritma *Agglomerative Hierarchical Clustering.*

a) Tidak dapat menganalisis data kategorik secara langsung (terhambat pada penghitungan jarak yang hanya bisa dilakukan untuk data numerik, sehingga data kategorik perlu dipre-process terlebih dahulu).
b) Tidak diperuntukkan untuk menghasilkan jumlah cluster optimal yang mutlak (jumlah cluster dapat berubah-ubah tergantung level pemotongan dendrogram).
c) Sensitif terhadap data yang memiliki skala berbeda (sehingga data perlu dinormalisasi/standarisasi terlebih dahulu).
d) Sensitif terhadap *outlier.*
e) Cukup berat komputasinya untuk data berukuran besar.
f) Kompleksitas Ruang : $O(N^2)$
g) Kompleksitas Waktu : $O(N^3)$

4. Cara Kerja Algoritma *Agglomerative Hierarchical Clustering*

Secara umum, berikut adalah langkah-langkah yang dilakukan untuk melakukan *hierarchical cluster analysis*, menggabungkan konsep-konsep yang sudah kita pahami sebelumnya:

1. Menyiapkan data dimana data yang digunakan adalah data bertipe numerik agar dapat digunakan untuk penghitungan jarak.
2. Menghitung *(dis)similarity* atau jarak antar data yang berpasangan pada dataset. Metode penghitungan *(dis)similarity* dapat dipilih berdasarkan data. Nilai *(dis)similarity* tersebut kemudian akan disusun menjadi *distance matrix*.
3. Membuat dendrogram dari *distance matrix* menggunakan *linkage method* tertentu. Kita juga dapat mencoba beberapa *linkage method* kemudian memilih dedrogram paling baik.
4. Menentukan dimana akan melakukan pemotongan tree (dengan nilai *(dis)similarity* tertentu). Disinilah tahap dimana cluster akan terbentuk.
5. Melakukan interpretasi dari dendrogram yang telah didapat.

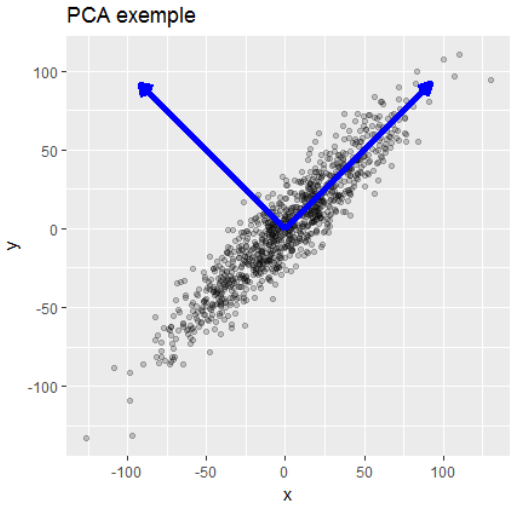


5. *Principal Component Analysis*

Principal Component Analysis (PCA) adalah suatu teknik analisis statistik multivariat. Bisa dibilang, inilah teknik analisis statistik yang paling populer sekarang. Biasanya, PCA digunakan dalam bidang pengenalan pola serta pemrosesan sinyal. PCA pada dasarnya merupakan dasar dari analisis data multivariat yang menerapkan metode proyeksi. Teknik analisis ini biasanya digunakan untuk meringkas tabel data multivariat dalam skala besar hingga bisa dijadikan kumpulan variabel yang lebih kecil atau indeks ringkasan. Dari situ, kemudian variabel dianalisis untuk mengetahui tren tertentu, cluster variabel, hingga outlier.

Principal Component Analysis adalah salah satu metode reduksi dimensi pada *Machine Learning*. PCA akan memilih variabel-variabel yang mampu menjelaskan sebagian besar variabilitas data. Metode PCA digunakan jika data yang

ada memiliki jumlah variabel yang besar dan memiliki korelasi antar variabelnya. Perhitungan dari principal component analysis didasarkan pada perhitungan nilai eigen dan vektor eigen yang menyatakan penyebaran data dari suatu dataset.



6. *Elbow Method*

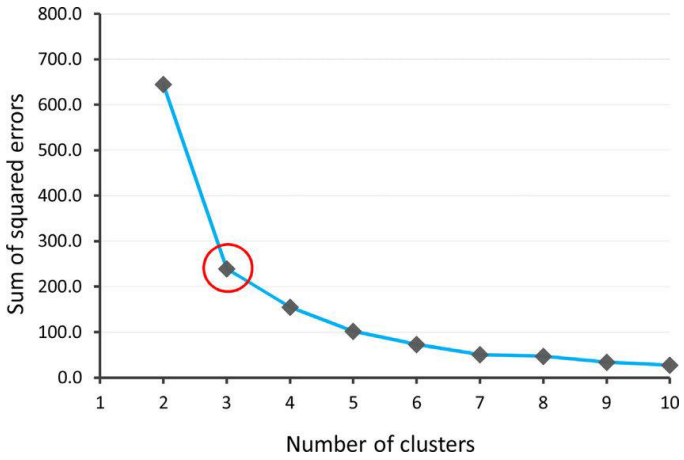
Elbow method adalah metode yang sering dipakai untuk menentukan jumlah cluster yang akan digunakan pada k-means dan hierarchical clustering. Clustering adalah meminimumkan jarak antara data point dan centroid, serta memaksimumkan jarak antara centroid yang dihitung.

Kriteria Elbow Method sebagai berikut:

- Semakin banyak cluster, *Sum-Square-Error* semakin rendah
- Memilih jumlah minimum cluster

Namun, ketika SSE mulai naik level

- Gunakan Cross Validation dan rata-rata SSE setiap fold



7. Skor *Silhouette*

   Pengujian model dilakukan untuk mengetahui seberapa dekat relasi antara objek dalam sebuah cluster dan seberapa jauh sebuah cluster terpisah dengan cluster lain. Metode pengujian yang akan digunakan adalah Silhouette Coefficient. Metode silhouette coefficient merupakan gabungan dari dua metode yaitu metode *cohesion* yang berfungsi untuk mengukur seberapa dekat relasi antara objek dalam sebuah cluster, dan metode *separation* yang berfungsi untuk mengukur seberapa jauh sebuah cluster terpisah dengan cluster lain.

8. Koefisien *Cophenet*

   Dalam statistik dan ilmu data, korelasi cophenetic (lebih tepatnya, koefisien korelasi *cophenetic*) adalah ukuran seberapa kuat dendrogram mempertahankan jarak berpasangan antara titik data asli yang tidak dimodelkan. Meskipun telah paling banyak diterapkan di bidang biostatistik (biasanya untuk menilai model sekuens DNA berbasis cluster, atau model taksonomi lainnya), ini juga dapat digunakan di bidang penyelidikan lain di mana data mentah cenderung terjadi dalam rumpun, atau cluster. Koefisien ini juga telah diusulkan untuk digunakan sebagai tes untuk cluster bersarang.

9. *Country Dataset*

   Objektif:

   Untuk mengkategorikan negara menggunakan faktor sosial-ekonomi dan kesehatan yang menentukan pembangunan negara secara keseluruhan.

   Problem:

   HELP International telah berhasil mengumpulkan sekitar $10 juta. Sekarang CEO LSM perlu memutuskan bagaimana menggunakan uang ini secara strategis dan efektif. Jadi, CEO harus membuat keputusan untuk memilih negara-negara yang paling membutuhkan bantuan. Karenanya, Pekerjaan kita harus mengkategorikan negara menggunakan beberapa faktor sosial ekonomi dan kesehatan yang menentukan perkembangan negara secara keseluruhan. Kemudian Anda perlu menyarankan negara-negara yang paling perlu menjadi fokus CEO.

country	Nama negara sebanyak 167 negara
child-mort	Kematian anak di bawah usia 5 tahun per 1000 kelahiran hidup
exports	Ekspor barang dan jasa per kapita. Diberikan sebagai % dari usia PDB per kapita
health	Total pengeluaran kesehatan per kapita. Diberikan sebagai % dari usia

	PDB per kapita
imports	Impor barang dan jasa per kapita. Diberikan sebagai % dari usia PDB per kapita
income	Pendapatan bersih per orang
inflation	Ukuran tingkat pertumbuhan tahunan Total PDB
life_expect	Jumlah rata-rata tahun hidup seorang anak yang baru lahir jika pola kematian saat ini tetap sama
total_fer	Jumlah anak yang akan dimiliki setiap wanita jika tingkat kesuburan usia saat ini tetap sama
gdpp	PDB per kapita. Dihitung sebagai Total PDB dibagi dengan total populasi

IMPLEMENTASI PENGKODEAN ALGORITMA

Untuk melakukan analisis dan evaluasi algoritma *Agglomerative Hierarchical Clustering,* berikut adalah tahapan implementasi untuk proses algoritma A*gglomerative Hierarchical Clustering* yang telah saya kerjakan sedemikian rupa dari tahap awal (*pre-processing*) sampai tahap akhir (analisis) dengan menggunakan bahasa pemrograman Python.

1. *Import Library*

```
[1]  #import library
     import pandas as pd
     import matplotlib.pyplot as plt
     import numpy as np

     #import pack untuk ignoring warning
     from warnings import filterwarnings
     filterwarnings('ignore')
```

2. *Import Dataset*

```
[2]  #import dataset
     #uploading file dataset

     from google.colab import files
     uploaded = files.upload()
```

Choose Files Country-data.csv

- **Country-data.csv**(text/csv) - 9229 bytes, last modified: 6/17/2020 - 100% done

```
Saving Country-data.csv to Country-data.csv
```

3. *Converting Dataset*

```
[3] #converting dataframe

    dataset = pd.read_csv("Country-data.csv")
    dataset.head()
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200

```
[4] dataset.tail()
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
162	Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.50	2970
163	Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.47	13500
164	Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.95	1310
165	Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.67	1310
166	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460

4. *Describe Data*

```
[42] dataset.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
child_mort	167.0	38.270060	40.328931	2.6000	8.250	19.30	62.10	208.00
exports	167.0	41.108976	27.412010	0.1090	23.800	35.00	51.35	200.00
health	167.0	6.815689	2.746837	1.8100	4.920	6.32	8.60	17.90
imports	167.0	46.890215	24.209589	0.0659	30.200	43.30	58.75	174.00
income	167.0	17144.688623	19278.067698	609.0000	3355.000	9960.00	22800.00	125000.00
inflation	167.0	7.781832	10.570704	-4.2100	1.810	5.39	10.75	104.00
life_expec	167.0	70.555689	8.893172	32.1000	65.300	73.10	76.80	82.80
total_fer	167.0	2.947964	1.513848	1.1500	1.795	2.41	3.88	7.49
gdpp	167.0	12964.155689	18328.704809	231.0000	1330.000	4660.00	14050.00	105000.00

```
[6]  dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   country     167 non-null    object 
 1   child_mort  167 non-null    float64
 2   exports     167 non-null    float64
 3   health      167 non-null    float64
 4   imports     167 non-null    float64
 5   income      167 non-null    int64  
 6   inflation   167 non-null    float64
 7   life_expec  167 non-null    float64
 8   total_fer   167 non-null    float64
 9   gdpp        167 non-null    int64  
dtypes: float64(7), int64(2), object(1)
memory usage: 13.2+ KB
```

```
[7]  dataset.shape
```

```
(167, 10)
```

5. *Detecting Missing Value*

```
[8]  #mendeteksi missing value

     dataset.isnull().sum()
```

```
country       0
child_mort    0
exports       0
health        0
imports       0
income        0
inflation     0
life_expec    0
total_fer     0
gdpp          0
dtype: int64
```

6. *Detecting Duplicate Value*

```
[9]  #mengecek duplikasi data

     dataset.duplicated().sum()
```

```
0
```

7. *Indexing* kolom Country (Object menjadi Index)

```
[54] #INDEXING
     #mengkonversi kolom country dari object menjadi index

     dataset['country'].nunique()
     dataset.set_index('country', inplace=True)
     dataset
```

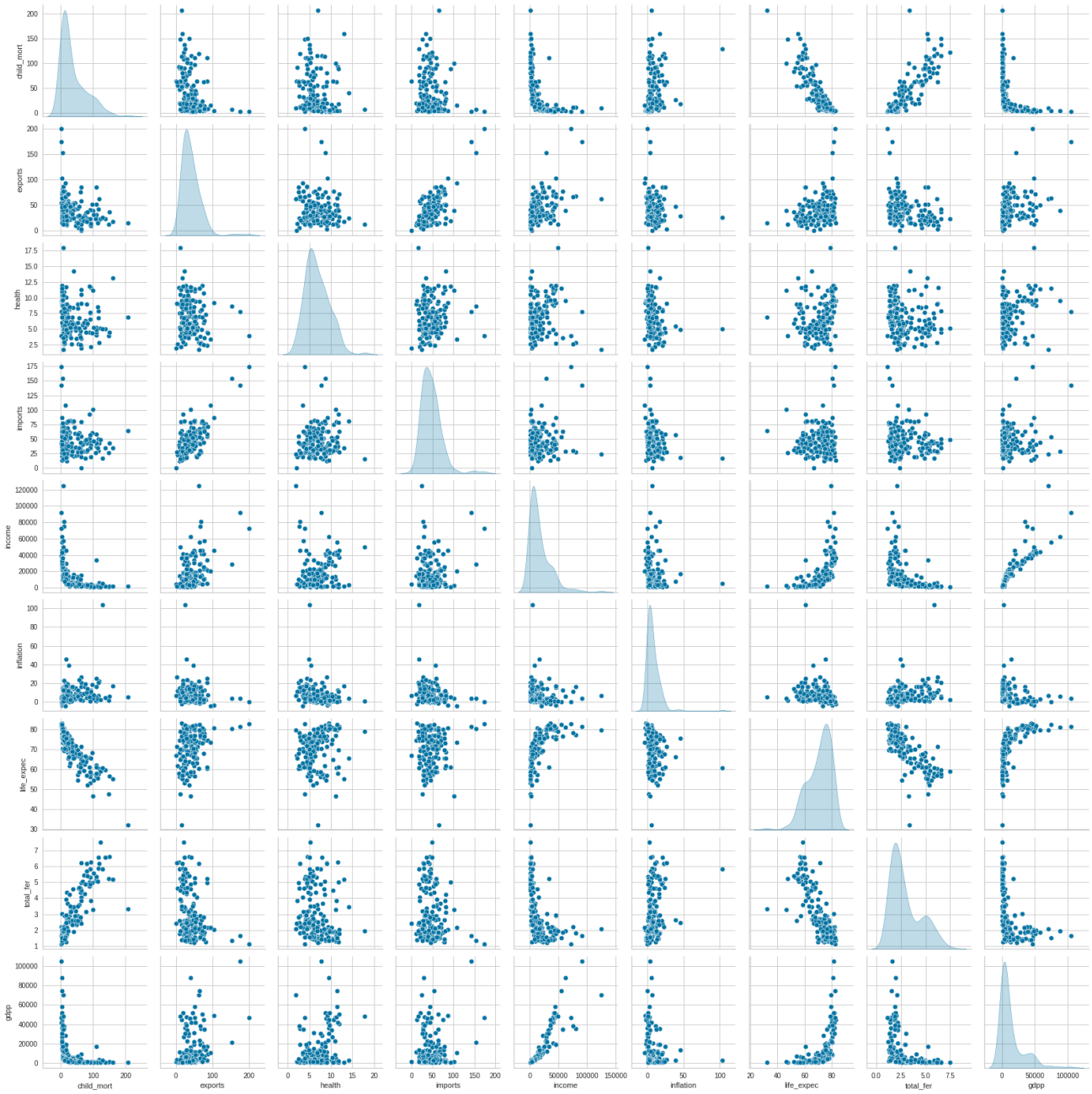
country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200
...	...	...	...	...	...	...	...	...	...
Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.50	2970
Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.47	13500
Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.95	1310
Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.67	1310
Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460

167 rows × 9 columns

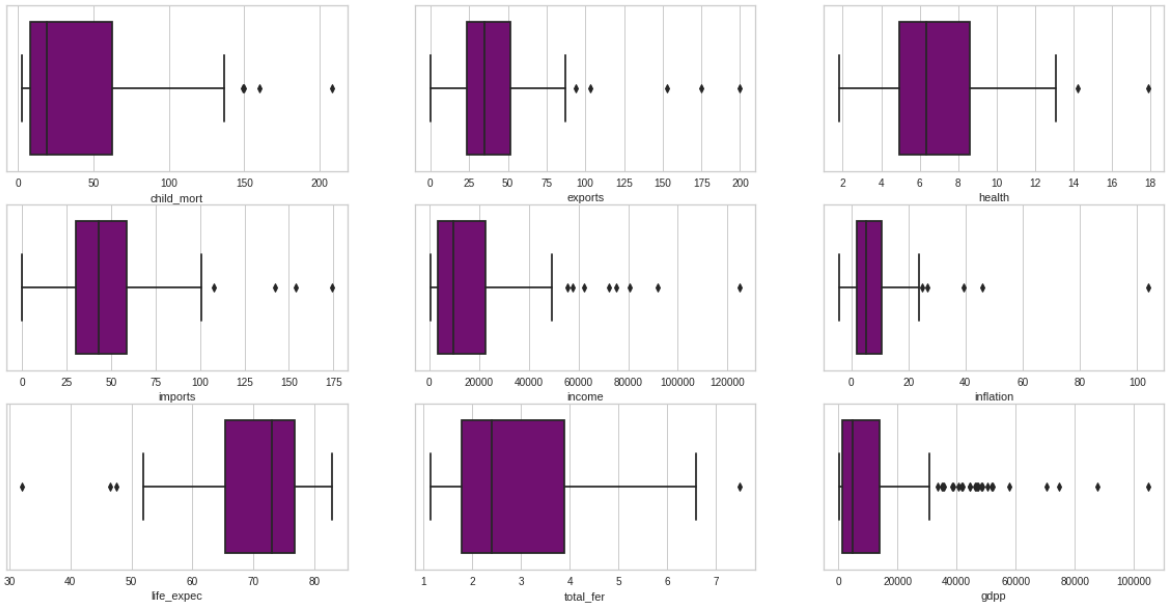
8. Visualisasi Data

```
[55] # import seaborn
     import seaborn as sns

     # import matplotlib
     import matplotlib.pyplot as plt
     import matplotlib.cm as cm

     # import plotly untuk grafik
     import plotly
     import plotly.express as px
```

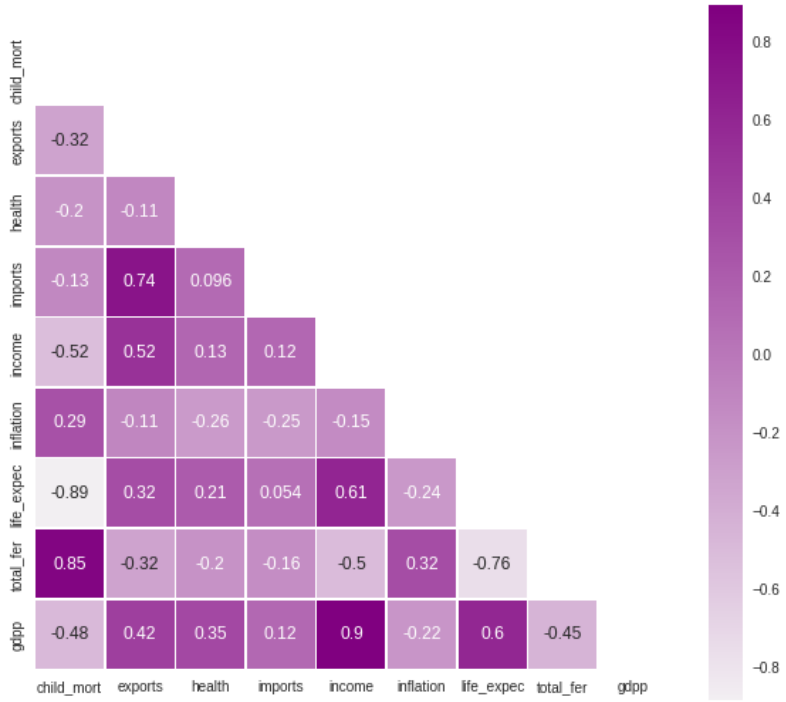
```
[56] #visualisasi

     sns.pairplot(data=dataset,diag_kind='kde')
     plt.show()
```




```
[57] #boxplot data pencilan

     fig, ax = plt.subplots(nrows = 3, ncols = 3, figsize=(20, 10))

     for variable, subplot in zip(dataset.columns, ax.flatten()):
           sns.boxplot(dataset[variable], ax = subplot, color='purple')
     plt.show()
```




```
[58] #korelasi data

     corr = dataset.corr()
     mask = np.triu(np.ones_like(corr, dtype=np.bool))
     f, ax = plt.subplots(figsize=(10, 10))
     cmap = sns.light_palette('purple', as_cmap=True)
     sns.heatmap(corr, mask=mask, cmap=cmap, vmax=None, center=0, square=True, annot=True,
                 linewidths=.5, cbar_kws={"shrink": .9})
     plt.show()
```



```
[59] dataset.corr()
```

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
child_mort	1.000000	-0.318093	-0.200402	-0.127211	-0.524315	0.288276	-0.886676	0.848478	-0.483032
exports	-0.318093	1.000000	-0.114408	0.737381	0.516784	-0.107294	0.316313	-0.320011	0.418725
health	-0.200402	-0.114408	1.000000	0.095717	0.129579	-0.255376	0.210692	-0.196674	0.345966
imports	-0.127211	0.737381	0.095717	1.000000	0.122406	-0.246994	0.054391	-0.159048	0.115498
income	-0.524315	0.516784	0.129579	0.122406	1.000000	-0.147756	0.611962	-0.501840	0.895571
inflation	0.288276	-0.107294	-0.255376	-0.246994	-0.147756	1.000000	-0.239705	0.316921	-0.221631
life_expec	-0.886676	0.316313	0.210692	0.054391	0.611962	-0.239705	1.000000	-0.760875	0.600089
total_fer	0.848478	-0.320011	-0.196674	-0.159048	-0.501840	0.316921	-0.760875	1.000000	-0.454910
gdpp	-0.483032	0.418725	0.345966	0.115498	0.895571	-0.221631	0.600089	-0.454910	1.000000

9. Standarisasi Data *(Standar Scaler)*

```
[60] #import untuk preprosesing
     from sklearn.preprocessing import StandardScaler

     #standarisasi

     ss = StandardScaler()

     scaled_df = ss.fit_transform(dataset)
     X = pd.DataFrame(scaled_df, columns=dataset.columns, index = dataset.index)

     X.head()
```

country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
Afghanistan	1.291532	-1.138280	0.279088	-0.082455	-0.808245	0.157336	-1.619092	1.902882	-0.679180
Albania	-0.538949	-0.479658	-0.097016	0.070837	-0.375369	-0.312347	0.647866	-0.859973	-0.485623
Algeria	-0.272833	-0.099122	-0.966073	-0.641762	-0.220844	0.789274	0.670423	-0.038404	-0.465376
Angola	2.007808	0.775381	-1.448071	-0.165315	-0.585043	1.387054	-1.179234	2.128151	-0.516268
Antigua and Barbuda	-0.695634	0.160668	-0.286894	0.497568	0.101732	-0.601749	0.704258	-0.541946	-0.041817

10. PCA

```
[61] from sklearn.decomposition import PCA

     pca = PCA()
     pcdata = pca.fit_transform(X)
```

```
[62] pca_df = pd.DataFrame(pcdata)
     pca_df.head()
```

	0	1	2	3	4	5	6	7	8
0	-2.913025	0.095621	-0.718118	1.005255	-0.158310	-0.254597	0.383000	0.415076	-0.014148
1	0.429911	-0.588156	-0.333486	-1.161059	0.174677	0.084579	0.248919	-0.221042	0.173316
2	-0.285225	-0.455174	1.221505	-0.868115	0.156475	-0.401696	-0.087214	-0.184162	0.084037
3	-2.932423	1.695555	1.525044	0.839625	-0.273209	-0.547996	-0.440835	-0.355998	-0.091339
4	1.033576	0.136659	-0.225721	-0.847063	-0.193007	-0.206919	0.241978	-0.023681	0.094270

```
[63] y = np.cumsum(pca.explained_variance_ratio_)
     y
```

```
array([0.4595174 , 0.63133365, 0.76137624, 0.87190786, 0.94530998,
       0.97015232, 0.98275663, 0.99256944, 1.        ])
```

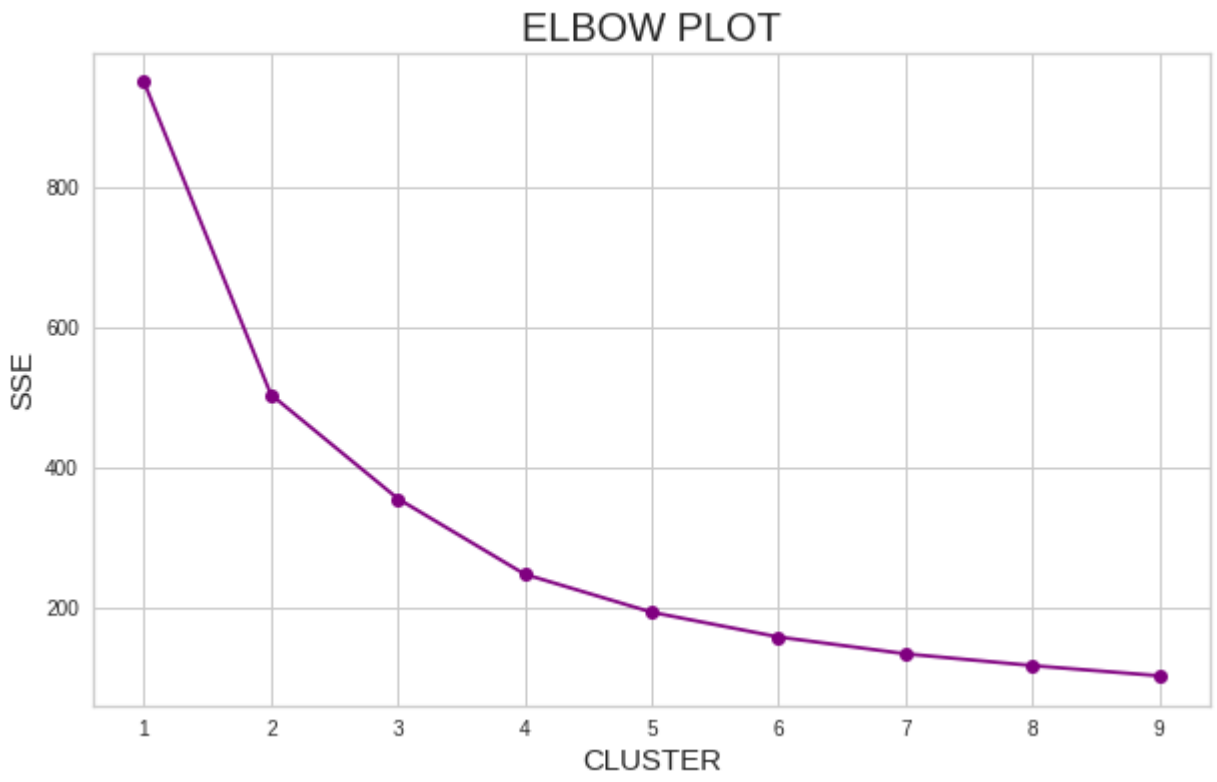
```
[64] mypca = PCA(n_components=2)
     pca5 = mypca.fit_transform(X)
     pca5_df = pd.DataFrame(pca5, index=dataset.index)
     pca5_df
```

country	0	1
Afghanistan	-2.913025	0.095621
Albania	0.429911	-0.588156
Algeria	-0.285225	-0.455174
Angola	-2.932423	1.695555
Antigua and Barbuda	1.033576	0.136659
...	...	...
Vanuatu	-0.820631	0.639570
Venezuela	-0.551036	-1.233886
Vietnam	0.498524	1.390744
Yemen	-1.887451	-0.109453
Zambia	-2.864064	0.485998

167 rows × 2 columns

11. *Elbow Method*

```
[65] #ELBOW PLOT

     from sklearn.cluster import KMeans

     elbow = []

     for i in range(1,10):
         kmeans = KMeans(n_clusters = i,random_state = 10)
         kmeans.fit(pca5_df)
         elbow.append(kmeans.inertia_)

     plt.figure(figsize=(10,6))
     plt.plot(range(1,10), elbow, marker="o", color="purple")
     plt.title('ELBOW PLOT', fontsize = 20)
     plt.xlabel('CLUSTER', fontsize = 15)
     plt.ylabel('SSE', fontsize = 15)
     plt.grid(True)
     plt.show()
```



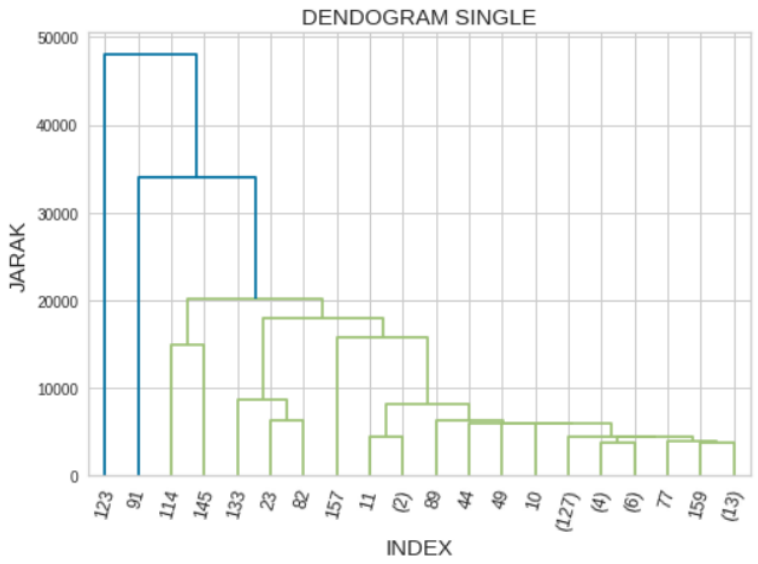
12. Clustering

```
[66] #import library untuk clustering
     from sklearn.cluster import AgglomerativeClustering
     from sklearn.metrics import silhouette_score,silhouette_samples,accuracy_score
     from scipy.cluster.hierarchy import  dendrogram, cophenet, linkage
     from scipy.spatial.distance import pdist

     #mencari keterkaitan data dengan 5 metode

     methods = ['single','complete','average','ward','centroid']

     for i in methods:
         link = linkage(X, method=i)
         coeff, cophenet_dist = cophenet(link, pdist(X))
         print('Koefisien Cophenet untuk', i, ':', coeff)
```

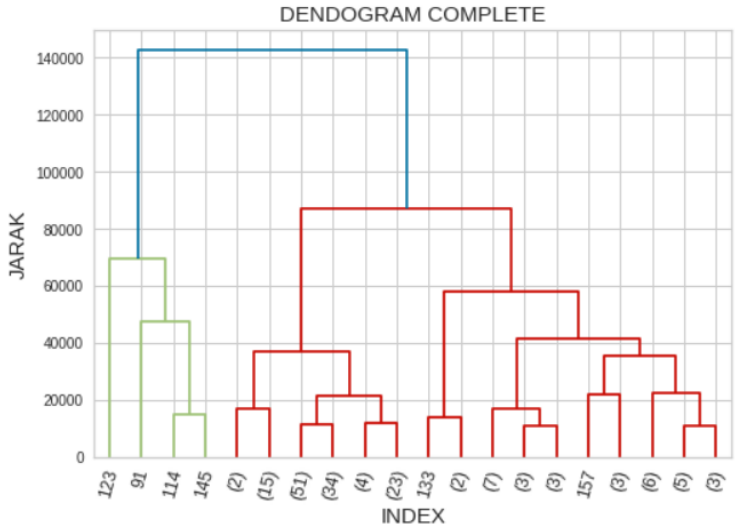
```
Koefisien Cophenet untuk single : 0.7604287846523685
Koefisien Cophenet untuk complete : 0.490896504155209
Koefisien Cophenet untuk average : 0.8394248289254103
Koefisien Cophenet untuk ward : 0.52902912158488
Koefisien Cophenet untuk centroid : 0.831962758604513
```

13. Dendrogram

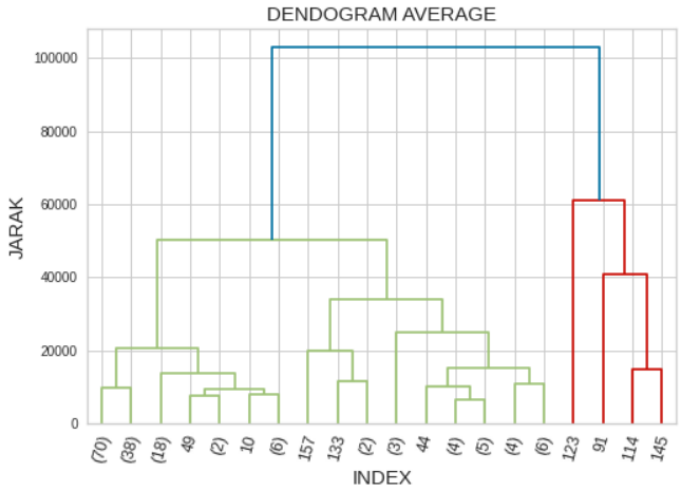
```
[67] #dendogram single linkage (keterkaitan single)

     plt.title('DENDOGRAM SINGLE', fontsize = 15)
     plt.xlabel('INDEX', fontsize = 15)
     plt.ylabel('JARAK', fontsize = 15)

     link=linkage(dataset, method='single')
     dendrogram(link, leaf_rotation=75., truncate_mode='lastp', p=20)

     plt.show()
```



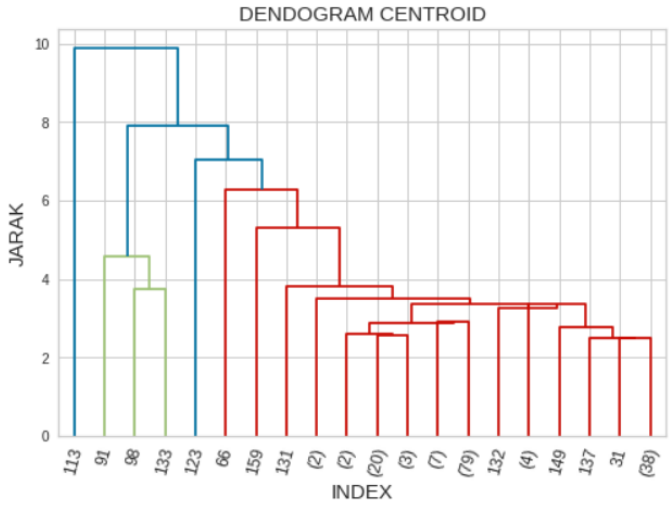
```
[68] #dendogram complete linkage (keterkaitan complete)

     plt.title('DENDOGRAM COMPLETE', fontsize = 15)
     plt.xlabel('INDEX', fontsize = 15)
     plt.ylabel('JARAK', fontsize = 15)

     link=linkage(dataset, method='complete')
     dendrogram(link, leaf_rotation=75., truncate_mode='lastp', p=20)

     plt.show()
```



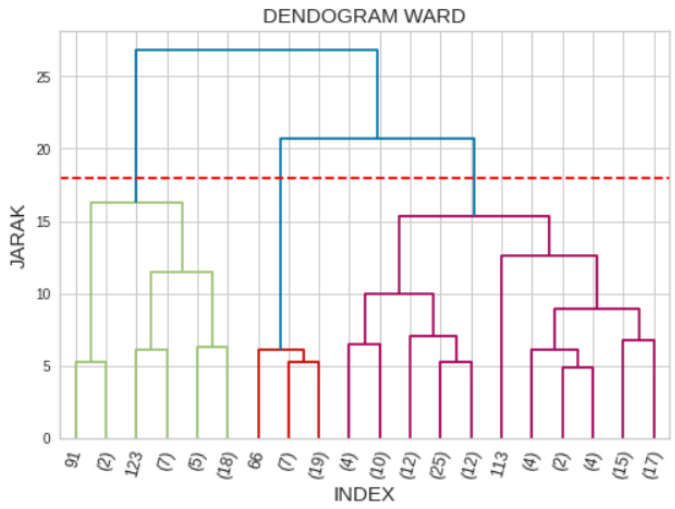

```
[69] #dendogram average linkage (keterkaitan rata-rata)

     plt.title('DENDOGRAM AVERAGE', fontsize = 15)
     plt.xlabel('INDEX', fontsize = 15)
     plt.ylabel('JARAK', fontsize = 15)

     link=linkage(dataset, method='average')
     dendrogram(link, leaf_rotation=75., truncate_mode='lastp', p=20)

     plt.show()
```



[70]
```
#dendogram centroid linkage (keterkaitan centroid)

plt.title('DENDOGRAM CENTROID', fontsize = 15)
plt.xlabel('INDEX', fontsize = 15)
plt.ylabel('JARAK', fontsize = 15)

link=linkage(X, method='centroid')
dendrogram(link, leaf_rotation=75., truncate_mode='lastp', p=20)

plt.axhline(y=18, c='red', ls='--')
plt.show()
```



[71]
```
#dendogram ward linkage (keterkaitan ward)

plt.title('DENDOGRAM WARD', fontsize = 15)
plt.xlabel('INDEX', fontsize = 15)
plt.ylabel('JARAK', fontsize = 15)

link=linkage(X, method='ward')
dendrogram(link, leaf_rotation=75., truncate_mode='lastp', p=20)

plt.axhline(y=18, c='red', ls='--')
plt.show()
```



14. Skor *Silhouette*

```
[72] #mencari skor silhouette

     n_cluster = [2,3,4,5]

     for K in n_cluster:
         siluet  = AgglomerativeClustering(n_clusters=K)
         predict = siluet.fit_predict(X)
         s_score = silhouette_score(dataset,predict,random_state=10)
         print("Untuk Cluster {} skor silhouettenya : {}".format(K, s_score) )
```

```
Untuk Cluster 2 skor silhouettenya : 0.7040968796654424
Untuk Cluster 3 skor silhouettenya : 0.17256555742114882
Untuk Cluster 4 skor silhouettenya : 0.1503417991741312
Untuk Cluster 5 skor silhouettenya : 0.03926320584377816
```

15. Modelling dengan *Agglomerative Hierarchical Clustering*

```
[73] #membangun model dengan k = 2
     agglo_clust = AgglomerativeClustering(n_clusters=2, linkage='ward')
     agglo_clust.fit(X)

     set(agglo_clust.labels_)
```

```
{0, 1}
```

```
[74] country_agglo_df = dataset.copy()
     country_agglo_df['clusters'] = agglo_clust.labels_
     country_agglo_df
```

country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp	clusters
Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553	0
Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090	0
Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460	0
Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530	0
Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200	0
...	...	...	...	...	...	...	...	...	...	...
Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.50	2970	0
Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.47	13500	0
Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.95	1310	0
Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.67	1310	0
Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460	0

ANALISIS DAN EVALUASI

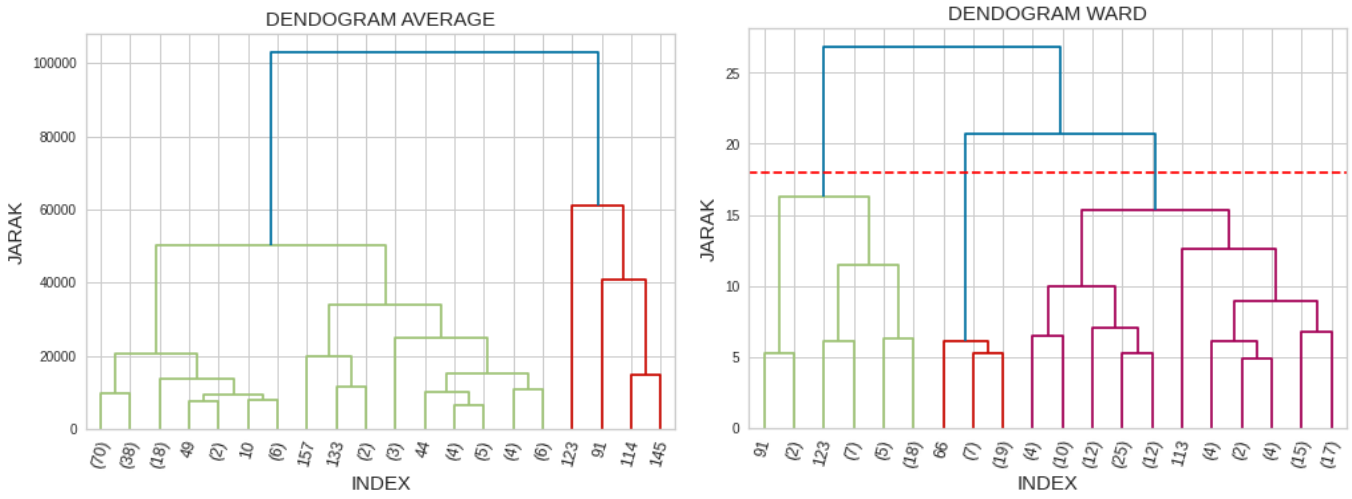
A. Analisis dan Evaluasi

Berikut merupakan hasil analisis dan evaluasi dari proses implementasi algoritma *Agglomerative Hierarchical Clustering* yang telah saya kerjakan.

- Nilai dari Koefisien *Cophenet*

```
Koefisien Cophenet untuk single : 0.7604287846523685
Koefisien Cophenet untuk complete : 0.490896504155209
Koefisien Cophenet untuk average : 0.8394248289254103
Koefisien Cophenet untuk ward : 0.52902912158488
Koefisien Cophenet untuk centroid : 0.831962758604513
```

Dari hasil nilai Koefisien *Cophenet* diatas, menunjukan bahwa dengan menggunakan metode *Average-Linkage* akan menghasilkan hasil yang terbaik. Namun jika dilihat dari output dendrogram, kualitas dari *Ward-Linkage* lebih baik dan lebih jelas dibandingkan dengan *Average-Linkage.* Berikut perbandingan dendrogram dari *Average-Linkage* dan *Ward-Linkage*.



Dengan perbandingan kualitas dendrogram diatas, saya memutuskan untuk menggunakan *Ward-Linkage* untuk tahap selanjutnya.

- Skor dari *Silhouette Coefficient*

```
Untuk Cluster 2 skor silhouettenya : 0.7040968796654424
Untuk Cluster 3 skor silhouettenya : 0.17256555742114882
Untuk Cluster 4 skor silhouettenya : 0.1503417991741312
Untuk Cluster 5 skor silhouettenya : 0.039263205843778l6
```

Jika dilihat dari output skor *silhouette* diatas, ditunjukkan bahwa jika jumlah Cluster sebanyak 2 yang terbaik dengan skor silhouettenya sebesar 0,704. Skor tersebut sangat tinggi dan berbeda sangat signifikan jika dibandingkan dengan skor silhouette diatas 2 cluster (3, 4, dan 5). Jadi, dengan jumlah Cluster sebanyak 2 selanjutnya akan dibuat modeling dengan algoritma *Agglomerative Hierarchical Clustering*.

- *Modeling* dengan 2 Cluster

```
[73] #membangun model dengan k = 2
     agglo_clust = AgglomerativeClustering(n_clusters=2, linkage='ward')
     agglo_clust.fit(X)

     set(agglo_clust.labels_)
```

```
{0, 1}
```

```
[75] country_agglo_df['clusters'].value_counts()
```

```
0    133
1     34
Name: clusters, dtype: int64
```

```
[76] #analisis hierarchial clustering

     country_agglo_df.groupby(["clusters"]).mean()
```

clusters	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	46.529323	36.660895	6.384812	46.375683	9362.127820	8.71909	68.145865	3.218797	5242.210526
1	5.961765	58.508824	8.501176	48.902941	47588.235294	4.11550	79.982353	1.888529	43170.588235

Dijelaskan bahwa kita dapat mengelompokkan 167 negara kedalam 2 kelompok (cluster), untuk cluster 1 sebanyak 133 negara dan cluster 2 sebanyak 34 negara.

- List Negara Berkembang (cluster = 0)

```
[77] # List negara berkembang
     n_berkembang=country_agglo_df[country_agglo_df['clusters']==0]
     n_berkembang.index
```

```
Index(['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Antigua and Barbuda',
       'Argentina', 'Armenia', 'Azerbaijan', 'Bahamas', 'Bangladesh',
       ...
       'Turkmenistan', 'Uganda', 'Ukraine', 'Uruguay', 'Uzbekistan', 'Vanuatu',
       'Venezuela', 'Vietnam', 'Yemen', 'Zambia'],
      dtype='object', name='country', length=133)
```

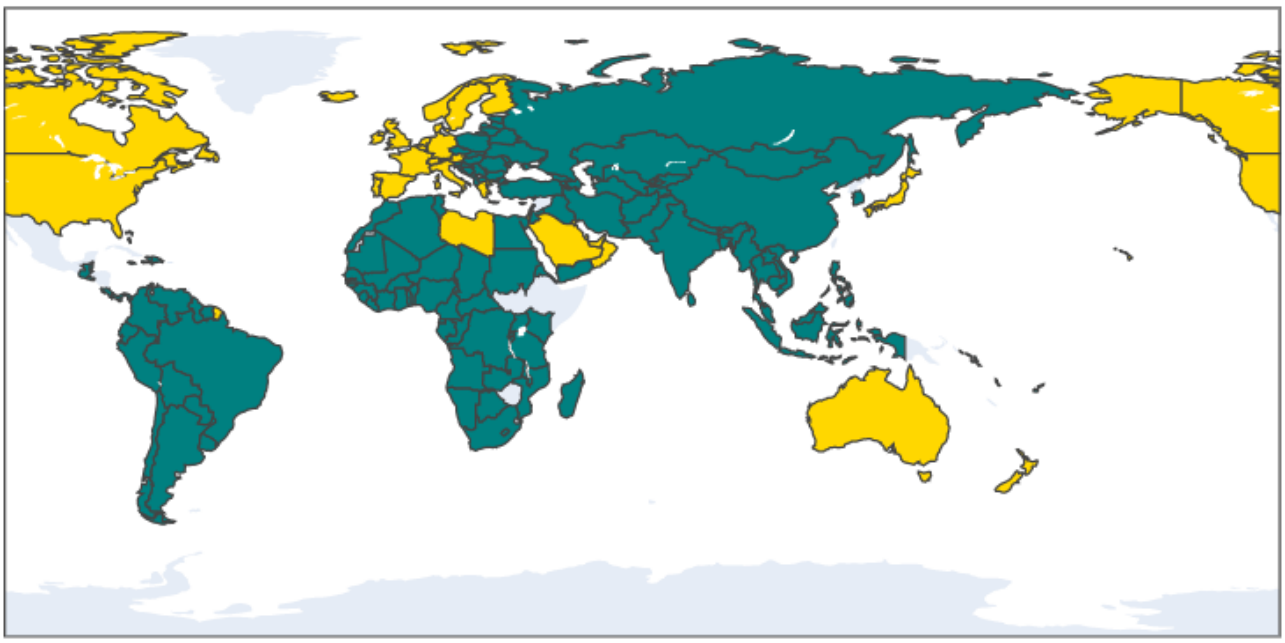
- List Negara Maju (cluster = 1)

```
[78] # List negara maju
     n_maju=country_agglo_df[country_agglo_df['clusters']==1]
     n_maju.index
```

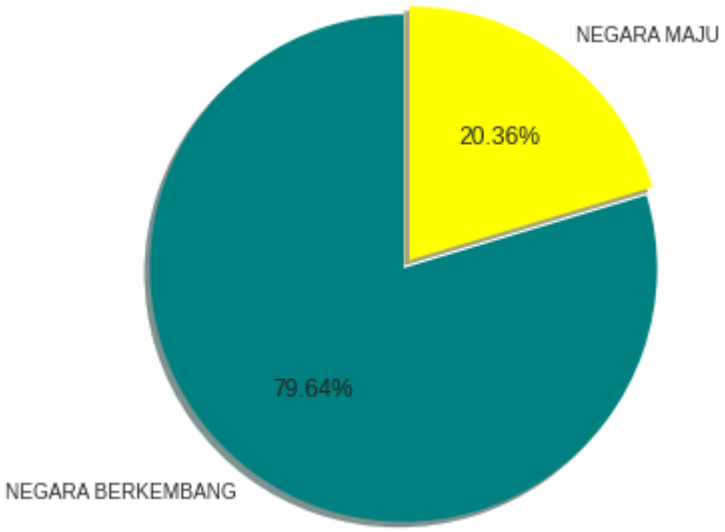
```
Index(['Australia', 'Austria', 'Bahrain', 'Belgium', 'Brunei', 'Canada',
       'Denmark', 'Finland', 'France', 'Germany', 'Greece', 'Iceland',
       'Ireland', 'Israel', 'Italy', 'Japan', 'Kuwait', 'Libya', 'Luxembourg',
       'Malta', 'Netherlands', 'New Zealand', 'Norway', 'Oman', 'Portugal',
       'Qatar', 'Saudi Arabia', 'Singapore', 'Spain', 'Sweden', 'Switzerland',
       'United Arab Emirates', 'United Kingdom', 'United States'],
      dtype='object', name='country')
```

- Visualisasi untuk Hasil Analisis

```
[79] from plotly.offline import iplot
     import plotly.graph_objects as go


     fig_cases = go.Figure(data = go.Choropleth(locations = country_agglo_df.index,
                                        z = country_agglo_df['clusters'],
                                        locationmode = 'country names',
                                        colorscale = [[0, 'teal'],[0.5, 'green'],[1, 'gold']],
                                        colorbar = {'title':'CLUSTER'},
                                        colorbar_title = "CLUSTER"))

     fig_cases.update_layout(
                           title_text='GRAFIK GEOGRAFIS UNTUK 167 NEGARA DI DUNIA',
                           geo = dict(showframe = True,showcoastlines = False)
                           )

     iplot(fig_cases)
```



```
[80] plt.figure(figsize=(5,5))
     plt.pie(country_agglo_df.clusters.value_counts(), colors=['teal', 'yellow'], explode=(0.04, 0),
     autopct='%1.2f%%', shadow=True, startangle=90, labels=['NEGARA BERKEMBANG','NEGARA MAJU'])

     plt.axis('equal')
     plt.show()
```



Jadi, dengan keluaran output diatas, saya dapat menganalisa bahwa dengan penerapan *Ward-Linkage* yang sebelumnya didapat dari nilai koefisien *Cophenet*, untuk menentukan jarak cluster dapat memberikan hasil yang terbaik (jelas). Selanjutnya dengan pengimplimentasian algoritma untuk menentukan skor *Silhouette* didapat bahwa dengan jumlah 2 Cluster adalah jumlah yang terbaik dibandingkan dengan jumlah cluster diatas 2. Perbedaan tersebut dihasilkan sangat signifikan yakni skornya sebesar 0,70.

Dengan hasil data diatas, saya dapat membangun model dengan penerapan algoritma *agglomerative hierarchical clustering.* Pada proses *modelling* saya telah berhasil mengelompokkan 167 negara dengan 2 kelompok (cluster), untuk cluster 1 sebanyak 133 negara (negara berkembang) dan cluster 2 sebanyak 34 negara (negara maju).

Ada 133 negara di Cluster 0 berwarna hijau (ditandai dengan menunjukkan nilai rata-rata menuju negatif untuk semua fitur jika dibandingkan dengan Cluster 1) yang berlokasi hampir di seluruh Amerika Selatan, sebagian Afrika, Eropa, dan Asia. Negara berkembang menunjukan hasil pembangunan negara yang kurang baik dan beberapa negara menunjukan hasil yang buruk (negatif), jadi bantuan internasional perlu memberikan perhatian khusus untuk beberapa negara berkembang yang dirasa sangat membutuhkan bantuan.

Ada 34 negara di Cluster 1 berwarna kuning (ditandai dengan menunjukkan nilai-nilai positif seperti pembangunan ekonomi yang baik, harapan hidup yang tinggi, angka kematian anak yang rendah) terletak di Amerika Utara, Australia, Eropa dan beberapa di Asia. Negara maju menunjukan hasil pembangunan negara yang sangat baik, jadi untuk bantuan internasional tidak perlu memberikan perhatian khusus untuk negara maju.

PENUTUP

A. Kesimpulan

Unsupervised learning adalah salah satu tipe algoritma *machine learning* yang digunakan untuk menarik kesimpulan dari dataset. Metode ini hanya akan mempelajari suatu data berdasarkan kedekatannya saja atau yang biasa disebut dengan *clustering*. Metode *unsupervised learning* yang paling umum adalah analisis cluster, yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data.

Hierarchical Clustering adalah algoritma yang mengelompokkan objek serupa ke dalam kelompok yang disebut cluster. Titik akhir adalah kumpulan cluster, di mana setiap cluster berbeda satu sama lain, dan objek dalam setiap cluster secara umum mirip satu sama lain. Dalam metode ini, setiap titik data dianggap sebagai satu cluster dan cluster ini dikelompokkan untuk membentuk cluster yang lebih besar dan akhirnya cluster tunggal dari semua pengamatan dibuat.

Pada proses membangun model, untuk menemukan jumlah cluster yang optimal dengan Dendrogram dan Metode Skor Silhouette. Pertama, mencari matriks keterkaitan yang mewakili jarak antara cluster berdasarkan metode keterkaitan yang diberikan. Ada beberapa metode linkage seperti *single, complete, average, centroid,* dan *ward*. Selanjutnya proses memutuskan metode tautan mana yang terbaik, untuk menentukan metode hubungan terbaik dengan menggunakan koefisien *cophenet*.

Dengan penjelasan kasus diatas, saya dapat menyimpulkan bahwa dengan penerapan algoritma agglomerative hierarchical clustering, dapat memberikan keluaran dengan baik apabila dilakukan preprocessing terlebih dahulu sebagai tahap awal sebelum masuk ke dalam tahapan modeling. Dengan menggunakan metode tahapan linkage serta skor Silhouette yang terbaik akan dihasilkan output berupa pengelompokkan clustering yang terbaik.

B. Daftar Pustaka

Unsupervised Learning on Country Data. (2020, June 17). Kaggle. https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data

https://github.com/rahulacj/Unsupervised-Learning-on-Country-Dataset

GeeksforGeeks. (2022, September 19). ML | Hierarchical clustering (Agglomerative and Divisive clustering). https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/

Murtagh, F., & Contreras, P. (2011). Algorithms for hierarchical clustering: an overview. WIREs Data Mining and Knowledge Discovery, 2(1), 86–97. https://doi.org/10.1002/widm.53

Slide Perkuliahan Telkom University Mata Kuliah Pembelajaran Mesin, 08 - Hierarchical Clustering

C. Lampiran

1. Link GitHub: https://github.com/berlianm/Case-Based02-ML
2. Link Hasil Pengerjaan Pemrograman: https://colab.research.google.com/drive/1NY288AKCkQtB2tmk-niPM8UR-FeXDCJ9?usp=sharing
3. Link Dokumen Laporan: https://docs.google.com/document/d/10VLu_VF3S4kjrWInbXJt-F1-S3GLgjQofk6bl0sbITA/edit?usp=sharing
4. Link Dokumen Presentasi: https://docs.google.com/presentation/d/1WKcd8Qxp7jYPB_KnH-W1f1cBXqCgWCXviXl9VZ7IMN8/edit?usp=sharing
5. Link Video Presentasi: https://drive.google.com/file/d/1Z25S9EwT4vCMiByE8LMIyz_ZslHjDMx5/view?usp=sharing
6. Link Google Drive: https://drive.google.com/drive/folders/1fcZto5_YA-G3JFuXKrVCe_8QQhiKJguj