HIERARCHICAL CLUSTERING
PRAKTIKUM PEMBELAJARAN MESIN

Disusun Oleh:

Nashirudin Baqiy 24060119130045

Lab A2

DEPARTEMEN ILMU KOMPUTER / INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

2021

BAB I PENDAHULUAN

1.1. Rumusan Masalah

1.1.1. Lakukan agglomerative clustering untuk dataset random yang tersedia di atas dengan single linkage dan average linkage! Jelaskan perbedaannya!

1.1.2. Lakukan agglomerative clustering menggunakan scipy dan scikit-learn dengan single linkage dan average linkage untuk dataset cars_clustering! Jelaskan perbedaannya!

1.1.3. Lakukan agglomerative clustering menggunakan scipy dan scikit-learn dengan single linkage, average linkage, dan complete linkage untuk dataset iris! Jelaskan perbedaannya!

1.2. Tujuan

1.2.1. Melakukan agglomerative clustering dengan dataset random

1.2.2. Melakukan agglomerative clustering dengan dataset cars

1.2.3. Melakukan agglomerative clustering dengan dataset bunga iris

1.3. Dasar Teori

Hierarchical methods 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. Sedangkan Divisive (metode pembagian) adalah strategi pengelompokan hirarki yang dimulai dari semua objek dikelompokkan menjadi cluster tunggal kemudian dipisah sampai setiap objek berada dalam cluster yang terpisah. Pada praktikum ini kita hanya akan focus pada metode agglomerative.

Terdapat tiga teknik pengelompokan yang paling dikenal dalam Agglomerative Method, yaitu:

a. Single linkage (jarak terdekat atau tautan tunggal)

Teknik yang menggabungkan cluster-cluster menurut jarak antara anggota-anggota terdekat di antara dua cluster.

b. Average linkage (jarak rata-rata atau tautan rata-rata)

Teknik yang menggabungkan cluster-cluster menurut jarak rata-rata pasangan anggota masing-masing pada himpunan antara dua cluster.

c. Complete linkage (jarak terjauh atau tautan lengkap)

Teknik yang menggabungkan cluster-cluster menurut jarak antara anggota-anggota terjauh di antara dua cluster.

BAB II PEMBAHASAN

1. Agglomerative dengan data random

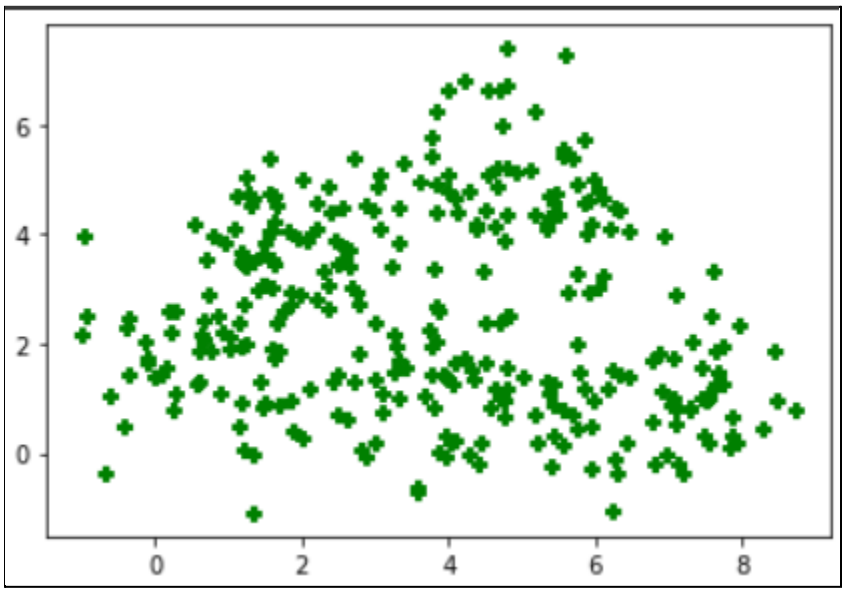
a. Data

Langkah pertama yang dilakukan adalah membentuk data random dengan menggunakan make_blobs

```
X1, y1 = make_blobs(n_samples=313, centers=[[1, 2], [2, 4], [5, 5], [7, 1], [4, 1]], cluster_std=0.9)
```

b. Plotting Persebaran Data

Langkah selanjutnya adalah melakukan plotting persebaran data untuk melihat gambaran umum dari data yang telah dibuat secara random sebelumnya

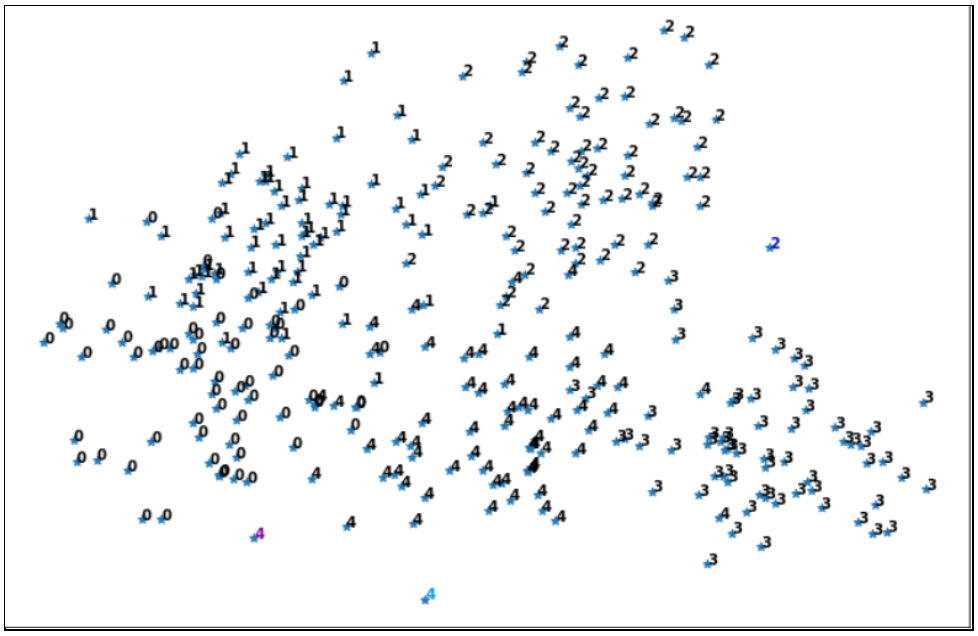


c. Agglomerative Clustering

Clustering pertama yang akan dilakukan adalah clustering dengan single linkage.

```
agglom = AgglomerativeClustering(n_clusters=4, linkage= 'single')
agglom.fit(X1, y1)
```

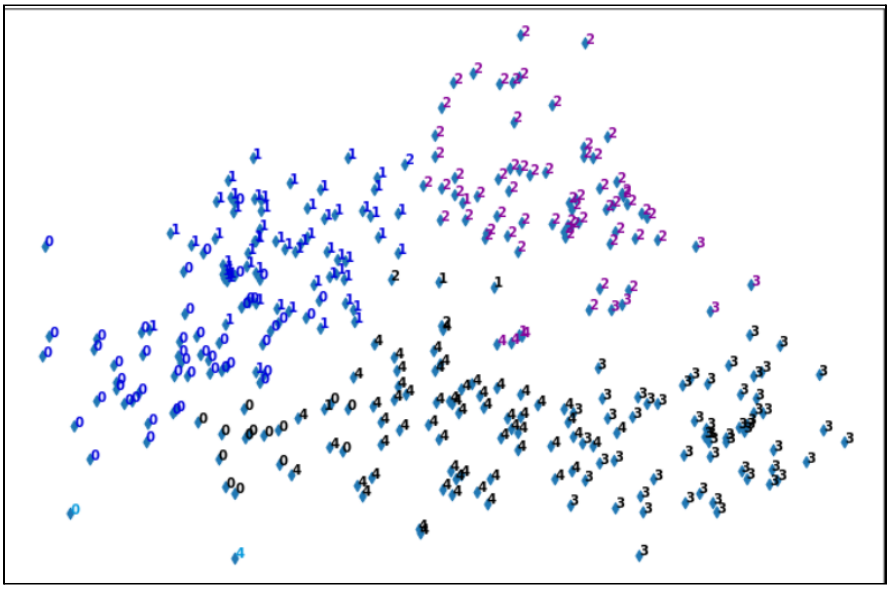
Yang akan menghasilkan pengelompokkan data dengan plotting seperti gambar di bawah ini.



Yang kedua, yaitu clustering dengan average linkage.

```
agglom2 = AgglomerativeClustering(n_clusters=4, linkage= 'average')
agglom2.fit(X1, y1)
```

Dengan hasil seperti pada gambar berikut.



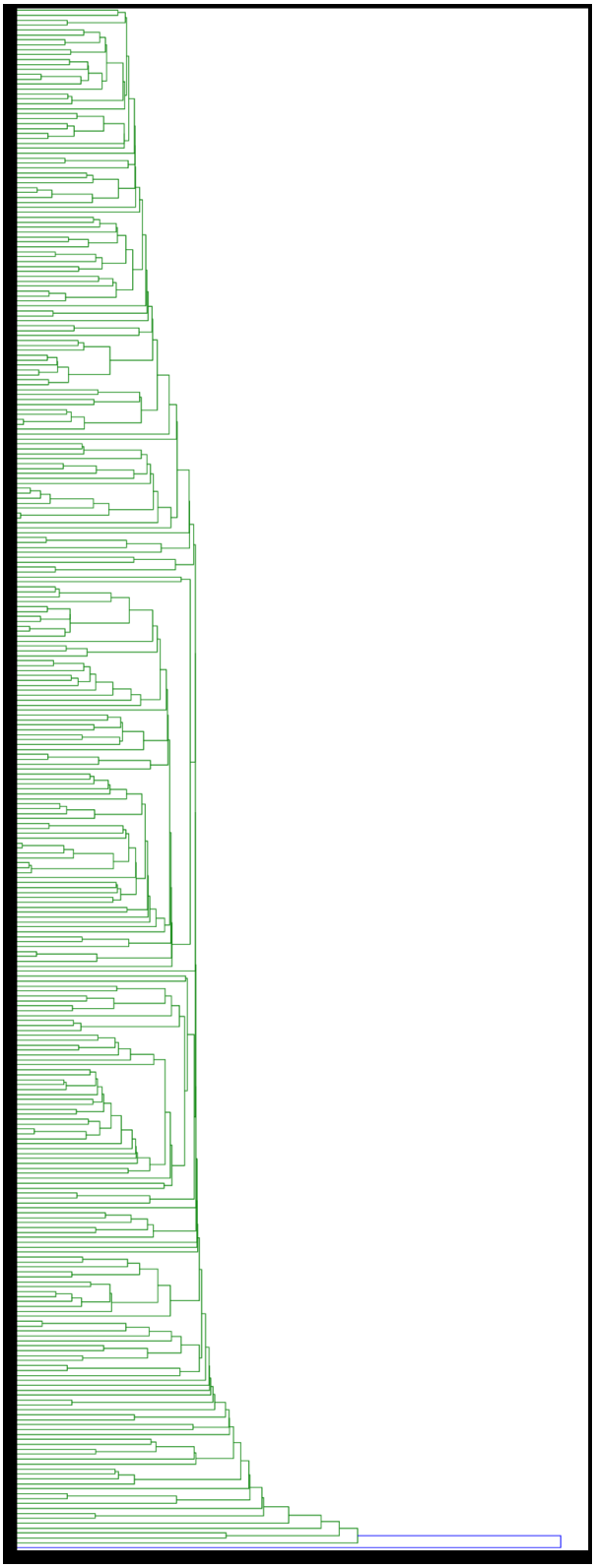
d. Plotting Dendrogram

Sebelum melakukan plotting dendogram, tahap yang diperlukan adalah membentuk distance matrix yang menggambarkan jarak antar data yang satu dengan data yang lain.

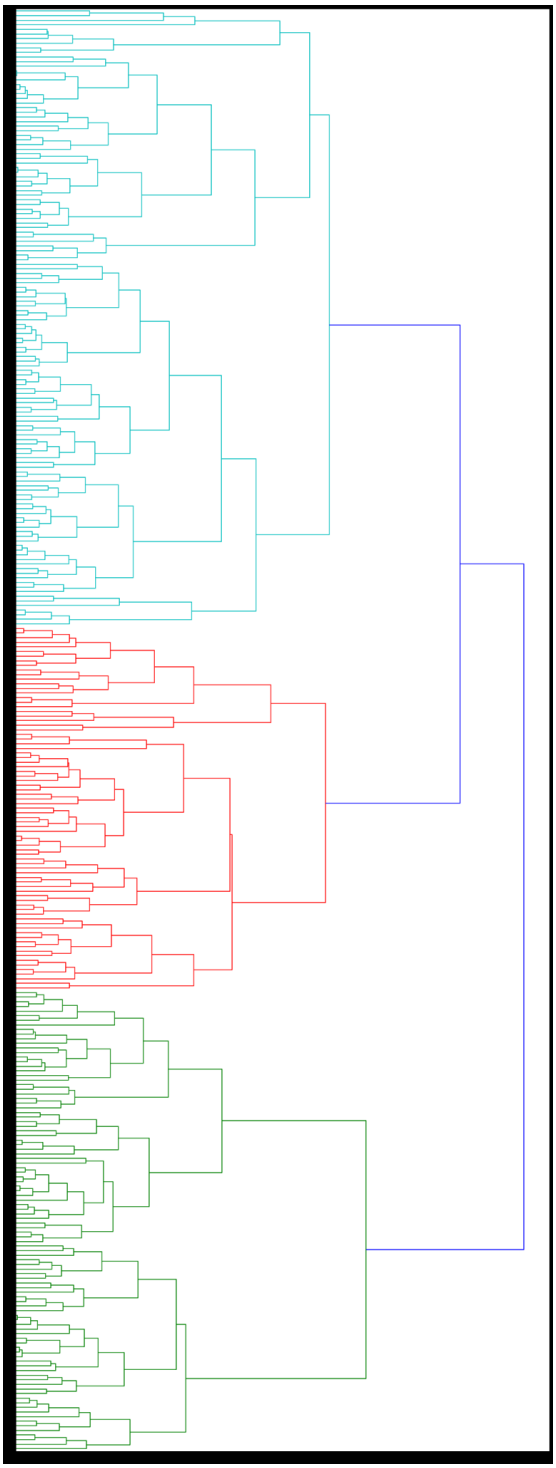
```
dist_matrix = distance_matrix(X1, X1)
print(dist_matrix)
```

```
[[0.         0.65061467 0.71732927 ... 0.6974852  0.14094819 0.66107255]
 [0.65061467 0.         0.27122401 ... 0.16446775 0.65509477 0.1927772 ]
 [0.71732927 0.27122401 0.         ... 0.10969549 0.77217581 0.08504608]
 ...
 [0.6974852  0.16446775 0.10969549 ... 0.         0.73300785 0.05464704]
 [0.14094819 0.65509477 0.77217581 ... 0.73300785 0.         0.70512543]
 [0.66107255 0.1927772  0.08504608 ... 0.05464704 0.70512543 0.        ]]
```

Hasil single linkage



Hasil average linkage



2. Agglomerative dengan data cars

a. Data

Dataset yang digunakan untuk soal nomor 2 ini adalah data cars yang berisi spesifikasi dari mobil-mobil.

```
BEFORE:  2544
AFTER:  1872
```

	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition
112	Volkswagen	Golf	9.761	11.425	0.0	14.90	2.0	115.0	98.9	68.3	163.3	2.767	14.5	26.0	2.278	0
113	Volkswagen	Jetta	83.721	13.240	0.0	16.70	2.0	115.0	98.9	68.3	172.3	2.853	14.5	26.0	4.427	0
114	Volkswagen	Passat	51.102	16.725	0.0	21.20	1.8	150.0	106.4	68.5	184.1	3.043	16.4	27.0	3.934	0
115	Volkswagen	Cabrio	9.569	16.575	0.0	19.99	2.0	115.0	97.4	66.7	160.4	3.079	13.7	26.0	2.259	0
116	Volkswagen	GTI	5.596	13.760	0.0	17.50	2.0	115.0	98.9	68.3	163.3	2.762	14.6	26.0	1.722	0

b. Feature selection

Tidak semua fitur akan digunak di praktikum ini hanya beberapa kolom saja. Maka dari itu perlu dilakukan feature selection yang digunakan untuk memilih kolom-kolom yang akan dijadikan fitur dan membuang kolom yang lain.

```
featureset = pdf[['engine_s', 'horsepow', 'wheelbas', 'width', 'length', 'curb_wgt', 'fuel_cap', 'mpg']]
```

c. Normalization

Selanjutnya adalah melakukan normalisasi dari data yang ada. Meskipun jika dilihat, nilai data yang ada tidak terlalu jauh perbedaanya normalisasi tetaplah dilakukan untuk menghindari kemungkinan data yang *overvalue*.

```
from sklearn.preprocessing import MinMaxScaler

x = featureset.values
min_max_scaler = MinMaxScaler()

feature_mtx = min_max_scaler.fit_transform(x)
feature_mtx[0:5]
```

```
array([[0.11428571, 0.21518987, 0.18655098, 0.28143713, 0.30625832,
        0.2310559 , 0.13364055, 0.43333333],
       [0.31428571, 0.43037975, 0.3362256 , 0.46107784, 0.5792277 ,
        0.50372671, 0.31797235, 0.33333333],
       [0.35714286, 0.39240506, 0.47722343, 0.52694611, 0.62849534,
        0.60714286, 0.35483871, 0.23333333],
       [0.11428571, 0.24050633, 0.21691974, 0.33532934, 0.38082557,
        0.34254658, 0.28110599, 0.4       ],
       [0.25714286, 0.36708861, 0.34924078, 0.80838323, 0.56724368,
        0.5173913 , 0.37788018, 0.23333333]])
```

d. Distance Matrix

Dalam agglomerative clustering, pada setiap iterasi, algoritma harus memperbarui matriks jarak untuk mencerminkan jarak cluster yang baru terbentuk dengan cluster yang tersisa di hutan. Metode berikut ini didukung di Scipy untuk menghitung jarak antara cluster yang baru terbentuk.

```
length = feature_mtx.shape[0]
D = scipy.zeros([length, length])

for i in range(length):
    for j in range(length):
        D[i, j] = scipy.spatial.distance.euclidean(feature_mtx[i], feature_mtx[j])

print(D)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: DeprecationWarning: scipy.zeros is d

[[0.         0.57777143 0.75455727 ... 0.28530295 0.24917241 0.18879995]
 [0.57777143 0.         0.22798938 ... 0.36087756 0.66346677 0.62201282]
 [0.75455727 0.22798938 0.         ... 0.51727787 0.81786095 0.77930119]
 ...
 [0.28530295 0.36087756 0.51727787 ... 0.         0.41797928 0.35720492]
 [0.24917241 0.66346677 0.81786095 ... 0.41797928 0.         0.15212198]
 [0.18879995 0.62201282 0.77930119 ... 0.35720492 0.15212198 0.        ]]
```

```
dist_matrix = distance_matrix(feature_mtx, feature_mtx)
print(dist_matrix)
```

```
[[0.         0.57777143 0.75455727 ... 0.28530295 0.24917241 0.18879995]
 [0.57777143 0.         0.22798938 ... 0.36087756 0.66346677 0.62201282]
 [0.75455727 0.22798938 0.         ... 0.51727787 0.81786095 0.77930119]
 ...
 [0.28530295 0.36087756 0.51727787 ... 0.         0.41797928 0.35720492]
 [0.24917241 0.66346677 0.81786095 ... 0.41797928 0.         0.15212198]
 [0.18879995 0.62201282 0.77930119 ... 0.35720492 0.15212198 0.        ]]
```

e. Clustering

Berikutnya adalah melakukan clustering dengan cara memanggil class Agglomerative yang telah di import dan memanggil fungsi fit.

Berikut hasil menggunakan **single** linkage.

```
agglom = AgglomerativeClustering(n_clusters=6, linkage='single')
agglom.fit(feature_mtx)
agglom.labels_
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       3, 0, 0, 0, 0, 0, 0, 0, 0, 5, 1, 4, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 2, 0, 0, 0, 0, 0])
```

Berikut hasil menggunakan **average** linkage

```
agglom2 = AgglomerativeClustering(n_clusters=6, linkage='average')
agglom2.fit(feature_mtx)
agglom2.labels_
```

```
array([0, 4, 4, 0, 4, 4, 0, 4, 4, 4, 4, 4, 4, 4, 0, 0, 4, 4, 4, 1, 0,
       3, 0, 0, 4, 0, 4, 0, 0, 0, 1, 5, 2, 2, 4, 4, 0, 4, 0, 4, 4, 4, 4,
       2, 4, 5, 0, 0, 0, 4, 4, 0, 0, 0, 4, 0, 0, 4, 4, 4, 4, 4, 4, 0, 0,
       0, 4, 0, 4, 0, 0, 0, 4, 4, 4, 4, 0, 4, 4, 1, 0, 0, 4, 4, 4, 0, 4,
       4, 4, 0, 0, 4, 0, 0, 4, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0,
       0, 2, 0, 0, 0, 0, 0])
```

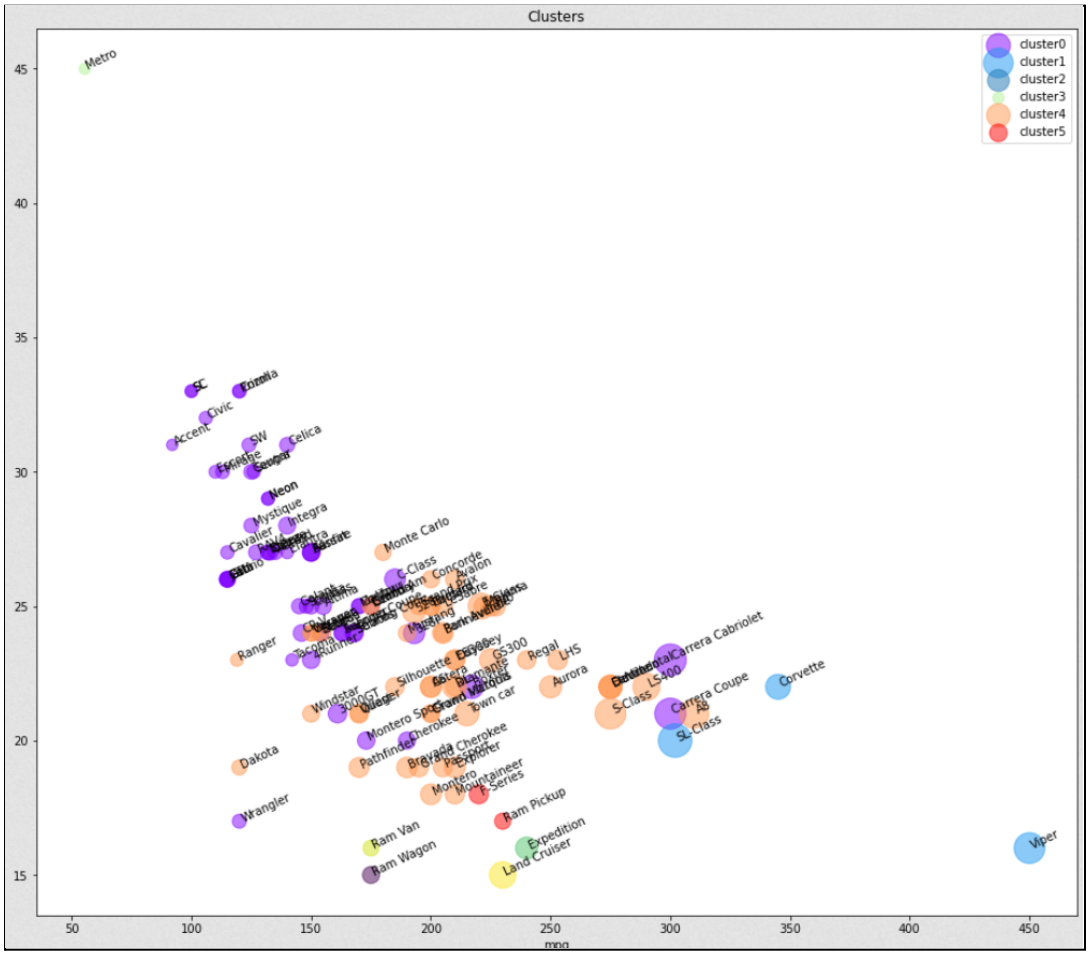
f. Hasil

Ini adalah hasil dari agglomerative clustering.

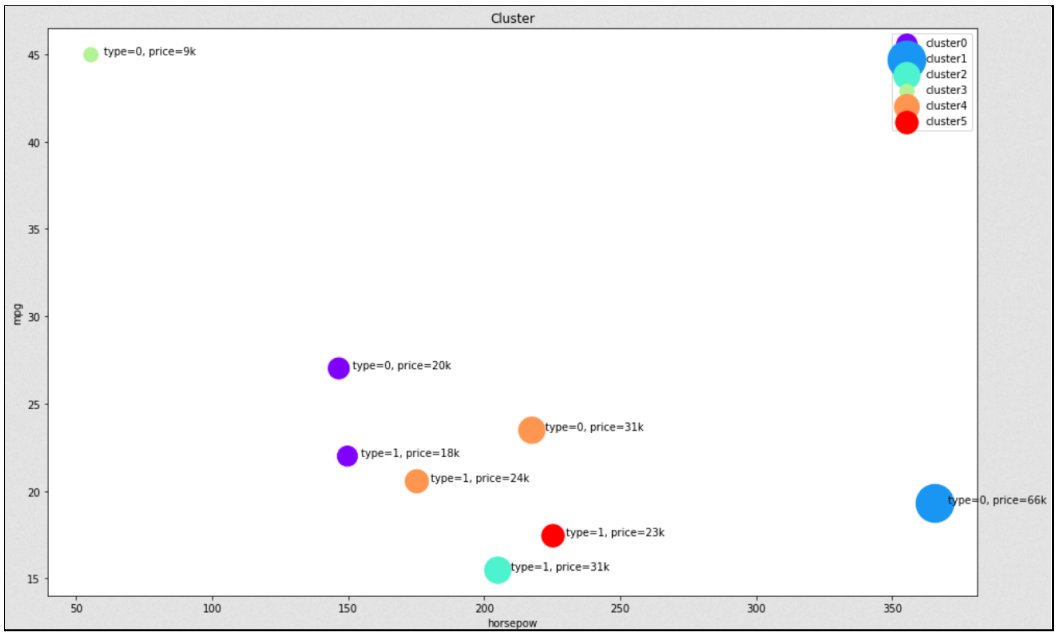
	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition	cluster_
112	Volkswagen	Golf	9.761	11.425	0.0	14.90	2.0	115.0	98.9	68.3	163.3	2.767	14.5	26.0	2.278	0	0
113	Volkswagen	Jetta	83.721	13.240	0.0	16.70	2.0	115.0	98.9	68.3	172.3	2.853	14.5	26.0	4.427	0	0
114	Volkswagen	Passat	51.102	16.725	0.0	21.20	1.8	150.0	106.4	68.5	184.1	3.043	16.4	27.0	3.934	0	0
115	Volkswagen	Cabrio	9.569	16.575	0.0	19.99	2.0	115.0	97.4	66.7	160.4	3.079	13.7	26.0	2.259	0	0
116	Volkswagen	GTI	5.596	13.760	0.0	17.50	2.0	115.0	98.9	68.3	163.3	2.762	14.6	26.0	1.722	0	0

g. Persebaran cluster data

Ini adalah gambar persebaran cluster yang dibentuk dari data cars. Karena masih antar cluster masih tumpang tindih, maka selanjutnya akan dilakukan pengelompokkan sehingga data cluster lebih mudah dilihat.



Hasil dari pengelompokkan data kurang lebih seperti ini.



h. Karakteristik tiap kluster

Terakhir, ini ada karakteristik dari cluster yang dibentuk. Hasil ini biasanya digunakan untuk bahan analisis dalam menentukan harga mobil berdasarkan kriteria/fitur tertentu.

		horsepow	engine_s	mpg	price
cluster_	type				
0	0.0	146.531915	2.246809	27.021277	20.306128
	1.0	149.714286	2.657143	22.000000	18.551571
1	0.0	365.666667	6.233333	19.333333	66.010000
2	1.0	205.000000	4.275000	15.500000	31.938250
3	0.0	55.000000	1.000000	45.000000	9.235000
4	0.0	217.540541	3.602703	23.481081	31.837027
	1.0	175.250000	3.287500	20.562500	24.674875
5	1.0	225.000000	4.900000	17.500000	23.197500

3. Agglomerative dengan data bunga iris

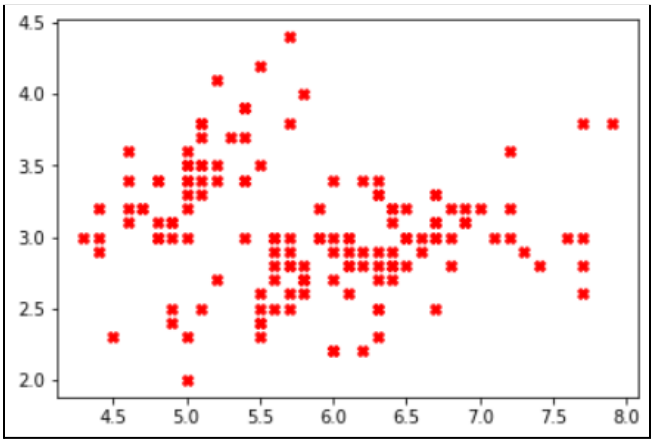
a. Data

Dataset yang digunakan untuk soal nomor 3 adalah dataset bunga iris dengan isi sebagai berikut.

	sepal-length	sepal-width	petal-length	petal-width	class
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

b. Plotting

Untuk melihat persebaran dari dataset ini, maka akan dilakukan proses pembuatan plotting data.



c. Agglomerative Clustering

Sebelum masuk ke clustering, ada beberapa hal yang perlu dilakukan dikarenakan dataset bunga iris ini agak sedikit berbeda dengan dataset sebelumnya. Hal yang dilakukan hanyalah memisahkan fitur dengan label.

```
feature = df.iloc[:, 0:4]
label = df.iloc[:, 4:6]

feature = feature.to_numpy()
label = label.to_numpy()
```

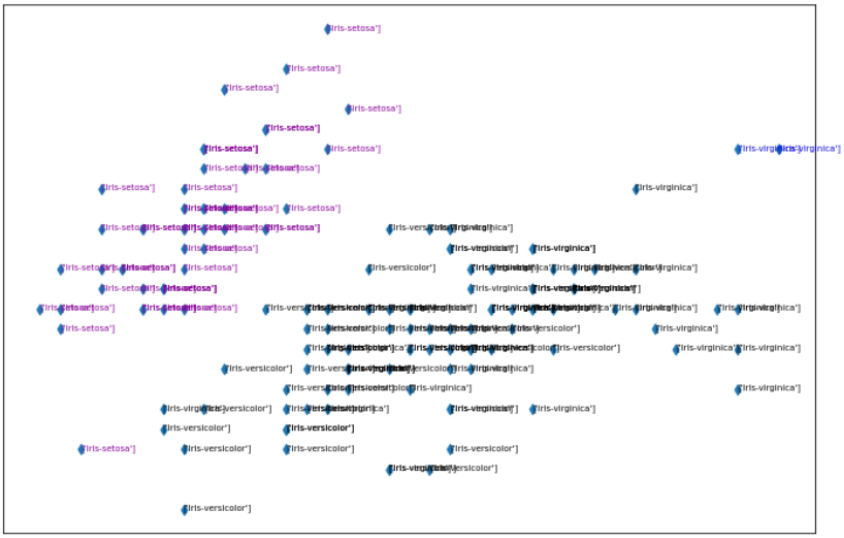
Selanjutnya adalah melakukan clustering.

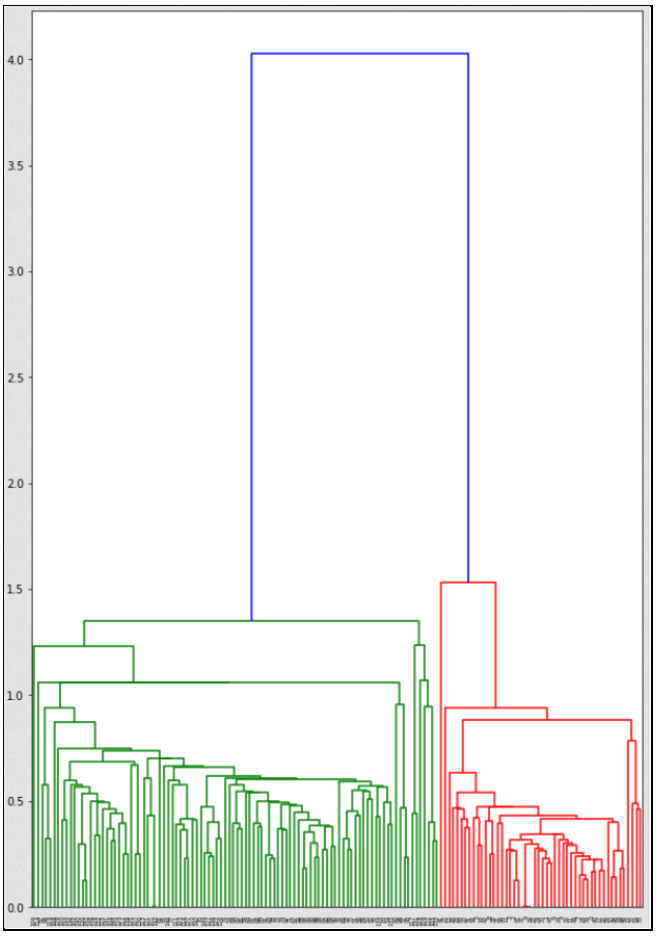
```
agg_sing = AgglomerativeClustering(n_clusters=3, linkage='single')
agg_sing.fit(feature, label)
print(agg_sing)

agg_ave = AgglomerativeClustering(n_clusters=3, linkage='average')
agg_ave.fit(feature, label)
print(agg_ave)

agg_com = AgglomerativeClustering(n_clusters=3, linkage='complete')
agg_com.fit(feature, label)
print(agg_com)
```

d. Plotting Dendrogram

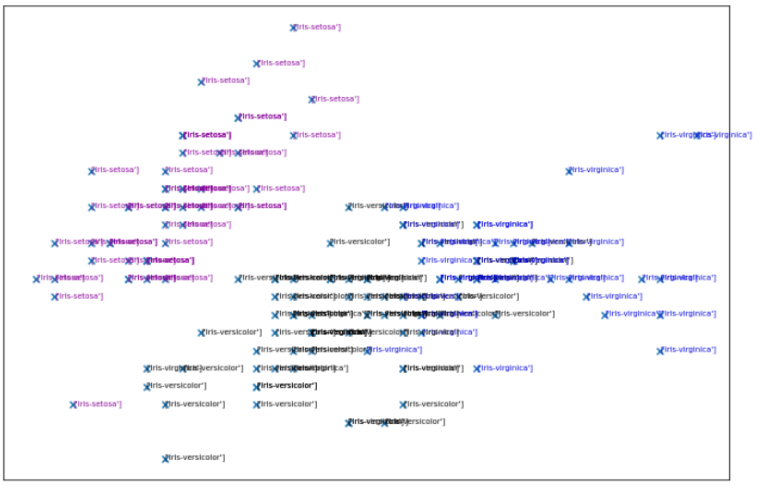
Selanjutnya adalah plotting hasil clustering dan dendogram untuk masing-masing metode clustering.

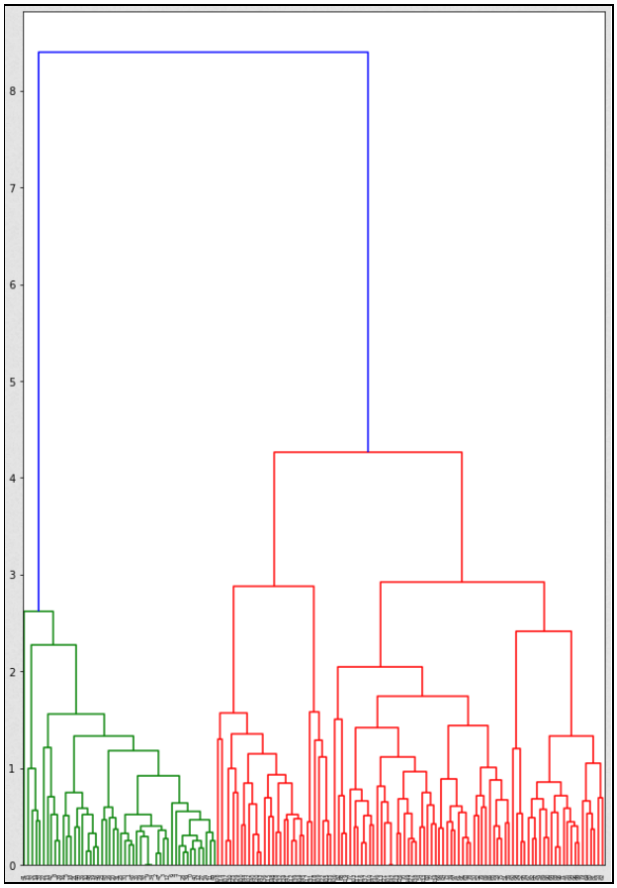
- Single



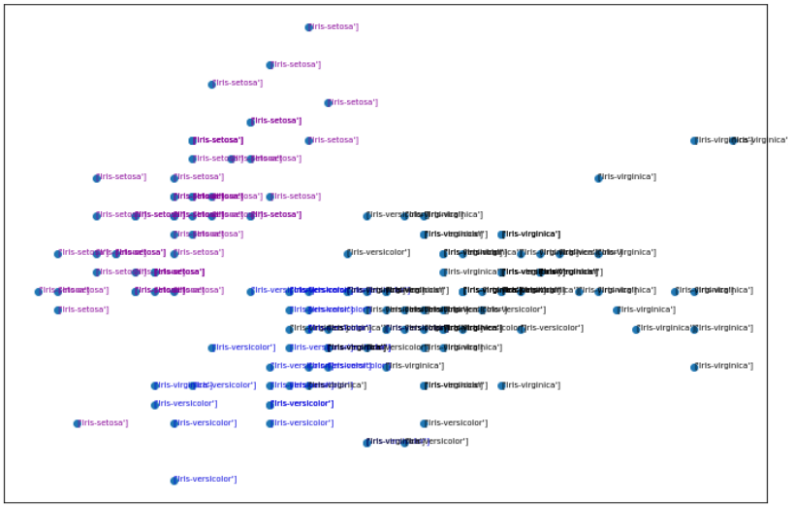


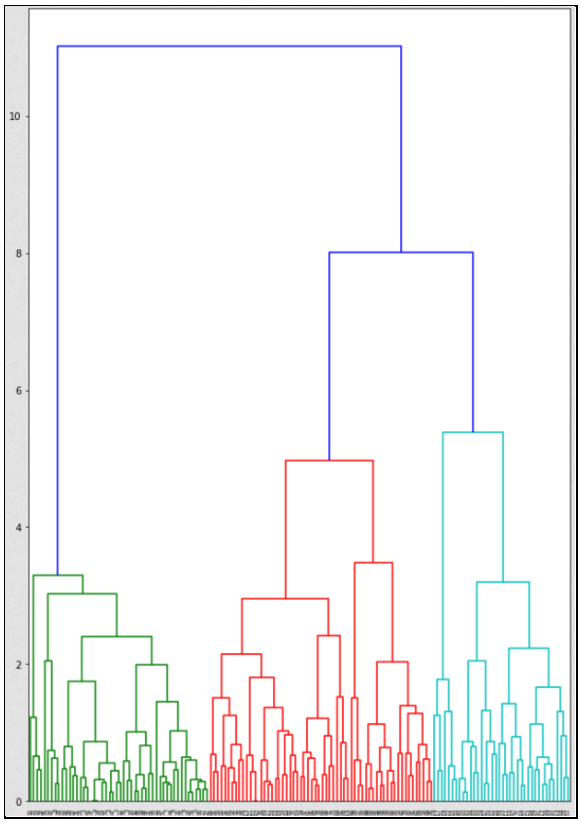
- Average





- Complete




10
8
6
4
2
0

BAB III PENUTUP

3.1. Kesimpulan

Pada praktikum pertemuan kelima ini membahas mengenai agglomerative clustering dengan tiga buah dataset yang berbeda, yaitu random dataset, cars dataset, dan dataset bunga iris. Untuk metode agglomerative yang digunakan ada tiga, single linkage, average linkage, dan complete linkage. Di praktikum ini juga dilakukan pembuatan dendogram untuk menunjukkan histori dari pembentukan cluster.

DAFTAR PUSTAKA

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Systems with Applications, 129, 273-285.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12, 2825–2830.*

Mitchell, Tom (1997). *Machine Learning*. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.

Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F., "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning" IEEE Transactions on Vehicular Technology, 2020.

"What is Machine Learning?". *www.ibm.com*. Retrieved 2021-08-15.

Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks". *Medium*. Retrieved 2021-08-15.