Bab 3
Hierarchical Clustering

A. KOMPETENSI DASAR

- Memahami konsep *Hierarchical Clustering*.
- Memahami eksperimen clustering menggunakan Agglomerative Hierarchical Clustering (AGNES)
- Mengetahui cara mengevaluasi AGNES

B. ALOKASI WAKTU

4 js (4x50 menit)

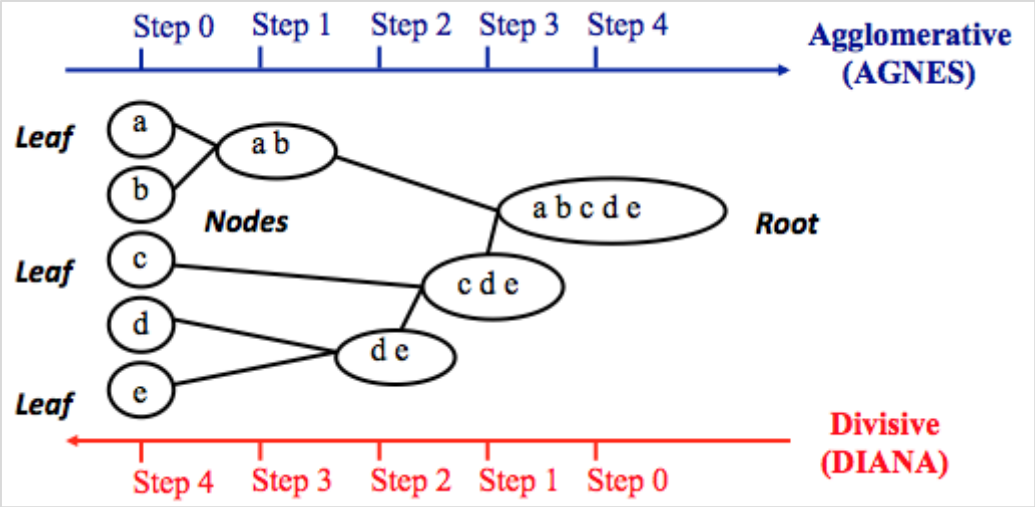
C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami Tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas dengan baik, sabar, dan jujur.
- Tanyakan kepada teman lalu asisten/dosen apabila ada hal-hal yang kurang jelas.

D. DASAR TEORI HIERARCHICAL CLUSTERING

Hierarchical clustering adalah keluarga umum dari algoritma clustering yang membangun cluster bersarang dengan menggabungkan atau memecahnya secara berturut-turut. Hirarki cluster ini direpresentasikan sebagai pohon (atau *dendrogram*). Akar pohon (*root*) adalah kluster unik yang mengumpulkan semua sampel, daun (*leaf*) menjadi cluster dengan hanya satu sampel.

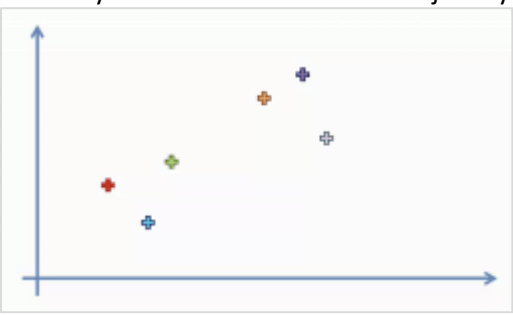
Scikit-learn menyediakan algoritma dengan nama AgglomerativeClustering untuk digunakan dalam *hierarchical clustering*. Algoritma AgglomerativeClustering (AGNES) melakukan pengelompokan hierarkis menggunakan pendekatan *bottom-up*: setiap objek pada awalnya dianggap sebagai satu elemen-cluster (leaf). Pada setiap langkah algoritma, dua cluster yang paling mirip digabungkan menjadi sebuah cluster baru yang lebih besar (node). Prosedur ini diulang sampai semua titik adalah anggota dari hanya satu cluster pangkal (root). Kebalikan dari AgglomerativeClustering adalah Divisive Clustering, yang juga dikenal sebagai DIANA (Divisive Analysis) dan berfungsi secara "top-down". Itu dimulai dengan root, di mana semua objek termasuk dalam satu cluster. Pada setiap langkah iterasi, kluster yang paling heterogen dibagi menjadi dua. Proses ini diulangi sampai semua objek berada di cluster mereka sendiri (lihat gambar di bawah).



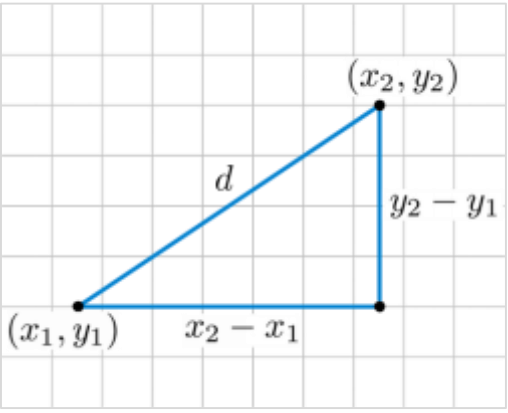
Gambar 1. Tahapan Agglomerative dan Divisive Hierarchical Clustering

Algoritma AGNES:

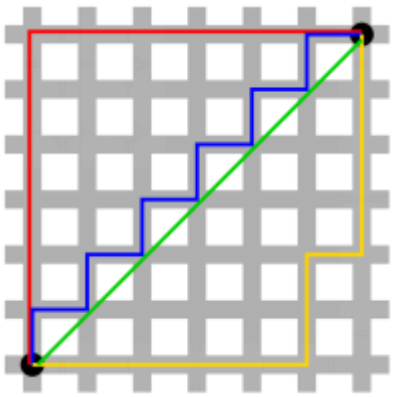
1. **Persiapan data** dalam bentuk tabel: kolom sebagai atribut dan baris sebagai sampel.
2. **Menghitung jarak (dis-*similarity*) tiap objek/instance dalam dataset**. Jenis jarak yang diukur pada umumnya adalah **Euclidean**. Pilihan jarak yang lain adalah Manhattan, Sine, Cosine, Mahalanobis, dll.



Gambar 2. Perlakuan tiap objek sebagai satu klaster.



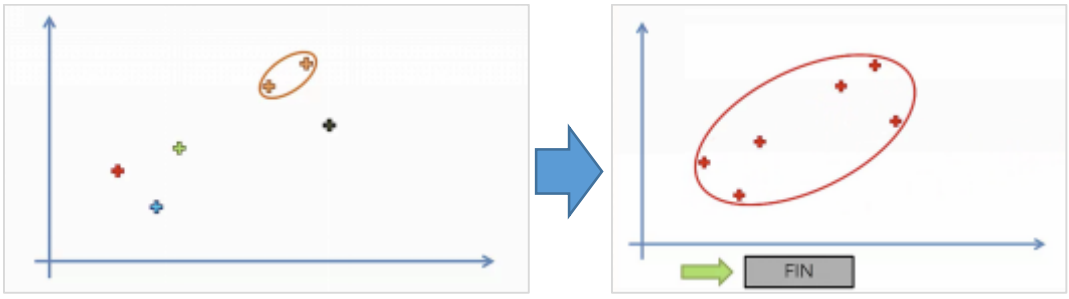
a)



b)

Gambar 3. a) Ilustrasi penghitungan jarak Euclidean, b) Ilustrasi penghitungan jarak Manhattan.

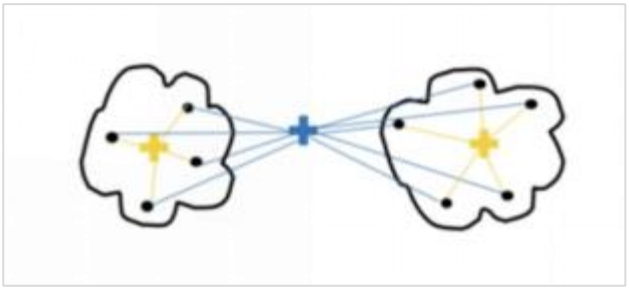
3. **Menghitung *linkage*** untuk membentuk pohon klaster. Objek dengan jarak terdekat akan menjadi bagian dari klaster sesuai fungsi *linkage*. Tahap ini dilakukan secara iteratif hingga tersisa satu klaster.



Gambar 4. Pembentukan klaster dari dua objek terdekat sampai terbentuksatu klaster besar.

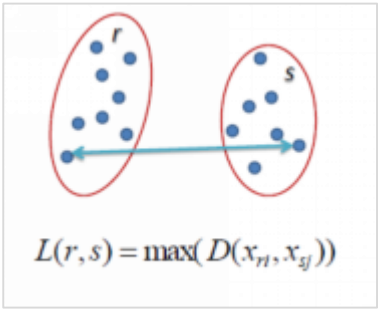
Kriteria *linkage* menentukan metrik yang digunakan untuk strategi penggabungan klaster:

- **Ward** meminimalkan nilai terkecil jarak kuadrat dalam semua cluster. Ini adalah pendekatan meminimalkan varians dan dalam hal ini mirip dengan fungsi objektif k-means tetapi ditangani dengan pendekatan hierarkis aglomeratif.



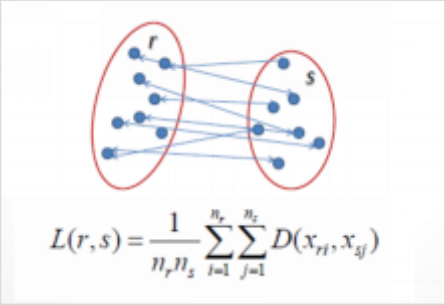
Gambar 5. Ward linkage

- **Maximum** or **complete linkage** meminimalkan jarak maksimum antara pengamatan pasangan cluster.



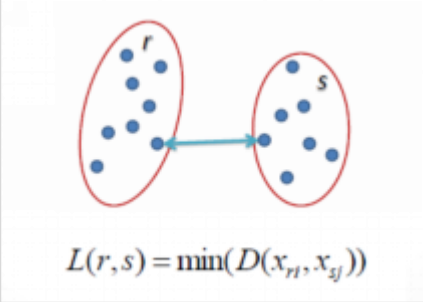
Gambar 6. Complete linkage

- **Average linkage** meminimalkan rata-rata jarak antara semua pengamatan pasangan cluster.



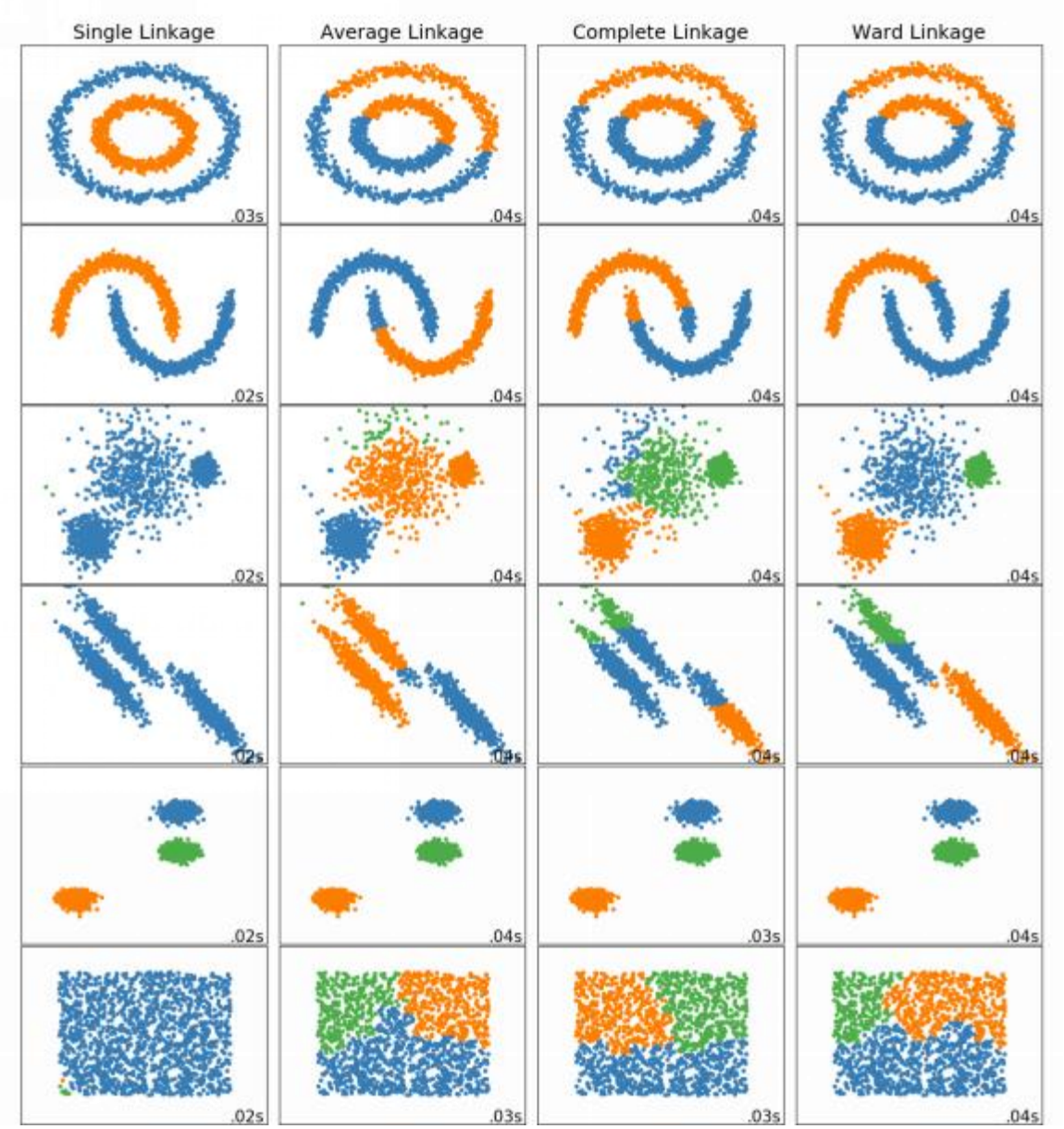
Gambar 7. Average linkage

- **Single linkage** meminimalkan jarak antara pengamatan terdekat dari pasangan cluster.



Gambar 8. Single linkage

AgglomerativeClustering memiliki perilaku "kaya menjadi lebih kaya" yang mengarah pada ukuran kluster yang tidak merata. Dalam hal ini, *single linkage* adalah strategi terburuk, dan *Ward* memberikan ukuran paling teratur. Namun, afinitas (atau jarak yang digunakan dalam pengelompokan) yang tepat untuk linkage Ward adalah metode pengukuran jarak Euclidean saja yang dapat digunakan. Sehingga untuk metrik afinitas selain Euclidean, *average linkage* adalah alternatif yang baik. *Single linkage*, meskipun tidak kuat untuk data yang kotor, dapat dihitung dengan sangat efisien dan karenanya dapat berguna untuk menyediakan pengelompokan hierarkis dari kumpulan data yang lebih besar. Tautan tunggal juga dapat bekerja dengan baik pada data non-global.



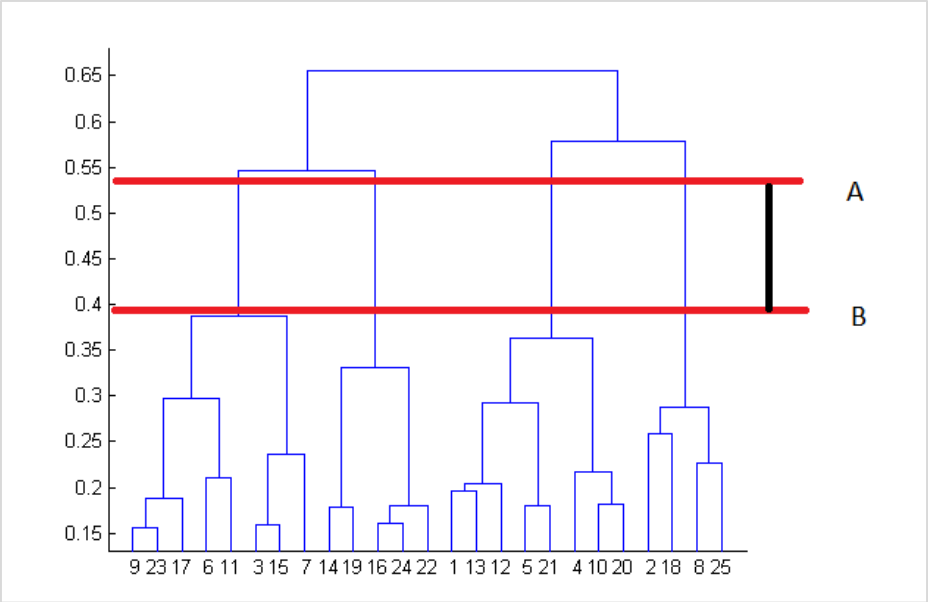
Gambar 9. Linkage pada AgglomerativeClustering

4. Menentukan posisi “pemotongan” pohon untuk mengetahui jumlah klaster yang digunakan.

Dendrogram

Kita dapat menggunakan dendrogram untuk memvisualisasikan tahapan pengelompokan dan mengidentifikasi jumlah cluster yang terbaik menggunakan beberapa metode sebagai berikut:

- Identifikasi jarak vertikal terbesar yang tidak memotong salah satu dari kluster lainnya (lihat contoh garis A dan B)
- Gambar garis horizontal pada kedua ekstremitas
- Jumlah optimal cluster sama dengan jumlah garis vertikal yang melewati garis horizontal



Gambar 10. Dendrogram sebagai observasi untuk pencarian klaster optimal.

E. LATIHAN (jawaban dapat ditulis pada halaman baru)

1. Persiapkan data berupa array 2 dimensi sebagai berikut: `[[5,3], [10,15], [15,12], [24,10], [30,30], [85,70], [71,80], [60,78], [70,55], [80,91],];` Kemudian simpan ke dalam file `data.csv`. Perhatikan format penulisan data menggunakan teknik comma separated values (CSV). **Tulislah data tsb dalam format .csv!**

2. Load data.csv dan Plotting data menggunakan program di bawah ini. **Bagaimana program yang benar?Bagaimana tampilan plot data nya?**

```
import matplotlib.pyplot as plt
import pandas as pd

x = pd.read-csv('data.csv')
labels = range(1, 11)
plt.figure(figsize:(10, 7))
plt.subplots_adjust(bottom=0,1)
plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
        label,
        xy=(x, y), xytext=(-3, 3),
        textcoords='offset points', ha='right', va='bottom')
plt.show()
```

3. Visualisasi data melalui AGNES dan dendrogram

```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

linked = linkage(X, 'single')

labelList = range(1, 11)

plt.figure(figsize=(10, 7))
dendrogram(linked,
            orientation='top',
            labels=labelList,
            distance_sort='descending',
            show_leaf_counts=True)
plt.show()
```

F. LATIHAN (AGNES menggunakan library sklearn)

1. **Impor *libraries*** yang diperlukan

```
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
import numpy as np
```

2. **Load dataset**

```
x = pd.read_csv('data.csv')
```

3. **Implementasi fungsi AGNES. Eksperimen dengan parameter n_cluster, affinity dan linkage yang berbeda!**

```
from sklearn.clusters import AgglomerativeClustering

cluster = AgglomerativeClustering(n_clusters=2, affinity='euclidean',
linkage='ward')
cluster.fit_predicts(X)
```

4. **Tampilkan hasil cluster.**

```
print(cluster.labels)
```

5. **Visualisasi Plot. Apakah cluster yang terbentuk memiliki pengelompokan yang tepat?**

```
plt.scater(X[:,0],X[:,1], c=cluster.labels_, cmap='rainbow')
```

G. LATIHAN (AGNES menggunakan library sklearn dan dendrogram pada dataset baru)

1. **Download dataset dari**
https://stackabuse.s3.amazonaws.com/files/hierarchical-clustering-with-python-and-scikit-learn-shopping-data.csv

2. **Load dataset tersebut menjadi variabel array**

```
data_pelanggan = pd.read_csv('D:\Datasets\shopping-data.csv')
```

3. **Tampilkan dan Amati dimensi dari dataset. Berapa jumlah kolom dan baris dataset nya?**

```
customer_data.shaped
```

4. Tampilkan dan amati bentuk tabular dari dataset. Bagaimana bentuk tampilannya?kolom merepresentasikan apa?baris merepresantasikan apa?

```
customer_data.heading()
```

5. **Pada eksperimen ini, lakukan pemilihan data untuk clustering hanya pada Annual Income dan Spending Score. Ubahlah tanda tanya pada script sehingga dataset yang dipakai adalah Annual Income dan Spending Score.**

```
data = customer_data.iloc[:, ?:?].values[]
```

6. **Observasi dendrogram**

```
import scipy.cluster.hierarchy as shc

plt.figure(figsize=(10, 7))
plt.title("Customer Dendograms")
dend = shc.dendrogram(shc.linkage(data, method='ward'))
```

7. **Estimasi jumlah cluster terbaik dengan cara menarik garis horisontal pada *branch* terdekat. Gunakan editor gambar untuk menambahkan garis horisontal tersebut. Berapa jumlah cluster terbaik?**

8. **Gunakan library AGNES untuk clustering dan bandingkan dengan cluster yang dihasilkan dari langkah #7.**

```
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n_cluster=5, distance='euclidean',
lingkage='ward')
cluster.fit-predict(data)
```

9. Plot hasil clustering.

```
plt.figure(figsize=(10, 7))
plt.scater(data[:,0], data[:,1], c=cluster.labels_, cmap='rainbow')
```

H. TUGAS

1. Buatlah/carilah fungsi untuk menampilkan garis horisontal pada dendrogram! Tampilkan dan jelaskan jalannya fungsi!
2. Buatlah fungsi yang secara otomatis menampilkan garis horisontal pada posisi jumlah cluster terbaik! Tampilkan dan jelaskan jalannya fungsi!
3. Jawaban-jawaban **Latihan E,F,G** dan **Tugas #1 dan #2** ditulis pada halaman baru.
4. Buatlah kesimpulan tentang Model Hirearchial Clustering (Algoritma AGNES)