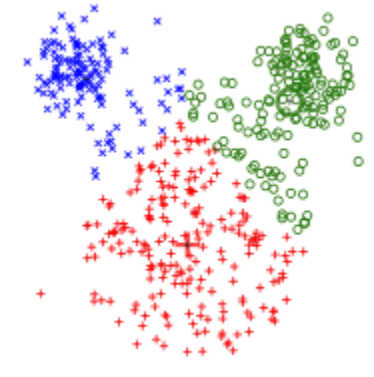
Webinar Series, 7 Nopember 2020
PROGRAM PASCASARJANA TERAPAN
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

Workshop & Tutorial
Data Mining with Python



Clustering & Cluster Analysis

Ali Ridho Barakbah

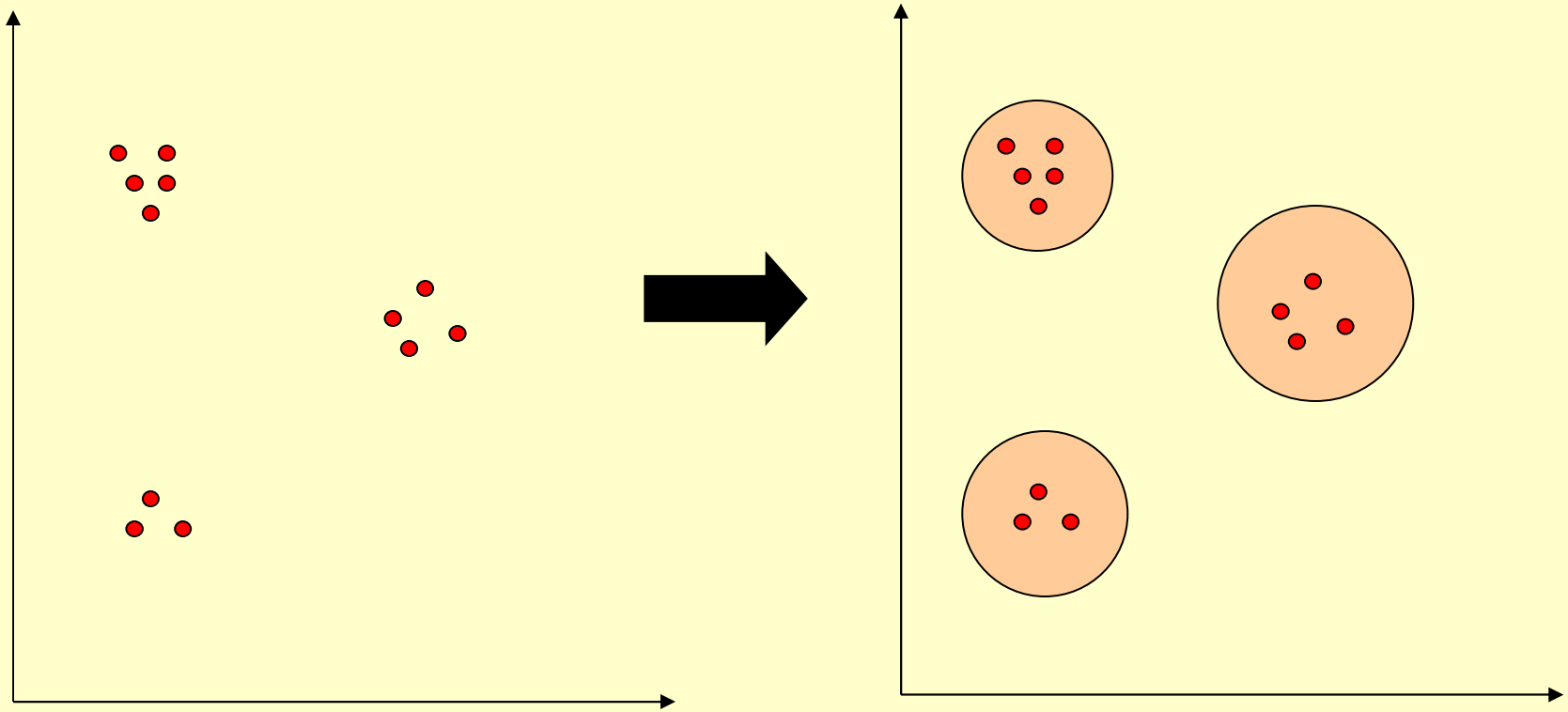
Knowledge Engineering Laboratory
Department of Information and Computer Engineering
Politeknik Elektronika Negeri Surabaya

What is clustering?

the process of organizing objects into groups whose members are similar in some way

http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/index.html

Ilustrasi clustering



Similaritas berdasarkan jarak

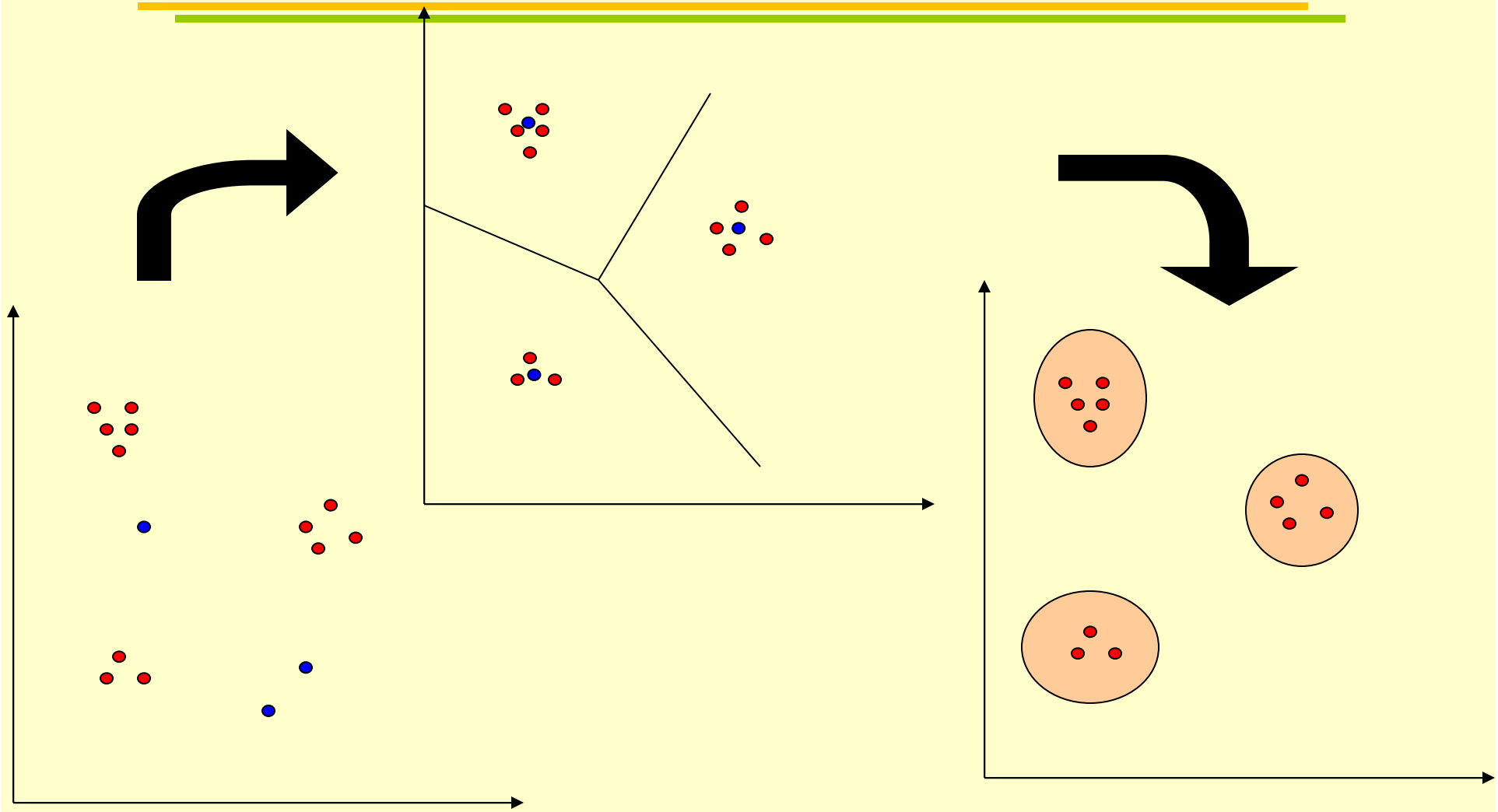
K-means

- Termasuk partitioning clustering yang memisahkan data ke *k* daerah bagian yang terpisah
- K-means algorithm sangat terkenal karena kemudahan dan kemampuannya untuk mengklaster data besar dan data outlier dengan sangat cepat
- Setiap data harus termasuk ke cluster tertentu
- Memungkinkan bagi setiap data yang termasuk cluster tertentu pada suatu tahapan proses, pada tahapan berikutnya berpindah ke cluster yang lain

Algoritma K-means

1. Tentukan *k* sebagai jumlah cluster yang ingin dibentuk
2. Bangkitkan *k* centroids (titik pusat cluster) awal secara random
3. Hitung jarak setiap data ke masing-masing centroids
4. Setiap data memilih centroids yang terdekat
5. Tentukan posisi centroids baru dengan cara menghitung nilai rata-rata dari data-data yang memilih pada centroid yang sama
6. Kembali ke langkah 3 jika posisi centroids baru dengan centroids lama tidak sama.

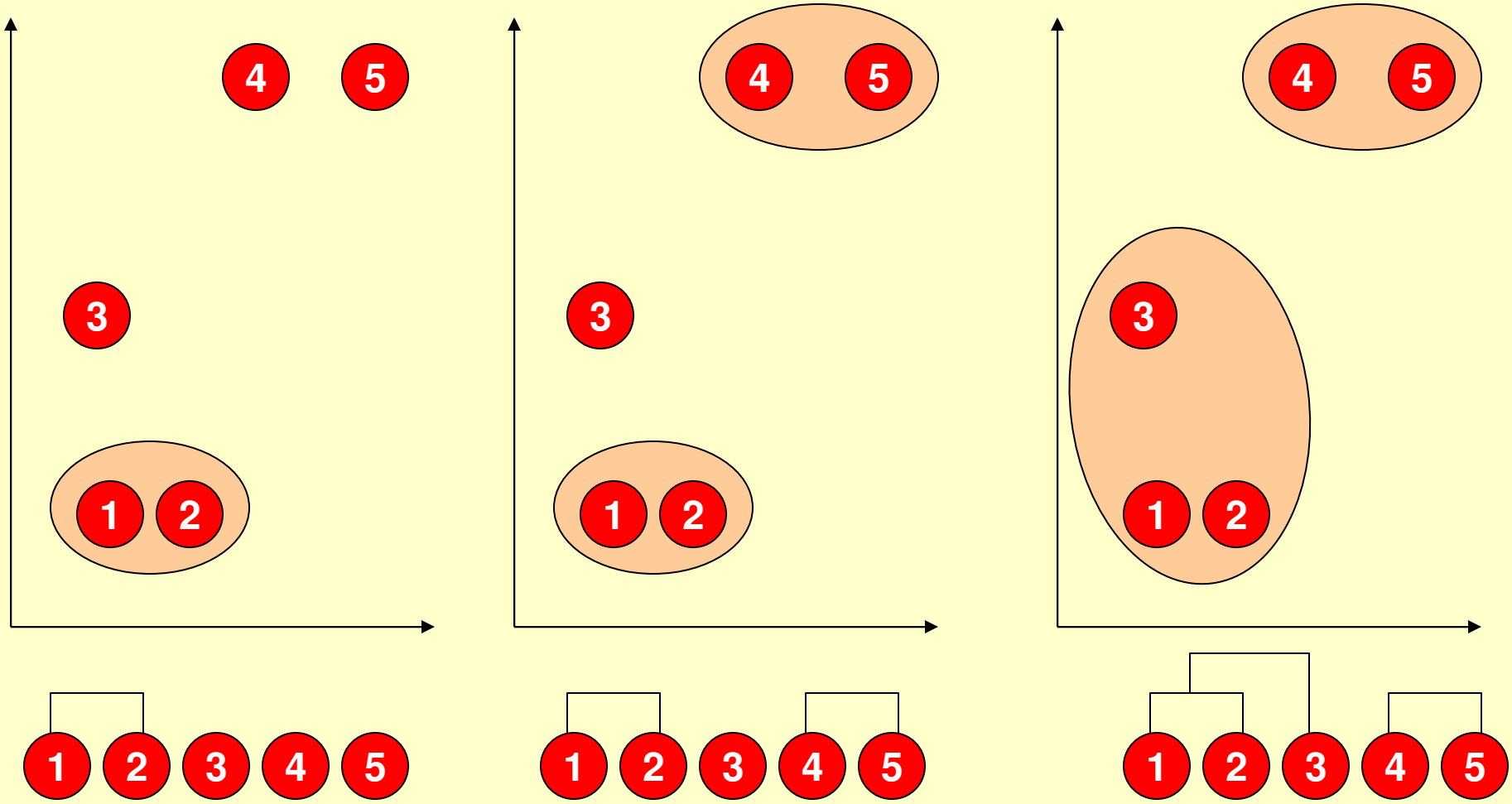
Algoritma K-means



Algoritma Hierarchical clustering

1. Tentukan *k* sebagai jumlah cluster yang ingin dibentuk
2. Setiap data dianggap sebagai cluster. Kalau *n*=jumlah data dan *nc*=jumlah cluster, berarti ada *nc=n*.
3. Hitung jarak antar cluster
4. Cari 2 cluster yang mempunyai jarak antar cluster yang paling minimal dan gabungkan (berarti *nc* berkurang)
5. Jika *nc>k*, kembali ke langkah 3

Algoritma Hierarchical clustering



Similarity between clusters?

- Single Linkage

  → Minimum distance between cluster
- Centroid Linkage

  → Centroid distance between cluster
- Complete Linkage

  → Maximum distance between cluster
- Average Linkage

  → Average distance between cluster

Hierarchical Clustering & Dataset

- **Single Linkage**

  Metode ini sangat cocok untuk dipakai pada kasus shape independent clustering, karena kemampuannya untuk membentuk pattern tertentu dari cluster. Untuk kasus condensed clustering, metode ini tidak bagus.

- **Centroid Linkage**

  Metode ini baik untuk kasus clustering dengan normal data set distribution. Akan tetapi, metode ini tidak cocok untuk data yang mengandung outlier.

- **Complete Linkage**

  Metode ini sangat ampuh untuk memperkecil variance within cluster karena melibatkan centroid pada saat penggabungan antar cluster. Metode ini juga baik untuk data yang mengandung outlier.

- **Average Linkage**

  Metode ini relatif yang terbaik dari metode-metode hierarchical. Namun, ini harus dibayar dengan waktu komputasi yang paling tinggi dibandingkan dengan metode-metode hierarchical yang lain.

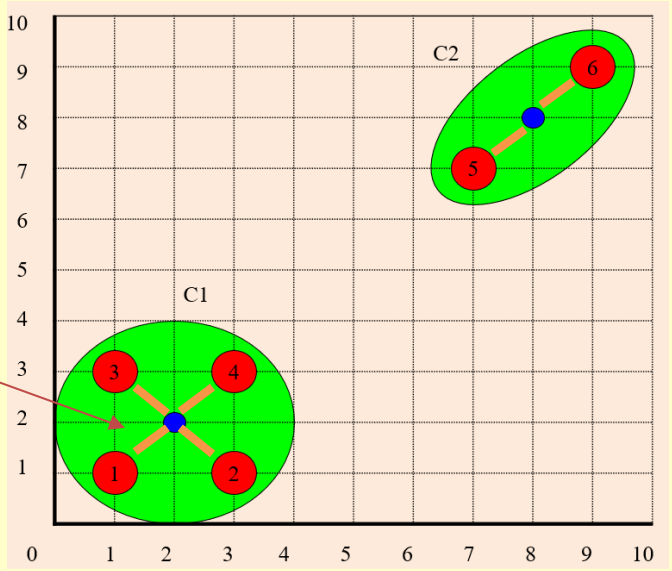
Cluster Analysis

is when the members of a cluster have a high degree of similarity to each other (internal homogeneity) and are not like members of other clusters (external homogeneity)

Cluster Analysis – (Sum of Squared Error)

- The most widely used criterion to quantify cluster homogeneity is the Sum of Squared Error (SSE) criterion

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n(s_i)} \left\| m_{ij} - \bar{s}_i \right\|^2$$



Eksperimen dengan Dataset IPM

Nama Provinsi	Kode Kabkota	Nama Kabkota	Koordinat Kabkota Latitude	Koordinat Kabkota Longitude	Tahun	Angka Harapan Hidup	Angka Melek Huruf	Lama Sekolah	Pengeluaran Perkapita	Ipm

Sumber: Indeks Pembangunan Manusia 2004-2012, BPS

Clustering – (K-Means)
Cluster Analysis – (SSE)

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans


dataset = pd.read_csv('ipm.csv')
dataset = dataset.dropna()
avg_ipm = dataset.groupby('nama_provinsi')['ipm'].mean()

print('Rata-rata IPM:\n', avg_ipm)

clustering = KMeans(n_clusters=3, init="random", n_init=1)
clusters=clustering.fit_predict(avg_ipm.values.reshape(-1, 1))

print('\nHasil clustering:\n', clusters)
```

```
Rata-rata IPM:
 nama_provinsi
Prov. Bali                          71.394288
Prov. Banten                        70.839972
Prov. Bengkulu                      69.712691
Prov. D I Yogyakarta                74.515497
Prov. DKI Jakarta                   76.556860
Prov. Gorontalo                     69.157070
Prov. Jambi                         72.235563
Prov. Jawa Barat                    71.658552
Prov. Jawa Tengah                   71.457085
Prov. Jawa Timur                    69.494746
Prov. Kalimantan Barat              67.199175
Prov. Kalimantan Selatan            69.576503
Prov. Kalimantan Tengah             72.740934
Prov. Kalimantan Timur              73.598787
Prov. Kepulauan Bangka Belitung     70.675152
Prov. Kepulauan Riau                72.515945
Prov. Lampung                       70.352195
Prov. Maluku                        69.987349
Prov. Maluku Utara                  68.116775
Prov. Nanggroe Aceh Darussalam      70.711394
Prov. Nusa Tenggara Barat           64.089688
Prov. Nusa Tenggara Timur           65.241750
Prov. Papua                         58.891675
Prov. Papua Barat                   66.272960
Prov. Riau                          73.620722
Prov. Sulawesi Barat                68.220363
Prov. Sulawesi Selatan              70.660213
Prov. Sulawesi Tengah               68.999099
Prov. Sulawesi Tenggara             68.650826
Prov. Sulawesi Utara                74.017236
Prov. Sumatera Barat                72.008713
Prov. Sumatera Selatan              70.182083
Prov. Sumatera Utara                72.581676
Name: ipm, dtype: float64

Hasil clustering:
 [1 2 2 1 1 2 1 1 1 2 2 2 1 1 2 1 2 2 2 2 0 0 0 0 1 2 2 2 2 1 1 2 1]

SSE =  75.15996113707094
```

Clustering dengan Interpretasi

```
import pandas as pd
import numpy as np
from sklearn.cluster import AgglomerativeClustering

dataset = pd.read_csv('ipm.csv')
dataset = dataset.dropna()
avg_ipm = dataset.groupby('nama_provinsi')['ipm'].mean()

clustering=AgglomerativeClustering(n_clusters=3, linkage='average')
clusters=clustering.fit_predict(avg_ipm.values.reshape(-1, 1))

avg_ipm=pd.DataFrame({'Provinsi':avg_ipm.index, 'Rata-Rata IPM':avg_ipm.values, 'Cluster':clusters})

centroid_perdata=avg_ipm.groupby('Cluster')['Rata-Rata IPM'].transform('mean')
centroid=np.unique(centroid_perdata)

sorted_centroid=np.sort(centroid)

rendah, sedang, tinggi = sorted_centroid[0], sorted_centroid[1], sorted_centroid[2]
category=centroid_perdata.map({rendah:'rendah', sedang:'sedang', tinggi:'tinggi'})
avg_ipm['Category']=category

print(avg_ipm[['Provinsi', 'Category']])
```

	Provinsi	Category
0	Prov. Bali	sedang
1	Prov. Banten	sedang
2	Prov. Bengkulu	sedang
3	Prov. D I Yogyakarta	sedang
4	Prov. DKI Jakarta	tinggi
5	Prov. Gorontalo	sedang
6	Prov. Jambi	sedang
7	Prov. Jawa Barat	sedang
8	Prov. Jawa Tengah	sedang
9	Prov. Jawa Timur	sedang
10	Prov. Kalimantan Barat	sedang
11	Prov. Kalimantan Selatan	sedang
12	Prov. Kalimantan Tengah	sedang
13	Prov. Kalimantan Timur	sedang
14	Prov. Kepulauan Bangka Belitung	sedang
15	Prov. Kepulauan Riau	sedang
16	Prov. Lampung	sedang
17	Prov. Maluku	sedang
18	Prov. Maluku Utara	sedang
19	Prov. Nanggroe Aceh Darussalam	sedang
20	Prov. Nusa Tenggara Barat	sedang
21	Prov. Nusa Tenggara Timur	sedang
22	Prov. Papua	rendah
23	Prov. Papua Barat	sedang
24	Prov. Riau	sedang
25	Prov. Sulawesi Barat	sedang
26	Prov. Sulawesi Selatan	sedang
27	Prov. Sulawesi Tengah	sedang
28	Prov. Sulawesi Tenggara	sedang
29	Prov. Sulawesi Utara	sedang
30	Prov. Sumatera Barat	sedang
31	Prov. Sumatera Selatan	sedang
32	Prov. Sumatera Utara	sedang

Clustering dengan Visualisasi

```
import pandas as pd
import numpy as np
from sklearn.cluster import AgglomerativeClustering
from matplotlib import pyplot as plt

dataset = pd.read_csv('ipm.csv')
dataset = dataset.dropna()
avg_ipm = dataset.groupby('nama_provinsi')['ipm'].mean()

clustering=AgglomerativeClustering(n_clusters=3, linkage='average')
clusters=clustering.fit_predict(avg_ipm.values.reshape(-1, 1))

avg_ipm=pd.DataFrame({'Provinsi':avg_ipm.index, 'Rata-Rata IPM':avg_ipm.values, 'Cluster':clusters})

centroid_perdata=avg_ipm.groupby('Cluster')['Rata-Rata IPM'].transform('mean')
centroid=np.unique(centroid_perdata)

sorted_centroid=np.sort(centroid)

rendah, sedang, tinggi = sorted_centroid[0], sorted_centroid[1], sorted_centroid[2]
category=centroid_perdata.map({rendah:'rendah', sedang:'sedang', tinggi:'tinggi'})
avg_ipm['Category']=category

print(avg_ipm[['Provinsi', 'Category']])

x=avg_ipm.index
y=avg_ipm['Rata-Rata IPM']
colors={'rendah':'red', 'sedang':'blue', 'tinggi':'green'}
fig, ax = plt.subplots()
ax.scatter(x, y, c=avg_ipm['Category'].apply(lambda x:colors[x]))

plt.xlabel('Provinsi')
plt.ylabel('Rata-Rata IPM')
plt.xticks(x, avg_ipm['Provinsi'], rotation=90)

plt.show()
```