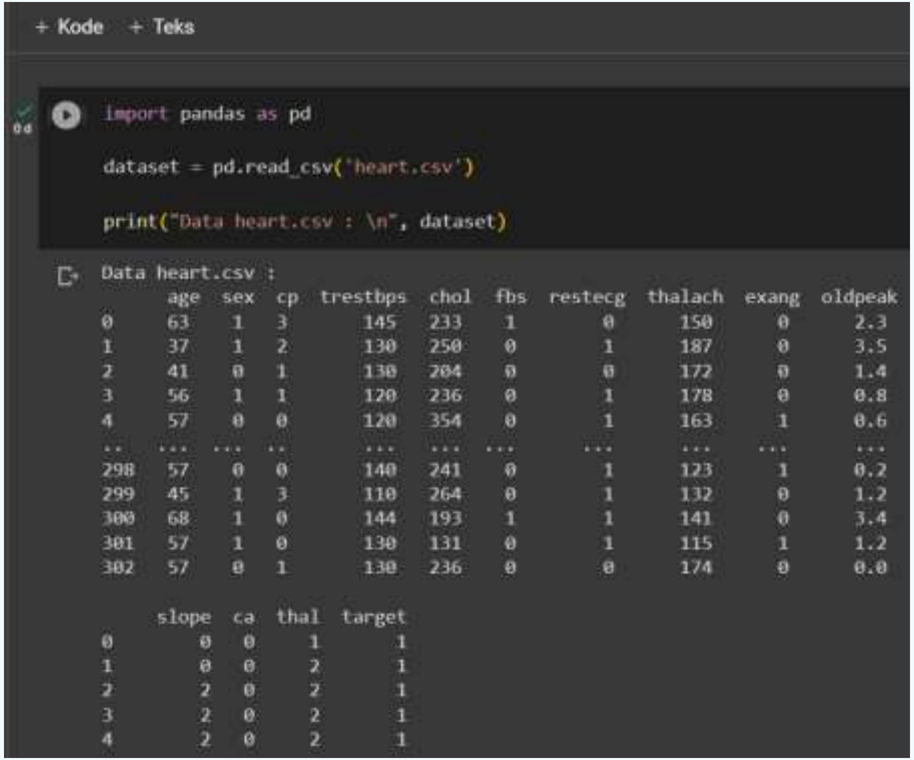
Clustering

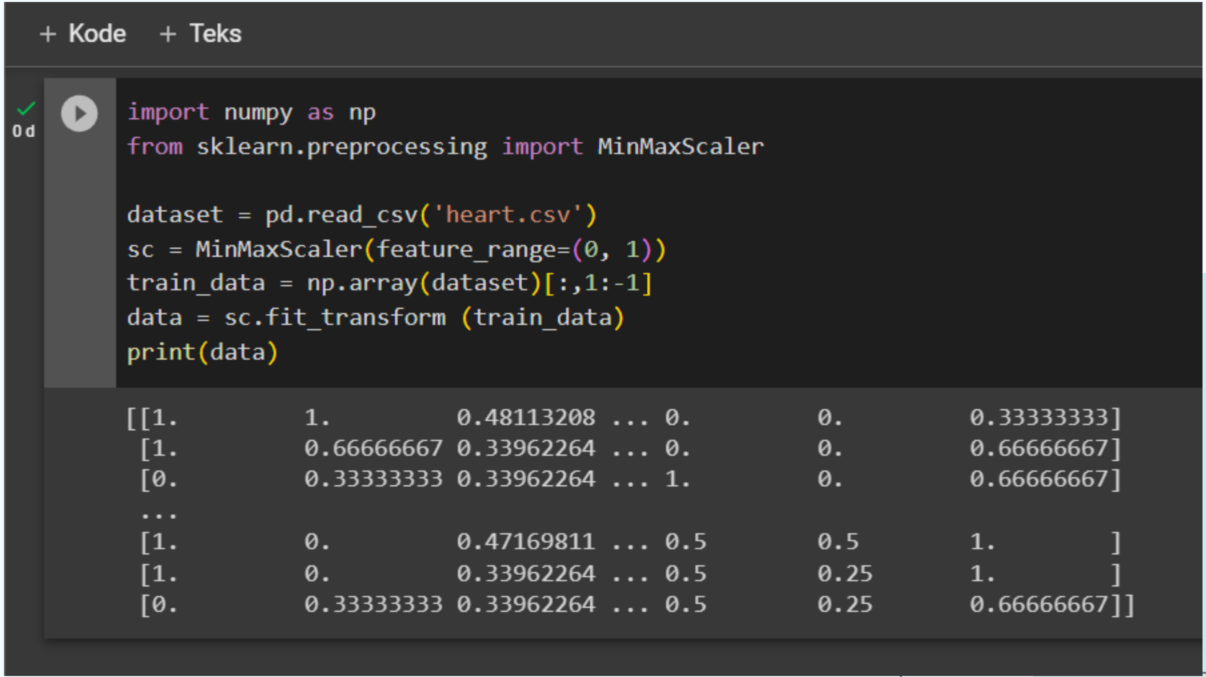
1. dataset ← heart.csv, tampilkan



```python
import pandas as pd

dataset = pd.read_csv('heart.csv')

print("Data heart.csv : \n", dataset)
```

```
Data heart.csv :
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak
0     63    1   3       145   233    1        0      150      0      2.3
1     37    1   2       130   250    0        1      187      0      3.5
2     41    0   1       130   204    0        0      172      0      1.4
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
0        0   0     1       1
1        0   0     2       1
2        2   0     2       1
3        2   0     2       1
4        2   0     2       1
```

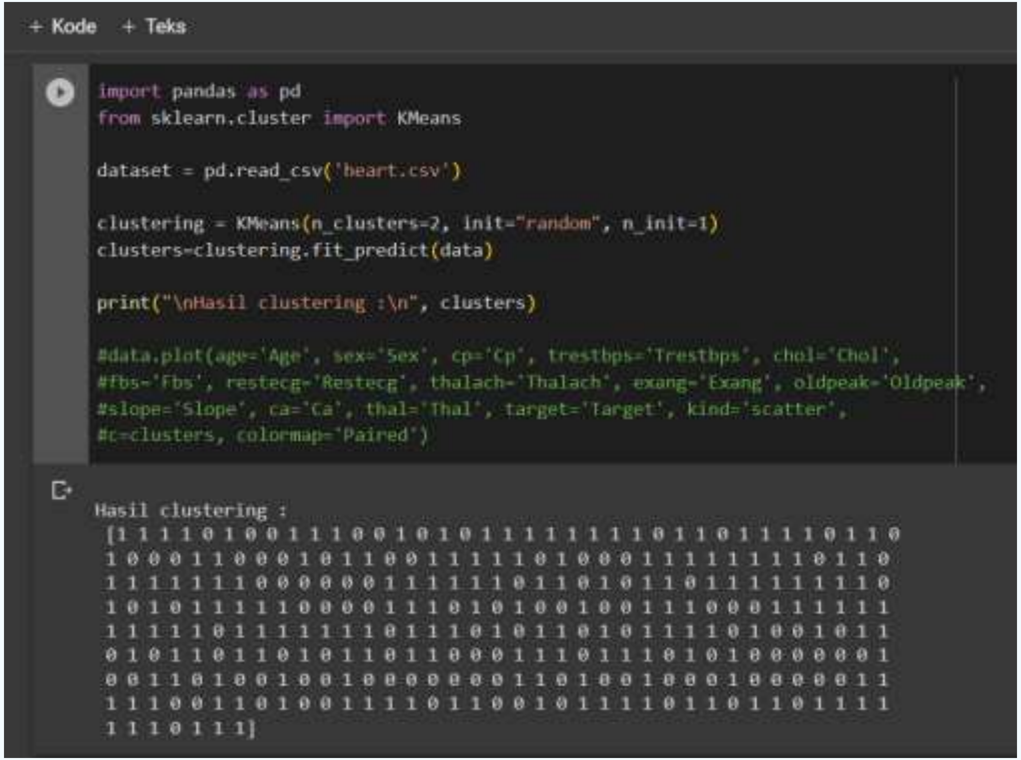
2. data← normalisasi dengan min-max(0-1)



```
import numpy as np
from sklearn.preprocessing import MinMaxScaler

dataset = pd.read_csv('heart.csv')
sc = MinMaxScaler(feature_range=(0, 1))
train_data = np.array(dataset)[:,1:-1]
data = sc.fit_transform (train_data)
print(data)
```

```
[[1.         1.         0.48113208 ... 0.         0.         0.33333333]
 [1.         0.66666667 0.33962264 ... 0.         0.         0.66666667]
 [0.         0.33333333 0.33962264 ... 1.         0.         0.66666667]
 ...
 [1.         0.         0.47169811 ... 0.5        0.5        1.        ]
 [1.         0.         0.33962264 ... 0.5        0.25       1.        ]
 [0.         0.33333333 0.33962264 ... 0.5        0.25       0.66666667]]
```

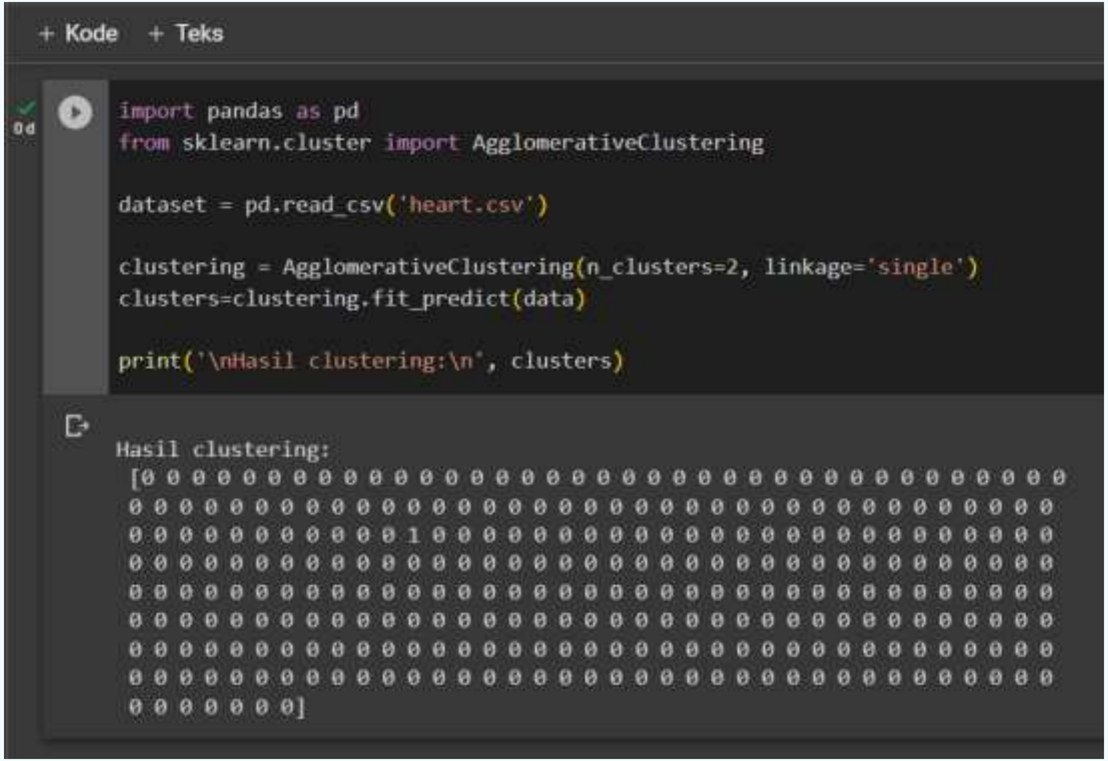
3. cluster ← lakukan clustering pada data dengan menggunakan K-Means (k=2)



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

dataset = pd.read_csv('heart.csv')

clustering = KMeans(n_clusters=2, init="random", n_init=1)
clusters=clustering.fit_predict(data)

print("\nHasil clustering :\n", clusters)

#data.plot(age='Age', sex='Sex', cp='Cp', trestbps='Trestbps', chol='Chol',
#fbs='Fbs', restecg='Restecg', thalach='Thalach', exang='Exang', oldpeak='Oldpeak',
#slope='Slope', ca='Ca', thal='Thal', target='Target', kind='scatter',
#c=clusters, colormap='Paired')
```

```
Hasil clustering :
 [1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0
 1 0 0 0 1 1 0 0 0 1 0 1 1 0 0 1 1 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 0
 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 0 1 0 1 1 0 1 1 1 1 1 1 1 1 0
 1 0 1 0 1 1 1 1 1 0 0 0 0 1 1 1 0 1 0 1 0 0 1 0 0 1 1 1 0 0 0 1 1 1 1 1 1
 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 0 0 1 0 1 1
 0 1 0 1 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 1 1 1 0 1 1 1 0 1 0 1 0 0 0 0 0 0 1
 0 0 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 1
 1 1 1 0 0 1 1 0 1 0 0 1 1 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1
 1 1 1 0 1 1 1]
```

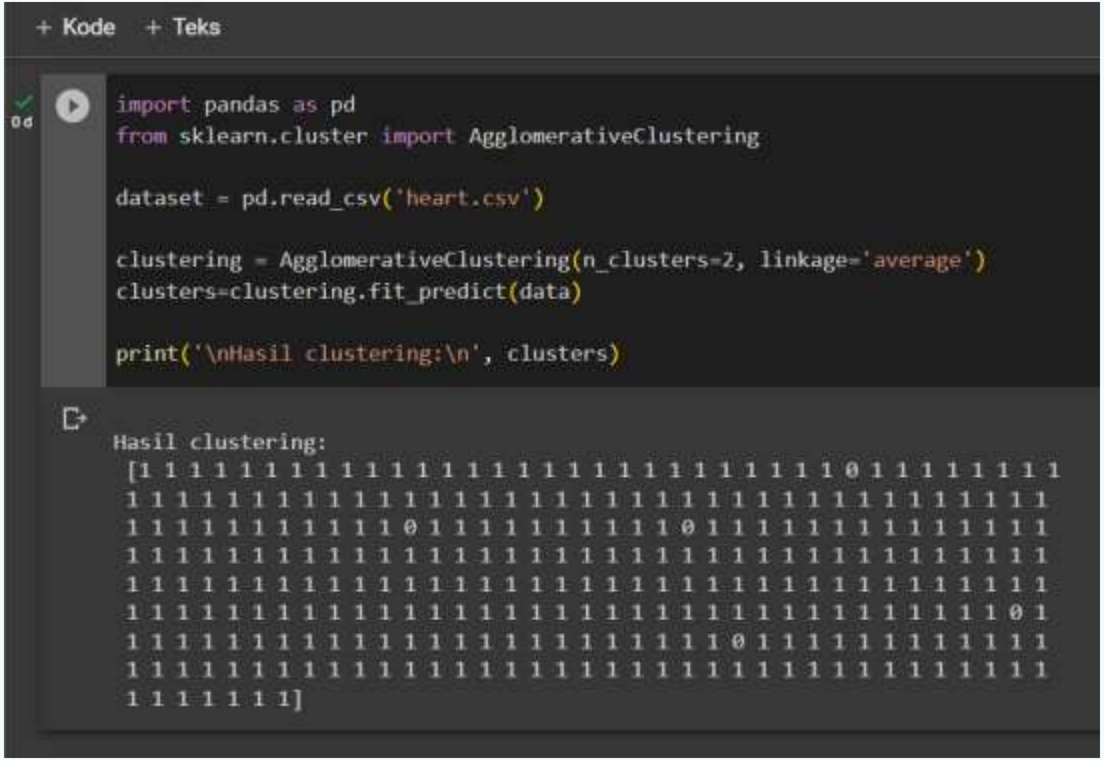
4. cluster ← lakukan clustering pada data dengan Single, Average, Complete Linkage (k=2)



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

dataset = pd.read_csv('heart.csv')

clustering = AgglomerativeClustering(n_clusters=2, linkage='single')
clusters=clustering.fit_predict(data)

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

```
Hasil clustering:
 [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 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 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 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 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 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]
```

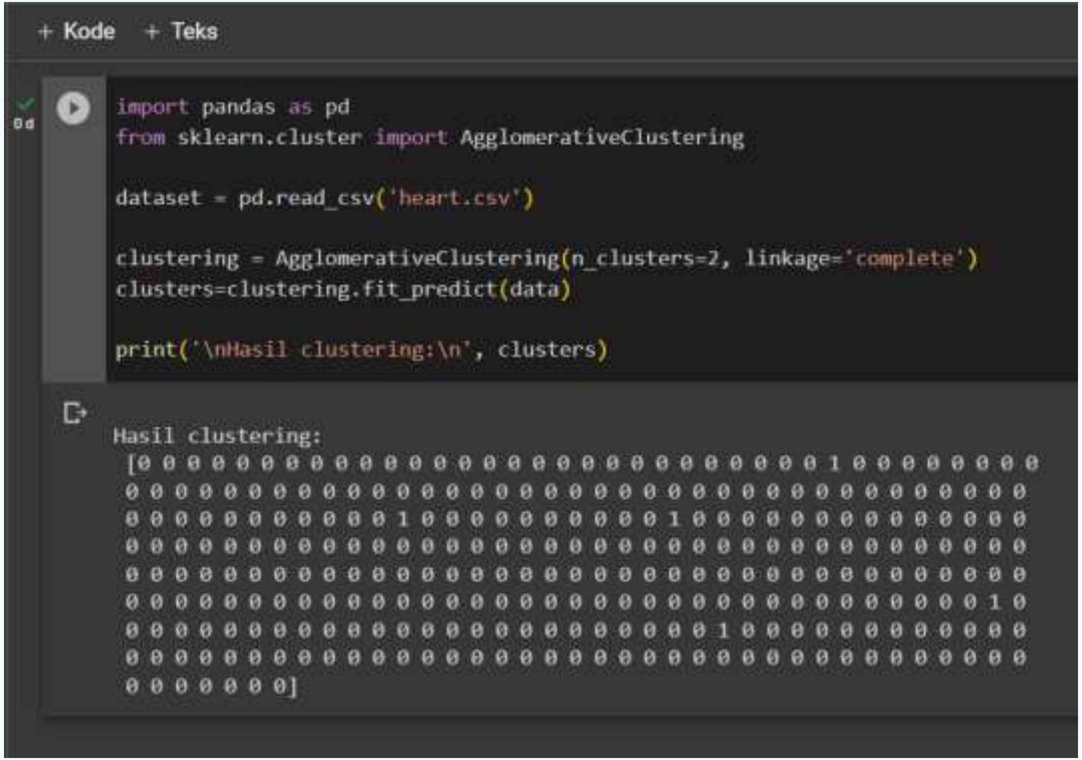
4. cluster ← lakukan clustering pada data dengan Single, Average, Complete Linkage (k=2)



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

dataset = pd.read_csv('heart.csv')

clustering = AgglomerativeClustering(n_clusters=2, linkage='average')
clusters=clustering.fit_predict(data)

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

```
Hasil clustering:
 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1]
```

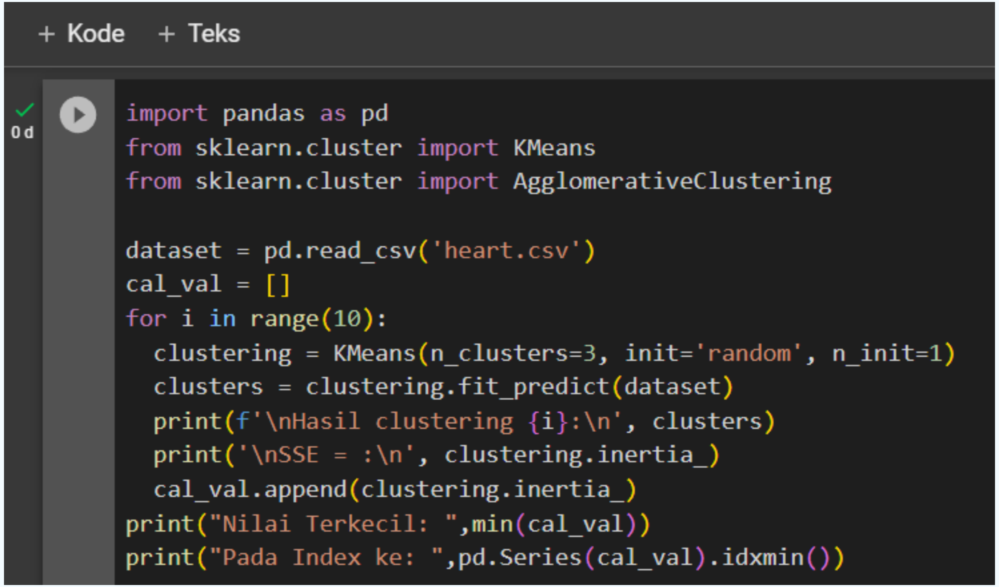
4. cluster ← lakukan clustering pada data dengan Single, Average, Complete Linkage (k=2)



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

dataset = pd.read_csv('heart.csv')

clustering = AgglomerativeClustering(n_clusters=2, linkage='complete')
clusters=clustering.fit_predict(data)

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

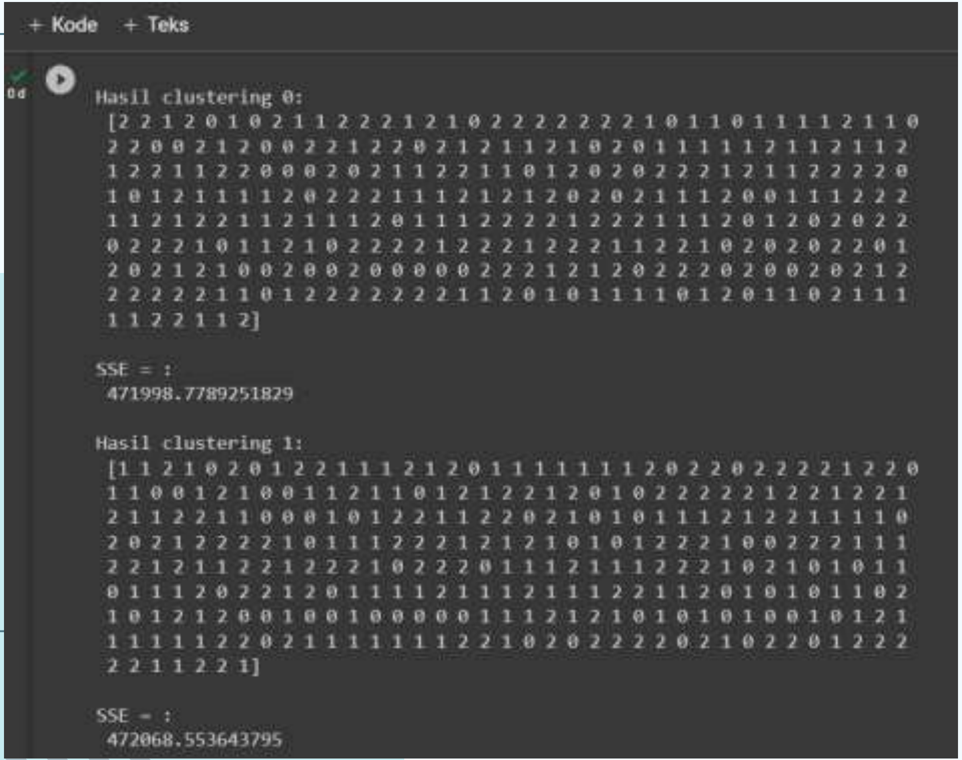
```
Hasil clustering:
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 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 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 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]
```

5. Lakukan untuk setiap jumlah atribut dilangkah ke-4 : cluster_i[1-10], cluster_val[1-10] ← Lakukan clustering pada data dengan atribut yang paling berpengaruh dengan K-Means, dengan k=3, sebanyak 10 kali. Setiap kali selesai clustering, lakukan cluster analysis dengan SSE



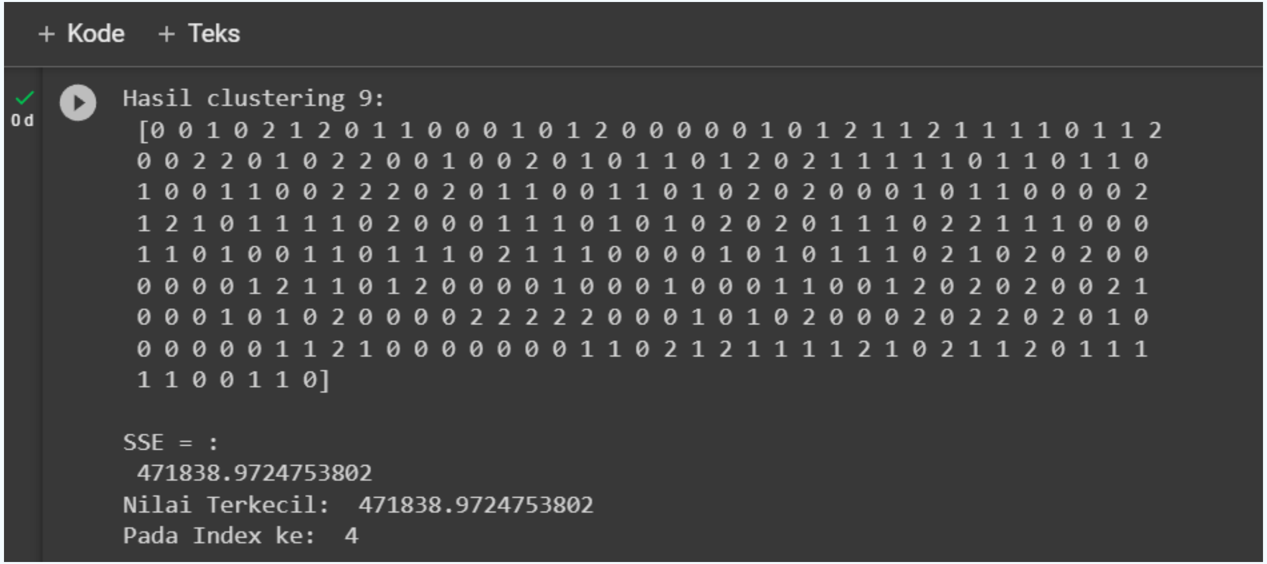
```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering

dataset = pd.read_csv('heart.csv')
cal_val = []
for i in range(10):
  clustering = KMeans(n_clusters=3, init='random', n_init=1)
  clusters = clustering.fit_predict(dataset)
  print(f'\nHasil clustering {i}:\n', clusters)
  print('\nSSE = :\n', clustering.inertia_)
  cal_val.append(clustering.inertia_)
print("Nilai Terkecil: ",min(cal_val))
print("Pada Index ke: ",pd.Series(cal_val).idxmin())
```

5. Hasil Clusterring Terhadap Indeks ke {i} dan SSE Pada Indeks ke {i}



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

SSE = :
 471998.7789251829

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

SSE = :
 472068.553643795
```

6. Cluster ← ambil cluster_i yang mempunyai cluster_val terkecil



```
Hasil clustering 9:
 [0 0 1 0 2 1 2 0 1 1 0 0 0 1 0 1 2 0 0 0 0 0 1 0 1 2 1 1 2 1 1 1 1 0 1 1 2
 0 0 2 2 0 1 0 2 2 0 0 1 0 0 2 0 1 0 1 1 0 1 2 0 2 1 1 1 1 1 0 1 1 0 1 1 0
 1 0 0 1 1 0 0 2 2 2 0 2 0 1 1 0 0 1 1 0 1 0 2 0 2 0 0 0 1 0 1 1 0 0 0 0 2
 1 2 1 0 1 1 1 1 0 2 0 0 0 1 1 1 0 1 0 1 0 2 0 2 0 1 1 1 0 2 2 1 1 1 0 0 0
 1 1 0 1 0 0 1 1 0 1 1 1 0 2 1 1 1 0 0 0 0 1 0 1 0 1 1 1 0 2 1 0 2 0 2 0 0
 0 0 0 0 1 2 1 1 0 1 2 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 1 2 0 2 0 2 0 0 2 1
 0 0 0 1 0 1 0 2 0 0 0 0 2 2 2 2 2 0 0 0 1 0 1 0 2 0 0 0 2 0 2 2 0 2 0 1 0
 0 0 0 0 0 1 1 2 1 0 0 0 0 0 0 0 1 1 0 2 1 2 1 1 1 1 2 1 0 2 1 1 2 0 1 1 1
 1 1 0 0 1 1 0]

SSE = :
 471838.9724753802
Nilai Terkecil:  471838.9724753802
Pada Index ke:  4
```