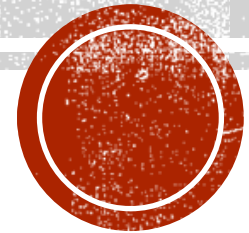


M11 KNOWLEDGE DISCOVERY

Nama : Ni Putu Devira Ayu Martini

Kelas : Teknik Elektro

NRP : 1120800012



PROGRAM MENGGUNAKAN JAVA



PROGRAM JAVA

```
package clusteringoutlayer2;
import ALI.*;
import java.util.Map;

VectorLib vlib = new VectorLib();
ClusteringLib cluslib = new ClusteringLib();
double[][] data = {{80, 60, 75, 73, 81, 66}, {70, 84, 88, 90, 65, 60},
{60, 40, 55, 58, 47, 49},{60, 65, 60, 70, 68, 72},{60, 35,40, 20, 56, 57},
{95, 98, 85, 87, 93,95},{55, 70, 53, 64, 74, 77},{50, 53, 57, 63, 58, 40},{62, 64, 53, 58, 40, 45}};
int k=2;
int[] cluster=cluslib.Clustering("centroid", data, k);
double[][] centroid=cluslib.getCentroid(data, cluster);

int[] outlier=vlib.initArray(data.length, 0);
double[] Davg=vlib.initArray(k, 0.0);
int[] n_cluster=vlib.initArray(k, 0);
double[] Dmin=new double[k];
double alpha=1.1;
```



PROGRAM JAVA

```
double[] dist_to_centroid=new double[data.length];
for (int i=0; i<cluster.length; i++) {
    dist_to_centroid[i]=vlib.getDistance(data[i], centroid[cluster[i]]);
    Davg[cluster[i]]=Davg[cluster[i]] + dist_to_centroid[i];
    n_cluster[cluster[i]] = n_cluster[cluster[i]] + 1;
}
```

```
for (int i=0; i<k; i++) {
    Davg[i]= Davg[i] / n_cluster[i];
    Dmin[i]=Davg[i] * alpha;
}
for (int i=0; i<cluster.length; i++)
    if (dist_to_centroid[i]>Dmin[cluster[i]])
        outlier[i]=1;
```

```
vlib.view(outlier);
```

run:

0 1 0 0 1 0 0 0 0

BUILD SUCCESSFUL (total time: 0 seconds)

Data Outlayer



PERUBAHAN PARAMETER

- Alpha = 1.2

```
run:
0  1  0  0  1  0  0  0  0

BUILD SUCCESSFUL (total time: 1 second)
```

- Alpha = 1.6

```
run:
0  0  0  0  1  0  0  0  0

BUILD SUCCESSFUL (total time: 0 seconds)
```

- Alpha = 1.8

```
run:
0  0  0  0  0  0  0  0  0

BUILD SUCCESSFUL (total time: 0 seconds)
```



PROGRAM MENGGUNAKAN PYTHON



```
In [15]: import pandas as pd
import numpy as np
!pip install -U scikit-learn
from sklearn.ensemble import IsolationForest

data={'Keaktifan1': [80,70,60,60,60,95,55,50,62],
'Keaktifan2': [60,84,40,65,35,98,70,53,64],
'Keaktifan3': [75,88,55,60,40,85,53,57,53],
'Keaktifan4': [73,90,58,70,20,87,64,63,58],
'Keaktifan5': [81,65,47,68,56,93,74,58,40],
'Keaktifan6': [66,60,49,72,57,95,77,40,45]}

df=pd.DataFrame(data, columns=['Keaktifan1','Keaktifan2','Keaktifan3','Keaktifan4','Keaktifan5','Keaktifan6'])

clf = IsolationForest(contamination=1)
pred = clf.fit_predict(df)

df['Outlier']=pred.reshape(-1,1)
print(df)
```

	Keaktifan1	Keaktifan2	Keaktifan3	Keaktifan4	Keaktifan5	Keaktifan6	\
0	80	60	75	73	81	66	
1	70	84	88	90	65	60	
2	60	40	55	58	47	49	
3	60	65	60	70	68	72	
4	60	35	40	20	56	57	
5	95	98	85	87	93	95	
6	55	70	53	64	74	77	
7	50	53	57	63	58	40	
8	62	64	53	58	40	45	

	Outlier
0	-1
1	-1
2	-1
3	1
4	-1
5	-1
6	-1
7	-1
8	-1



ANALISA

- Pada praktikum kali ini membahas Deteksi pada data Outlayer. Yang dimaksud data Outlayer adalah data yang habit/titiknya jauh dari segerombolan data lainnya. Data outlayer ini sebenarnya bisa sangat penting pada pengaplikasian nyatanya.
- Untuk mendeteksi data Outlayer bisa menggunakan Teknik Clustering, yaitu dengan mengelompokkan data kedalam suatu cluster terlebih dahulu, yang nantinya didapatkan centroid pada setiap cluster (sampai dengan tahap ini, menggunakan teknik K-Means), kemudian dicarilah D_{avg} /jarak rata-rata dari setiap data ke centroid pada setiap cluster, lalu menghitung jarak minimum/ D_{min} untuk data dapat dikatakan tidak outlayer, yaitu dengan rumus:

$$D_{min} = \alpha * D_{avg}$$

dimana, $\alpha > 1$.

- Semakin besar nilai α , maka penentuan banyaknya data outlayer-nya semakin kecil, dan sebaliknya. (berbanding terbalik). Hal ini dapat dilihat pada slide “perubahan parameter”

