

# *Tugas 2 Machine Learning* *Unsupervised - KMeans*



Oleh :

Mochamad Agusta Naofal H      1301154378      IF 39-14

## Analisa Masalah

Masalah yang dijadikan studi kasus pada Tugas 2 Machine Learning ini terkait dengan Pembelajaran unsupervised learning menggunakan metode KMeans, diberikan 2 file untuk dijadikan bahan sebagai data training set dan tes set, dengan jumlah data 688 data untuk training dan 100 untuk data tes.

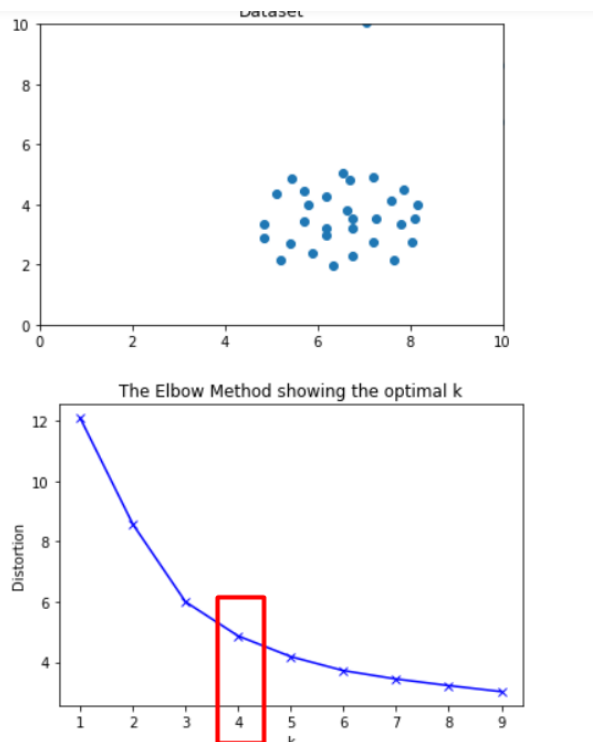
Sesuai dengan jenis metodenya yaitu metode Unsupervised jadi di dalam data train tidak diberikan label, dan disini lah masalah yang harus di selesaikan adalah menentukan nilai K ( atau banyaknya sebuah centroid atau klasifikasi ) yang optimal, yang bisa memisahkan semua data dengan benar sesuai dengan similaritynya.

## Desain

Konsep dasar dari metode KMeans adalah untuk menentukan kelas sebuah data, atau mengkalsifikasikan sebuah data dengan menggunakan nilai similarity antar data, jika similarity sebuah data dengan data lainnya memiliki nilai yang kecil maka, bisa dianggap bahwa sebuah data tersebut termasuk kedalam golongan data yang lain.

Kemudian untuk membuat metode KMeans ini optimal maka ada beberapa parameter yang harus ditentukan sebelumnya, salah satunya adalah nilai K atau cluster yang di inginkan.

Dalam Penyelesaian penentuan jumlah K yang optimal, saya menggunakan bantuan dengan menggunakan elbow method yang berfungsi untuk menentukan nilai yang optimal dari data train yang diberikan, berikut hasilnya ( untuk sourcecode sudah disertakan kedalam github).



Di sini yang saya jadikan sebagai **nilai K** adalah 4, karena saya anggap bahwa nilai 4 merupakan nilai yang paling optimal dari nilai K yang lain karena, pada titik dengan nilai k=4 sebelumnya memiliki penurunan distortion yang cukup drastis dibandingkan dengan titik yang lain.

Sumber : <https://pythonprogramminglanguage.com/kmeans-elbow-method/>

```
#tentukan nilai k dan centroid awal  
nilaiK = 4
```

**Penentuan titik awal centroid** saya menentukan dengan cara random dari titik titik yang terdapat pada data training set, dengan memilih ke empat nilai centroid secara random.

```
# tentukan nilai centroid default/awal dengan memilih index dari array data  
def isiCentroid(nilaiK):  
    for i in range(0,nilaiK):  
        centroid.append(listdata[random.randrange(0,len(listdata))])  
        print("centroid ke -",i, " ",centroid[i].x,"",centroid[i].y)
```

Untuk penentuan nilai jarak terdekat menggunakan metode euclidian

```
# pemrosesan pencarian jarak dengan rumus euclidian  
def fungsiEuclidian(titikA, titikB):  
    temp = math.sqrt(((titikA.x - titikB.x) ** 2) + ((titikA.y - titikB.y) ** 2)) #penghitungan dilakukan dengan mengurangi koord  
    return temp
```

Kemudian untuk parameter selanjutnya untuk menentukan **batasan looping atau pencarian dari nilai centroid finalnya** saya membuat kondisi dimana untuk melakukan pengecekan terhadap nilai centroid lama dan centroid baru apakah konvergen atau tidak. ( tidak mengalami perubahan )

```
#metode untuk mengecek sebuah centroid konvergen atau tidak dengan melakukan pencocokan dengan titik centroid sebelumnya  
def konvergenCentroid():  
    kebenaran = 0  
    for i in range(0,len(centroid)):  
        if( (centroid[i].x == cenTemp[i].x) and (centroid[i].y == cenTemp[i].y)):  
            kebenaran += 1  
    if(kebenaran == nilaiK):  
        return True  
    else:  
        return False
```

Desain Algoritma dan tahap – tahapn yang saya gunakan dalam program sebagai berikut.

1. Pesiapan data dengan melakukan pengimporan data kedalam progam
2. Penentuan nilai K dan juga nilai untuk setiap centroid nya dengan random dari nilai training set
3. Penhitugan jarak dengan menggunakan rumus euclidian antara semua titik dan titik centroid
4. Kemudian penghitungan rata rata nilai cnetroid untuk dilakukan update
5. Jika nilai centroid mengalami perubahan dengan nilai centroid awal ( belum konvergen ) maka lakukan update centroid
6. Jika sudah tidak mengalami perubahan maka program berhenti dan menemukan nilai centroid final

## 7. Lakukan klasifikasi teset dengan nilai cnetroid yang sudah final

```
makeplot(xtrain,ytrain) #pembuatan plot untuk data train
makeplot(xtest,ytest) #pembuatan plot untuk data test

isiCentroid(nilaiK) #inisialisasi atau pengisian centroid default dengan nilai random
iterasi = 1
while True:
    # printisiCentroid(iterasi)
    # print(iterasi)
    hitungJarak() #untuk menentukan anggota untuk tiap centroid, berdasarkan jarak euclidian terkecil, di dalam hitungja

    isiCentroidLama() #mengisi nilai centroid temporary dengan nilai centroid sebelumnya
    isiCentroidUpdate() #mengisi nilai centroid baru dengan nilai centroid yang baru

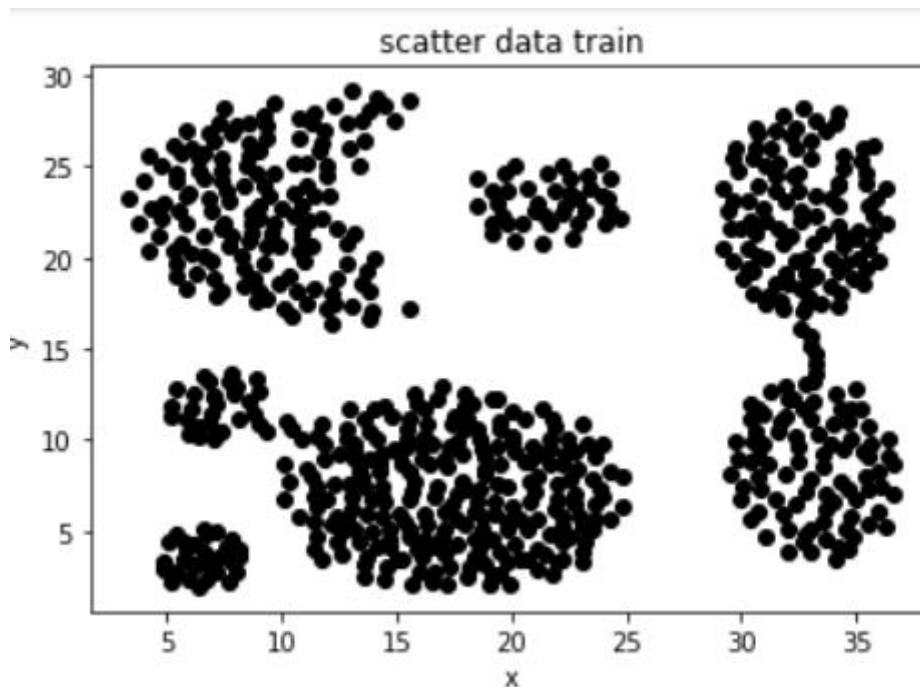
    iterasi += 1

    if konvergenCentroid(): #fungsi untuk dilakukan pengecekan konvergen kah nilai nilai centroid yang telah di hitung
        break

printisiCentroid(iterasi) #mengeluarkan nilai centroid yang paling baru
klasifikasiTest() #proses klasifikasi dari data test set, berdasarkan nilai centroid yang paling baru
printHasilTest() #mengeluarkan hasil klasifikasi dari test
```

## Eksperimen

Berdasarkan data train dalam scatter plot



Dan juga hasil yang di dapatkan menggunakan nilai K = 4, data sudah terklasifikasi dengan persebaran yang lumayan besar, dan selisih rata rata centroid yang di dapatkan juga besar

Pecobaan 1 :

```
ini update centroid ke 5
nilai centri 1 yang baru. x= 32.845876288659795 | y= 8.814432989690726
nilai centri 2 yang baru. x= 15.445328719723193 | y= 7.096712802768166
nilai centri 3 yang baru. x= 9.765853658536582 | y= 22.339024390243903
nilai centri 4 yang baru. x= 30.503985507246366 | y= 22.35978260869564
```

Hasil dari klasifikasi data test set dengan centroid random dan nilai K = 4

```
0 . ( 18.75 , 22.95 ) kelas = 3
1 . ( 21.45 , 21.45 ) kelas = 4
2 . ( 20.5 , 22.85 ) kelas = 4
3 . ( 20.65 , 24.3 ) kelas = 4
4 . ( 21.7 , 23.8 ) kelas = 4
5 . ( 23.1 , 21.7 ) kelas = 4
6 . ( 13.35 , 28.45 ) kelas = 3
7 . ( 12.4 , 27.85 ) kelas = 3
8 . ( 12.2 , 28.65 ) kelas = 3
9 . ( 12.9 , 26.5 ) kelas = 3
10 . ( 11.15 , 28.7 ) kelas = 3
11 . ( 10.5 , 28.35 ) kelas = 3
12 . ( 10.25 , 27.25 ) kelas = 3
13 . ( 12.6 , 24.05 ) kelas = 3
14 . ( 10.05 , 25.95 ) kelas = 3
15 . ( 8.5 , 27.05 ) kelas = 3
16 . ( 7.55 , 26.3 ) kelas = 3
17 . ( 9.4 , 25.55 ) kelas = 3
18 . ( 10.55 , 24.35 ) kelas = 3
19 . ( 5.4 , 25.25 ) kelas = 3
20 . ( 4.3 , 24.0 ) kelas = 3
21 . ( 6.1 , 22.6 ) kelas = 3
22 . ( 6.4 , 21.95 ) kelas = 3
23 . ( 8.45 , 17.2 ) kelas = 3
24 . ( 12.3 , 22.75 ) kelas = 3
25 . ( 9.95 , 19.8 ) kelas = 3
26 . ( 12.0 , 20.0 ) kelas = 3
27 . ( 11.4 , 19.25 ) kelas = 3
28 . ( 15.2 , 18.2 ) kelas = 3
29 . ( 31.9 , 4.4 ) kelas = 1
30 . ( 32.8 , 6.0 ) kelas = 1
31 . ( 35.45 , 4.1 ) kelas = 1
32 . ( 32.3 , 7.65 ) kelas = 1
33 . ( 31.8 , 9.45 ) kelas = 1
34 . ( 34.7 , 8.0 ) kelas = 1
35 . ( 34.55 , 8.85 ) kelas = 1
36 . ( 31.2 , 12.0 ) kelas = 1
37 . ( 33.1 , 12.75 ) kelas = 1
38 . ( 35.8 , 11.55 ) kelas = 1
39 . ( 34.05 , 13.05 ) kelas = 1
40 . ( 11.05 , 9.1 ) kelas = 2
41 . ( 14.3 , 12.45 ) kelas = 2
42 . ( 14.45 , 10.75 ) kelas = 2
43 . ( 12.35 , 8.45 ) kelas = 2
44 . ( 12.4 , 7.1 ) kelas = 2
45 . ( 15.1 , 10.25 ) kelas = 2
46 . ( 16.75 , 11.5 ) kelas = 2
47 . ( 15.05 , 7.8 ) kelas = 2
48 . ( 10.6 , 5.05 ) kelas = 2
49 . ( 13.9 , 5.3 ) kelas = 2
50 . ( 12.75 , 3.0 ) kelas = 2
51 . ( 15.15 , 7.1 ) kelas = 2
52 . ( 15.25 , 2.7 ) kelas = 2
53 . ( 15.7 , 6.35 ) kelas = 2
54 . ( 17.3 , 4.8 ) kelas = 2
55 . ( 18.15 , 1.95 ) kelas = 2
```

56 . ( 20.45 , 2.8 ) kelas = 2  
57 . ( 18.65 , 4.75 ) kelas = 2  
58 . ( 16.6 , 7.95 ) kelas = 2  
59 . ( 20.8 , 4.7 ) kelas = 2  
60 . ( 20.55 , 5.75 ) kelas = 2  
61 . ( 17.5 , 8.25 ) kelas = 2  
62 . ( 18.0 , 8.55 ) kelas = 2  
63 . ( 20.05 , 6.95 ) kelas = 2  
64 . ( 24.15 , 4.55 ) kelas = 2  
65 . ( 20.0 , 8.2 ) kelas = 2  
66 . ( 16.95 , 10.35 ) kelas = 2  
67 . ( 19.0 , 11.65 ) kelas = 2  
68 . ( 20.7 , 10.65 ) kelas = 2  
69 . ( 20.95 , 10.2 ) kelas = 2  
70 . ( 21.75 , 8.2 ) kelas = 2  
71 . ( 23.0 , 7.35 ) kelas = 2  
72 . ( 23.7 , 8.85 ) kelas = 2  
73 . ( 5.15 , 3.45 ) kelas = 2  
74 . ( 4.95 , 4.05 ) kelas = 2  
75 . ( 7.1 , 4.3 ) kelas = 2  
76 . ( 8.5 , 3.25 ) kelas = 2  
77 . ( 32.45 , 16.75 ) kelas = 4  
78 . ( 30.55 , 18.8 ) kelas = 4  
79 . ( 31.55 , 19.65 ) kelas = 4  
80 . ( 33.7 , 17.0 ) kelas = 4  
81 . ( 31.75 , 20.25 ) kelas = 4  
82 . ( 31.55 , 22.2 ) kelas = 4  
83 . ( 30.95 , 24.15 ) kelas = 4  
84 . ( 33.65 , 21.9 ) kelas = 4  
85 . ( 33.8 , 20.4 ) kelas = 4  
86 . ( 36.35 , 20.6 ) kelas = 4  
87 . ( 34.6 , 22.05 ) kelas = 4  
88 . ( 34.9 , 23.5 ) kelas = 4  
89 . ( 33.6 , 23.9 ) kelas = 4  
90 . ( 33.7 , 24.85 ) kelas = 4  
91 . ( 30.25 , 24.3 ) kelas = 4  
92 . ( 31.25 , 27.85 ) kelas = 4  
93 . ( 33.85 , 26.05 ) kelas = 4  
94 . ( 34.65 , 26.85 ) kelas = 4  
95 . ( 9.7 , 12.1 ) kelas = 2  
96 . ( 5.7 , 12.25 ) kelas = 3  
97 . ( 7.85 , 11.85 ) kelas = 2  
98 . ( 7.65 , 11.1 ) kelas = 2  
99 . ( 8.3 , 10.55 ) kelas = 2

Percobaan random 2 :

```
centroid ke - 0    19.0  2.1
centroid ke - 1    22.9  8.45
centroid ke - 2    35.25 26.0
centroid ke - 3    14.35  5.75
ini update centroid ke- 12
nilai centroid 1  yang baru. x= 15.506794425087117 | y= 7.059756097560975
nilai centroid 2  yang baru. x= 32.845876288659795 | y= 8.814432989690726
nilai centroid 3  yang baru. x= 30.503985507246366 | y= 22.35978260869564
nilai centroid 4  yang baru. x= 9.728012048192769 | y= 22.219277108433737
```

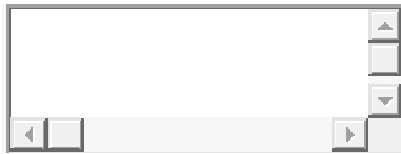
```
0 . ( 18.75 , 22.95 ) kelas = 4
1 . ( 21.45 , 21.45 ) kelas = 3
2 . ( 20.5 , 22.85 ) kelas = 3
3 . ( 20.65 , 24.3 ) kelas = 3
4 . ( 21.7 , 23.8 ) kelas = 3
5 . ( 23.1 , 21.7 ) kelas = 3
6 . ( 13.35 , 28.45 ) kelas = 4
7 . ( 12.4 , 27.85 ) kelas = 4
8 . ( 12.2 , 28.65 ) kelas = 4
9 . ( 12.9 , 26.5 ) kelas = 4
10 . ( 11.15 , 28.7 ) kelas = 4
11 . ( 10.5 , 28.35 ) kelas = 4
12 . ( 10.25 , 27.25 ) kelas = 4
13 . ( 12.6 , 24.05 ) kelas = 4
14 . ( 10.05 , 25.95 ) kelas = 4
15 . ( 8.5 , 27.05 ) kelas = 4
16 . ( 7.55 , 26.3 ) kelas = 4
17 . ( 9.4 , 25.55 ) kelas = 4
18 . ( 10.55 , 24.35 ) kelas = 4
19 . ( 5.4 , 25.25 ) kelas = 4
20 . ( 4.3 , 24.0 ) kelas = 4
21 . ( 6.1 , 22.6 ) kelas = 4
22 . ( 6.4 , 21.95 ) kelas = 4
23 . ( 8.45 , 17.2 ) kelas = 4
24 . ( 12.3 , 22.75 ) kelas = 4
25 . ( 9.95 , 19.8 ) kelas = 4
26 . ( 12.0 , 20.0 ) kelas = 4
27 . ( 11.4 , 19.25 ) kelas = 4
28 . ( 15.2 , 18.2 ) kelas = 4
29 . ( 31.9 , 4.4 ) kelas = 2
30 . ( 32.8 , 6.0 ) kelas = 2
31 . ( 35.45 , 4.1 ) kelas = 2
32 . ( 32.3 , 7.65 ) kelas = 2
33 . ( 31.8 , 9.45 ) kelas = 2
34 . ( 34.7 , 8.0 ) kelas = 2
35 . ( 34.55 , 8.85 ) kelas = 2
36 . ( 31.2 , 12.0 ) kelas = 2
37 . ( 33.1 , 12.75 ) kelas = 2
```

38 . ( 35.8 , 11.55 ) kelas = 2  
39 . ( 34.05 , 13.05 ) kelas = 2  
40 . ( 11.05 , 9.1 ) kelas = 1  
41 . ( 14.3 , 12.45 ) kelas = 1  
42 . ( 14.45 , 10.75 ) kelas = 1  
43 . ( 12.35 , 8.45 ) kelas = 1  
44 . ( 12.4 , 7.1 ) kelas = 1  
45 . ( 15.1 , 10.25 ) kelas = 1  
46 . ( 16.75 , 11.5 ) kelas = 1  
47 . ( 15.05 , 7.8 ) kelas = 1  
48 . ( 10.6 , 5.05 ) kelas = 1  
49 . ( 13.9 , 5.3 ) kelas = 1  
50 . ( 12.75 , 3.0 ) kelas = 1  
51 . ( 15.15 , 7.1 ) kelas = 1  
52 . ( 15.25 , 2.7 ) kelas = 1  
53 . ( 15.7 , 6.35 ) kelas = 1  
54 . ( 17.3 , 4.8 ) kelas = 1  
55 . ( 18.15 , 1.95 ) kelas = 1  
56 . ( 20.45 , 2.8 ) kelas = 1  
57 . ( 18.65 , 4.75 ) kelas = 1  
58 . ( 16.6 , 7.95 ) kelas = 1  
59 . ( 20.8 , 4.7 ) kelas = 1  
60 . ( 20.55 , 5.75 ) kelas = 1  
61 . ( 17.5 , 8.25 ) kelas = 1  
62 . ( 18.0 , 8.55 ) kelas = 1  
63 . ( 20.05 , 6.95 ) kelas = 1  
64 . ( 24.15 , 4.55 ) kelas = 1  
65 . ( 20.0 , 8.2 ) kelas = 1  
66 . ( 16.95 , 10.35 ) kelas = 1  
67 . ( 19.0 , 11.65 ) kelas = 1  
68 . ( 20.7 , 10.65 ) kelas = 1  
69 . ( 20.95 , 10.2 ) kelas = 1  
70 . ( 21.75 , 8.2 ) kelas = 1  
71 . ( 23.0 , 7.35 ) kelas = 1  
72 . ( 23.7 , 8.85 ) kelas = 1  
73 . ( 5.15 , 3.45 ) kelas = 1  
74 . ( 4.95 , 4.05 ) kelas = 1  
75 . ( 7.1 , 4.3 ) kelas = 1  
76 . ( 8.5 , 3.25 ) kelas = 1  
77 . ( 32.45 , 16.75 ) kelas = 3  
78 . ( 30.55 , 18.8 ) kelas = 3  
79 . ( 31.55 , 19.65 ) kelas = 3  
80 . ( 33.7 , 17.0 ) kelas = 3  
81 . ( 31.75 , 20.25 ) kelas = 3  
82 . ( 31.55 , 22.2 ) kelas = 3  
83 . ( 30.95 , 24.15 ) kelas = 3  
84 . ( 33.65 , 21.9 ) kelas = 3



```
85 . ( 33.8 , 20.4 ) kelas = 3
86 . ( 36.35 , 20.6 ) kelas = 3
87 . ( 34.6 , 22.05 ) kelas = 3
88 . ( 34.9 , 23.5 ) kelas = 3
89 . ( 33.6 , 23.9 ) kelas = 3
90 . ( 33.7 , 24.85 ) kelas = 3
91 . ( 30.25 , 24.3 ) kelas = 3
92 . ( 31.25 , 27.85 ) kelas = 3
93 . ( 33.85 , 26.05 ) kelas = 3
94 . ( 34.65 , 26.85 ) kelas = 3
95 . ( 9.7 , 12.1 ) kelas = 1
96 . ( 5.7 , 12.25 ) kelas = 4
97 . ( 7.85 , 11.85 ) kelas = 1
98 . ( 7.65 , 11.1 ) kelas = 1
99 . ( 8.3 , 10.55 ) kelas = 1
```

In [ ]:



In [ ]:

