

MODUL

STMIK WIDYA PRATAMA



DATA MINING

Unsupervised Learning



DESKRIPSI MATAKULIAH

CAPAJAN PEMBELAJARAN

DOSEN PENGAMPU

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

(STMIK) WIDYA PRATAMA

CAPAIAN PEMBELAJARAN

MATERI PEMBELAJARAN

1. Perhitungan Python K-Means

Adapun dataset yang digunakan adalah dataset_gizi.xlsx. berikut perintah nya :

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
```

Keterangan kode :

- Import pandas as pd biasa digunakan untuk mengubah dimensi data, membuat tabel, memeriksa data, membaca data dan lain sebagainya.
- Import numpy as np berfungsi untuk memudahkan operasi perhitungan tipe data numeric seperti penjumlahan, perkalian, pengurangan, pemangkatan dan operasi aritmatika lainnya
- Import seaborn as sns, seaborn merupakan library yang digunakan untuk menampilkan visualisasi data
- Import matplotlib.pyplot as plt, memanggil library matplotlib untuk membuat chart atau grafik
- From sklearn.cluster import KMeans, memanggil algoritma KMeans yang berada di dalam Sklearn
- From sklearn.preprocessing import MinMaxScaler, untuk melakukan normalisasi data

```
gizi = pd.read_excel("dataset_gizi.xlsx")
gizi.head()
```

No	Balita ke-	TB	BB
0	1	Balita 1	52.0 5.8
1	2	Balita 2	51.0 5.0
2	3	Balita 3	71.5 8.5
3	4	Balita 4	55.0 5.5
4	5	Balita 5	92.5 6.5

Keterangan kode :

- Membaca dataset yang sudah disediakan dalam format xlsx. Simpan dataset di folder yang sama dengan folder proyek yang sedang dikerjakan. Jika tidak, maka jalur data harus dijelaskan seperti “D/data/dataset_gizi.xlsx”
- Separator berfungsi untuk menjelaskan pemisah pada dataset. Jika menyimpan dataset dalam bentuk csv dan dipisahkan dengan koma, maka pilih seperti pada kode di atas. Jika bukan, maka ubah tanda koma dengan separator yang digunakan seperti ; atau tab
- Head() digunakan untuk menampilkan sebanyak 5 data teratas.

```
gizi.info()
```

```
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
 --- 
 0   No          10 non-null    int64  
 1   Balita ke-  10 non-null    object  
 2   TB          10 non-null    float64 
 3   BB          10 non-null    float64 
 dtypes: float64(2), int64(1), object(1)
 memory usage: 448.0+ bytes
```

Keterangan kode :

- gizi.info, untuk mendapatkan informasi tentang fitur-fitur dataset, tipe data yang digunakan, jumlah tupel dari setiap fitur dan memori yang digunakan.

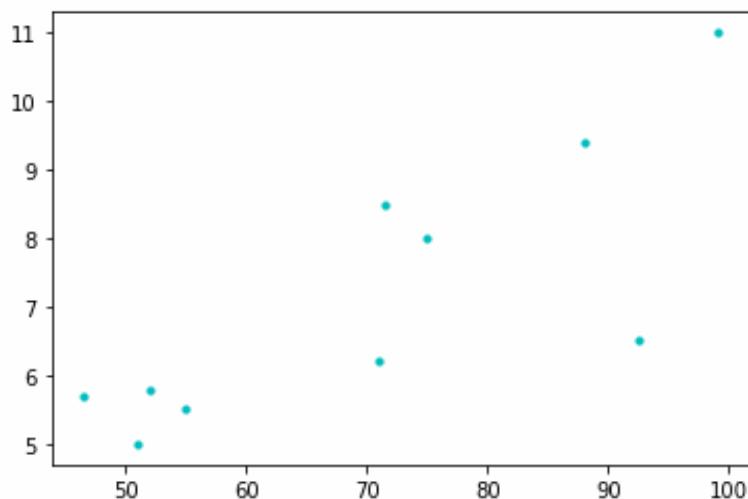
```
gizi_x = gizi.iloc[:, 2:4]
gizi_x.head()
```

	TB	BB
0	52.0	5.8
1	51.0	5.0
2	71.5	8.5
3	55.0	5.5
4	92.5	6.5

Keterangan kode :

- gizi_x = gizi.iloc[:, 2:4], untuk meilih lokasi fitur mana saja yang akan diproses. Pada kasus ini, fitur tinggi badan dan berat badan. [:, 2:4] artinya akan mengambil fitur mulai dari nomor 2 hingga nomor sebelum 4.

```
plt.scatter(gizi.TB, gizi.BB, s = 10, c = "c", marker = "o", alpha = 1)
plt.show()
```



Keterangan kode :

- plt.scatter(gizi.TB, gizi.BB, s =10, c ="c", marker="o", alpha = 1) untuk menampilkan visualisasi dalam bentuk scatter dan memanggil fitur yang akan dijadikan sumbu x dan sumbu y
- plt.show() untuk menampilkan visualisasi scatter plot
- Visualisasi ini penting untuk melihat sebaran data berdasarkan tinggi badan dan berat badan bayi sebelum dilakukan clustering

```
x_array = np.array(gizi_x)
print(x_array)
```

```
[[52.    5.8]
 [51.    5.   ]
 [71.5   8.5]
 [55.    5.5]
 [92.5   6.5]
 [46.5   5.7]
 [75.    8.   ]
 [99.    11.  ]
 [88.    9.4]
 [71.    6.2]]
```

Keterangan kode :

- Mengubah dataframe fitur berat badan dan tinggi badan menjadi array. Berfungsi untuk memudahkan saat normalisasi data di kode selanjutnya
- Print(x_array), menampilkan hasil array

```
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x_array)
x_scaled

array([[0.1047619 , 0.13333333],
       [0.08571429, 0.        ],
       [0.47619048, 0.58333333],
       [0.16190476, 0.08333333],
       [0.87619048, 0.25      ],
       [0.        , 0.11666667],
       [0.54285714, 0.5        ],
       [1.        , 1.        ],
       [0.79047619, 0.73333333],
       [0.46666667, 0.2      ]])
```

Keterangan kode :

- scaler = MinMaxScaler(), mewakilkan fungsi minmaxscaler kepada variabel scaler agar lebih mudah dipanggil
- x_scaled = scaler.fit_transform(x_array), melakukan normalisasi pada tinggi badan dan berat badan dalam bentuk array
- x_scaled, menampilkan hasil normalisasi data

```
kmeans = KMeans(n_clusters = 5, random_state=123)
kmeans.fit(x_scaled)
```

Keterangan kode :

- kmeans = KMeans(n_clusters = 5, random_state=123), n_cluster = 5 artinya akan membuat sebanyak 5 cluster. Random_state = 123 artinya pemilihan data testing tidak akan berubah setiap kali mengatur nilainya dengan 123
- kmeans.fit(x_scaled), proses clustering dengan kmeans terhadap data yang sudah dinormalisasi sebelumnya

```
print(kmeans.cluster_centers_)
gizi["cluster"] = kmeans.labels_
gizi.head()
```

Keterangan kode :

- print(kmeans.cluster_centers_), menampilkan hasil cluster kMeans
- gizi[“cluster”]=kmeans.labels_ menambahkan satu fitur baru bernama cluster yang berisi nilai cluster
- gizi.head(), menampilkan hasil akhir dari clustering dalam bentuk dataframe sebagai berikut

No	Balita ke-	TB	BB	cluster
0	1	Balita 1	52.0	5.8
1	2	Balita 2	51.0	5.0
2	3	Balita 3	71.5	8.5
3	4	Balita 4	55.0	5.5
4	5	Balita 5	92.5	6.5

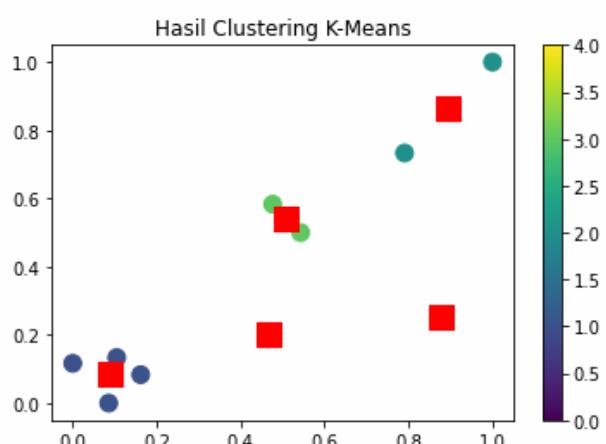
```
output = plt.scatter(x_scaled[:,0], x_scaled[:,1], s = 100, c = gizi.cluster, marker = "o", alpha = 1, )
centers = kmeans.cluster_centers_
plt.scatter(centers[:,0], centers[:,1], c='red', s=200, alpha=1, marker='s')

plt.title("Hasil Clustering K-Means")
plt.colorbar(output)

plt.show()
```

Keterangan kode :

- menampilkan hasil clustering dalam bentuk visualisasi scatter plot sebagai berikut :



- Jelas terlihat pada visualisasi, terdapat 5 centroid cluster yang ditandai dengan kotak persegi merah. Lingkaran merupakan data tiap bayi.
- Cluster 0 dan 4 tidak terlihat datanya karena tertutupi oleh centroid cluster keduanya.



TES FORMATIF