

BAB IV

HASIL DAN PEMBAHASAN

4.1 Representasi Data

Pengumpulan data dilakukan dengan mengajukan surat penelitian dari pihak Penulis kepada pihak SMP Negeri 3 Panyabungan. Data yang dibutuhkan adalah data siswa SMP Negeri 3 Panyabungan dari kelas VII sampai dengan kelas IX. Data yang didapatkan penulis dalam bentuk Excel. Pihak SMP Negeri 3 Panyabungan memberikan dataset tentang nilai-nilai siswa pada tahun ajaran 2022/2023. Penulis mendapatkan nilai siswa SMP Negeri 3 Panyabungan dengan beberapa rangkuman nilai dari Akhlak, Kedisiplinan, Nilai Rata-rata, Absensi, Ekstrakurikuler, Perlombaan. Penulis meringkas variabel Akhlak menjadi C1, Kedisiplinan menjadi C2, Nilai Rata-rata menjadi C3, Absensi menjadi C4, Ekstrakurikuler menjadi C5 dan Perlombaan menjadi C6. Jumlah data siswa sebanyak 242 siswa dari kelas VII sampai kelas IX. Penulis melakukan representasi data dengan 2 step, yaitu *Data Preprocessing* dan *Normalisasi Data*, *Normalisasi data* dilakukan menggunakan *tools Jupiter Notebook*.

4.1.1 Data Preprocessing

Data preprocessing dilakukan untuk menyederhanakan data agar menjadi lebih mudah untuk diproses. Penulis melakukan *Data preprocessing* dengan 1 step yaitu dengan normalisasi data menggunakan *tools Jupiter Notebook*. Normalisasi data dilakukan dengan menggunakan metode MinMax. MinMax ini mengubah nilai data yang memiliki rentang yang berbeda menjadi memiliki rentang yang sama, yaitu antara -1 sampai 1. Untuk mempermudah penulis, *Jupiter Notebook* memiliki library MinMax, yaitu *StandardScaler*. Dengan menggunakan library tersebut, penulis memilih variabel yang akan dinormalisasikan dalam bentuk array. *Jupiter Notebook* akan memproses normalisasi tersebut dengan menyesuaikan dengan data dari variabel tertentu.

Tabel 4. 1 Data Hasil Praprocessing.

| No | NAMA SISWA | C1 | C2 | C3 | C4 | C5 | C6 |
|-----|---------------------|----|----|------|----|----|----|
| 1 | ADELIA FEBRI | 90 | 9 | 89,3 | 5 | 5 | 90 |
| 2 | AGUSTINA RAHMADANI | 90 | 8 | 88,7 | 3 | 5 | 80 |
| 3 | AIDIL ASHARI | 90 | 8 | 89,4 | 5 | 3 | 70 |
| 4 | ALDO | 90 | 8 | 87,2 | 5 | 3 | 80 |
| 5 | ALFAROBI | 90 | 9 | 84,9 | 5 | 5 | 95 |
| 6 | ANGGINA SAPITRI | 80 | 7 | 85,6 | 3 | 2 | 70 |
| 7 | ARDI SAPUTRA | 80 | 7 | 83,3 | 5 | 5 | 70 |
| 8 | AZKIAL FIKRI | 70 | 8 | 85,7 | 5 | 5 | 70 |
| 9 | DIAN PUTRI HASIBUAN | 90 | 9 | 88,9 | 5 | 5 | 90 |
| 10 | KHAIDIR IKHSAN | 80 | 9 | 85,2 | 5 | 5 | 80 |
| 11 | KOTAN FITRI | 80 | 8 | 89,1 | 5 | 3 | 70 |
| 12 | LESTARI AULIYA | 70 | 8 | 84,4 | 3 | 5 | 70 |
| 13 | MARWAH ANGGINA | 90 | 9 | 85,8 | 5 | 5 | 80 |
| 14 | NABILA SAPITRI | 80 | 7 | 87,4 | 3 | 3 | 70 |
| 15 | NUR AIKA | 80 | 8 | 89 | 5 | 3 | 70 |
| 16 | NUR PADILAH | 90 | 8 | 85,4 | 5 | 5 | 90 |
| 17 | NUR SAKINAH | 90 | 8 | 84,3 | 5 | 5 | 70 |
| 18 | NURUL ASIFA | 80 | 7 | 86,1 | 3 | 4 | 70 |
| 19 | REGIANTO | 80 | 7 | 85,6 | 3 | 6 | 70 |
| 242 | SURIANI | 80 | 9 | 89,3 | 5 | 3 | 95 |

Sumber: Data sekolah

4.1.2 Min Max

Pada tahap ini, penulis melakukan tahapan MinMax pada data set menggunakan tools Microsoft Excel, dimana pada setiap atribut mendapatkan nilai Min Maxnya masing-masing. Dimana Pada atribut C1 nilai Minnya 70 dan nilai Maxnya 90 dan dilakukan mencari Min Max nya sampai ke atribut C6.

Dibawah ini adalah rumus mencari nilai Min Max:

$$X_{\text{normalisasi}} = \frac{X - X_{\text{minimum}}}{X_{\text{maksimum}} - X_{\text{minimum}}}$$

Tabel 4. 2 Data Hasil MinMax

| | | | | | |
|----------|-----|----------|------|-----|------|
| 86,50000 | 5,5 | 80,11111 | 4,75 | 4,5 | 87,2 |
| 86,50000 | 4,5 | 79,11111 | 2,75 | 4,5 | 77,2 |
| 86,50000 | 4,5 | 80,11111 | 4,75 | 2, | 67,2 |
| 86,50000 | 4,5 | 78,11111 | 4,75 | 2,5 | 77,2 |
| 86,50000 | 5,5 | 75,11111 | 4,75 | 4,5 | 92,2 |
| 76,50000 | 3,5 | 76,11111 | 2,75 | 1,5 | 67,2 |
| 76,50000 | 3,5 | 74,11111 | 4,75 | 4,5 | 67,2 |
| 66,50000 | 4,5 | 76,11111 | 4,75 | 4,5 | 67,2 |
| 86,50000 | 5,5 | 79,11111 | 4,75 | 4,5 | 87,2 |
| 76,50000 | 5,5 | 76,11111 | 4,75 | 4,5 | 77,2 |

Sumber: Data sekolah

4.1.3 Normalisasi Data (Feature Scaling)

Tahap selanjutnya, penulis melakukan normalisasi data atau *featurescaling*. Tahap ini dilakukan dengan menggunakan *tools Jupiter Notebook*. Tahap ini diperlukan untuk menyamakan rentang nilai data dari masing-masing variabel. Normalisasi yang dilakukan penulis menggunakan *package* yang terdapat pada *Jupiter Notebook* yang menggunakan metode *MinMax Scaler*, yaitu *StandardScaler*. *StandardScaler* ini akan di implementasikan pada 6 variabel yang digunakan, yaitu kolom C1,C2,C3,C4,C5,C6.

Sebelum menormalisasi data, penulis melakukan *import library* yang dibutuhkan. *Library* yang dibutuhkan adalah *pandas*, *numpy*, *matplotlib.pyplot*, dan *StandardScaler*, serta mendeklarasi dan memanggil data seperti pada gambar di bawah ini.

```
import pandas as pd
import numpy as np
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_samples, silhouette_score
from sklearn.neighbors import NearestNeighbors
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

```
df = pd.read_csv("DATASET.csv")
df.head()
```

| | NAMA SISWA | C1 | C2 | C3 | C4 | C5 | C6 |
|---|--------------------|----|----|----|----|----|----|
| 0 | ADELIA FEBRI | 90 | 9 | 89 | 5 | 5 | 90 |
| 1 | AGUSTINA RAHMADANI | 90 | 8 | 89 | 3 | 5 | 80 |
| 2 | AIDILASHARI | 90 | 8 | 89 | 5 | 3 | 70 |
| 3 | ALDO | 90 | 8 | 87 | 5 | 3 | 80 |
| 4 | ALFAROBI | 90 | 9 | 85 | 5 | 5 | 95 |

Gambar 4. 1 Import Library Dan Memanggil Data

Setelah itu, penulis melakukan proses normalisasi data dengan metode *MinMax Scaling* menggunakan library *StandardScaler* yang telah tersedia pada Jupyter Notebook seperti pada gambar di bawah ini.

```
#Proses Normalisasi Data
scaler = StandardScaler()
scaler.fit(df)
data_scaled = scaler.transform(df)
data_scaled
```

```
array([[ 0.85749293,  0.9948548 ,  1.143678 ,  0.72333377,  0.76711247,
         1.34202138],
       [ 0.85749293, -0.30951038,  1.143678 , -1.0926957 ,  0.76711247,
         0.19449585],
       [ 0.85749293, -0.30951038,  1.143678 ,  0.72333377, -1.0077752 ,
        -0.95302967],
```

Gambar 4. 2 Proses normalisasi data dengan StandardScaler.

Penulis mendeklarasikan nilai x sebagai data array dan mengambil nilai/values dari kolom kolom C1,C2,C3,C4,C5,C6. Penulis mendeklarasikan scaler sebagai fungsi *StandardScaler*. Dan selanjutnya, - data_scaler dideklarasikan sebagai data dalam bentuk array hasil dari normalisasi. Hasil dari normalisasi data

dapat dilihat seperti gambar berikut. Hasil tersebut memiliki rentang nilai yang sama, yaitu -1 hingga 1.

```
data_scaled
array([[ 0.96782361,  0.92536392,  1.50139083,  0.81819468,  0.99991831,
         1.30578036],
       [ 0.96782361, -0.51013652,  1.18369636, -0.92632613,  0.99991831,
         0.07621286],
       [ 0.96782361, -0.51013652,  1.55433991,  0.81819468, -0.78591365,
        -1.15335464],
       ...,
       [ 0.96782361,  0.92536392, -1.09311405,  0.81819468, -0.78591365,
         0.07621286],
       [ 0.96782361,  0.92536392, -1.09311405,  0.81819468, -0.78591365,
         0.07621286],
       [-0.46906788,  0.92536392, -1.09311405,  0.81819468, -0.78591365,
         0.07621286]])
```

Gambar 4. 3 Hasil normalisasi data

4.2 Penerapan Algoritma *DBSCAN*

Langkah – Langkah yang dibutuhkan dalam seleksi penentuan siswa terbaik menggunakan rumus algoritma *DBSCAN* sebagai berikut:

$$D_{ij} = \sqrt{\sum_a^p (X_{ia} - X_{ja})^2}$$

1. Tentukan parameter *Eps* dan *MinPts*

***Eps* = 11 dan *MinPts* = 11.**

Iterasi I

Pada iterasi I titik pusat awal ditentukan secara acak dengan demikian titik pusat adalah titik J dengan 1 = 80, 2 = 9, 3 = 85,2, 4 = 5, 5 = 5, 6 = 80. Berikut merupakan perhitungan jarak setiap titik dengan titik pusat.

$$\begin{aligned} 1,10 &= \text{SQRT}((90-80)^2 + (9-9)^2 + (89,3-85,2)^2 + (5-5)^2 + (5-5)^2 + (90-80)^2) \\ &= 14,72446943 \end{aligned}$$

$$2,10 = = \text{SQRT}((90-80)^2 + (8-9)^2 + (88,7-85,2)^2 + (3-5)^2 + (5-5)^2 + (80-80)^2)$$

$$= 10,82820391$$

$$3,10 = = \text{SQRT}((90-80)^2 + (8-9)^2 + (89,4-85,2)^2 + (5-5)^2 + (3-5)^2 + (70-80)^2)$$

$$= 14,92112596$$

$$4,10 = = \text{SQRT}((90-80)^2 + (8-9)^2 + (87,2-85,2)^2 + (5-5)^2 + (3-5)^2 + (80-80)^2)$$

$$= 10,4403066521$$

$$5,10 = = \text{SQRT}((90-80)^2 + (9-9)^2 + (84,9-85,2)^2 + (5-5)^2 + (5-5)^2 + (95-80)^2)$$

$$= 18,03025236$$

$$6,10 = = \text{SQRT}((80-80)^2 + (7-9)^2 + (85,6-85,2)^2 + (3-5)^2 + (2-5)^2 + (70-80)^2)$$

$$= 10,8240473$$

$$7,10 = = \text{SQRT}((80-80)^2 + (7-9)^2 + (83,3-85,2)^2 + (5-5)^2 + (5-5)^2 + (70-80)^2)$$

$$= 10,37352399$$

$$8,10 = = \text{SQRT}((70-80)^2 + (8-9)^2 + (85,7-85,2)^2 + (5-5)^2 + (5-5)^2 + (70-80)^2)$$

$$= 14,18626096$$

$$9,10 = = \text{SQRT}((90-80)^2 + (9-9)^2 + (85,2-85,2)^2 + (5-5)^2 + (5-5)^2 + (80-80)^2)$$

$$= 14,61813942$$

$$10,10 = = \text{SQRT}((80-80)^2 + (9-9)^2 + (85,2-85,2)^2 + (5-5)^2 + (5-5)^2 + (80-80)^2)$$

$$= 0$$

$$11,10 = = \text{SQRT}((80-80)^2 + (8-9)^2 + (85,3-85,2)^2 + (5-5)^2 + (5-5)^2 + (80-80)^2)$$

$$= 10,96403211$$

$$12,10 = = \text{SQRT}((70-80)^2 + (8-9)^2 + (84,3-85,2)^2 + (3-5)^2 + (5-5)^2 + (80-80)^2)$$

$$= 14,34015342$$

$$13,10 = = \text{SQRT}((90-80)^2 + (9-9)^2 + (85,8-85,2)^2 + (5-5)^2 + (5-5)^2 + (80-80)^2)$$

$$= 10,01798383$$

$$14,10 = = \text{SQRT}((80-80)^2 + (7-9)^2 + (87,4-85,2)^2 + (3-5)^2 + (3-5)^2 + (70-80)^2)$$

$$= 10,80925529$$

$$15,10 = = \text{SQRT}((80-80)^2 + (8-9)^2 + (89-85,2)^2 + (5-5)^2 + (3-5)^2 + (70-80)^2)$$

$$= 10,92886087$$

$$16,10 = = \text{SQRT}((90-80)^2 + (8-9)^2 + (85,4-85,2)^2 + (5-5)^2 + (5-5)^2 + (90-80)^2)$$

$$= 14,1788575$$

$$17,10 = = \text{SQRT}((90-80)^2 + (8-9)^2 + (84,3-85,2)^2 + (5-5)^2 + (5-5)^2 + (70-80)^2)$$

$$= 14,20598465$$

$$18,10 = = \text{SQRT}((80-80)^2 + (7-9)^2 + (86,1-85,2)^2 + (3-5)^2 + (4-5)^2 + (70-80)^2)$$

$$= 10,47902667$$

$$19,10 = = \text{SQRT}((80-80)^2 + (7-9)^2 + (85,6-85,2)^2 + (3-5)^2 + (6-5)^2 + (70-80)^2)$$

$$= 10,44796631$$

$$20,10 = = \text{SQRT}((80-80)^2 + (9-9)^2 + (89,3-85,2)^2 + (5-5)^2 + (3-5)^2 + (95-80)^2)$$

$$= 15,67832899$$

Setelah melakukan proses perhitungan maka dapat dilihat dalam bentuk tabel.

Tabel 4. 3 Data Jarak Iterasi I

| Jarak | Hasil |
|-------|-------------|
| 1,10 | 14,72446943 |
| 2,10 | 10,82820391 |
| 3,10 | 14,92112596 |
| 4,10 | 10,44030651 |
| 5,10 | 18,03025236 |
| 6,10 | 10,8240473 |
| 7,10 | 10,37352399 |
| 8,10 | 14,18626096 |
| 9,10 | 14,61813942 |
| 10,10 | 0 |
| 11,10 | 10,96403211 |
| 12,10 | 14,34015342 |
| 13,10 | 10,01798383 |
| 14,10 | 10,80925529 |
| 15,10 | 10,92886087 |
| 16,10 | 14,1788575 |
| 17,10 | 14,20598465 |
| 18,10 | 10,47902667 |
| 19,10 | 10,44796631 |
| 20,10 | 15,67832899 |

Density reachable diperoleh jika titik termasuk dalam radius dengan ketentuan nilai jarak kurang atau sama dengan Eps yang sudah diterapkan. *Neighborhood Core Object* tercapai lantaran total *e-neighborhood* telah melengkapi syarat karena total titik lebih dari sama dengan $\text{MinPts} = 11$. Titik selanjutnya dipilih berdasarkan jarak terjauh dari core object yaitu titik 11.

Iterasi II

Pada iterasi II titik pusat didapat dari iterasi II yaitu titik 11 dengan, 1 = 80, 2 = 9, 3 = 89,1, 4 = 5, 5 = 5, 6 = 80. Berikut tabel hasil perhitungan jarak setiap titik dengan titik pusat pada iterasi II.

Tabel 4. 4 Data Jarak Iterasi II

| Jarak | Hasil |
|-------|-------------|
| 1,11 | 22,47309502 |
| 2,11 | 14,42775104 |
| 3,11 | 10,00449899 |
| 4,11 | 14,26919759 |
| 5,11 | 27,34300642 |
| 6,11 | 4,272001873 |
| 7,11 | 6,216108107 |
| 8,11 | 10,74988372 |
| 9,11 | 22,47309502 |
| 10,11 | 10,96403211 |
| 11,11 | 0 |
| 12,11 | 11,40570033 |
| 13,11 | 14,6931957 |
| 14,11 | 2,808914381 |
| 15,11 | 0,1 |
| 16,11 | 22,75280203 |
| 17,11 | 11,27120224 |
| 18,11 | 3,872983346 |
| 19,11 | 5,123475383 |
| 20,11 | 25,02079135 |

Titik terjauh selanjutnya yang dapat dipilih yaitu titik 8.

Iterasi III

Tabel 4. 5 Data Jarak Iterasi III

| Jarak | Hasil |
|-------|-------------|
| 1,8 | 28,52998423 |
| 2,8 | 22,64950331 |
| 3,8 | 20,4374656 |
| 4,8 | 22,5 |
| 5,8 | 32,04122345 |
| 6,8 | 10,67754653 |
| 7,8 | 10,33247308 |
| 8,8 | 0 |
| 9,8 | 28,48227519 |
| 10,8 | 14,18626096 |
| 11,8 | 10,74988372 |
| 12,8 | 2,385372088 |
| 13,8 | 22,38325267 |
| 14,8 | 10,57780696 |
| 15,8 | 10,71867529 |
| 16,8 | 28,28586219 |
| 17,8 | 20,04894012 |
| 18,8 | 10,3033975 |
| 19,8 | 10,29611577 |
| 20,8 | 27,2572926 |

Titik terjauh selanjutnya yang dapat dipilih yaitu titik 6.

Iterasi IV

Tabel 4. 6 Data Jarak Iterasi IV

| Jarak | Hasil |
|-------|-------------|
| 1,6 | 22,47309502 |
| 2,6 | 14,42775104 |
| 3,6 | 10,00449899 |
| 4,6 | 14,26919759 |
| 5,6 | 27,34300642 |
| 6,6 | 4,272001873 |
| 7,6 | 6,216108107 |
| 8,6 | 10,74988372 |
| 9,6 | 22,47309502 |

| | |
|------|-------------|
| 10,6 | 10,96403211 |
| 11,6 | 0 |
| 12,6 | 11,40570033 |
| 13,6 | 14,6931957 |
| 14,6 | 2,808914381 |
| 15,6 | 0,1 |
| 16,6 | 22,75280203 |
| 17,6 | 11,27120224 |
| 18,6 | 3,872983346 |
| 19,6 | 5,123475383 |
| 20,6 | 25,02079135 |

Titik terjauh selanjutnya yang dapat dipilih yaitu titik 3.

Iterasi V

Tabel 4. 7 Data Jarak Iterasi V

| Jarak | Hasil |
|-------|-------------|
| 1,3 | 20,12486025 |
| 2,3 | 10,4158533 |
| 3,3 | 0 |
| 4,3 | 10,23914059 |
| 5,3 | 25,5 |
| 6,3 | 10,97451593 |
| 7,3 | 11,92518344 |
| 8,3 | 20,4374656 |
| 9,3 | 20,13082214 |
| 10,3 | 14,92112596 |
| 11,3 | 10,00449899 |
| 12,3 | 20,80865205 |
| 13,3 | 10,86093919 |
| 14,3 | 10,44030651 |
| 15,3 | 10,0079968 |
| 16,3 | 20,49390153 |
| 17,3 | 5,47813837 |
| 18,3 | 10,81156788 |
| 19,3 | 11,33313725 |
| 20,3 | 26,94457274 |

4.2.1 Implementasi DBSCAN

Pada tahap ini penulis melakukan import library yang akan digunakan, yaitu NearestNeighbor dan *DBSCAN*. Library NearestNeighbor akan digunakan untuk mendapatkan nilai epsilon yang optimal. Penulis juga memanggil data hasil preprocessing data yang akan digunakan pada tahap ini.

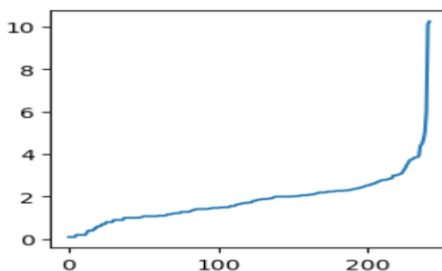
Setelah itu, penulis mendeklarasikan *x1* untuk mengambil nilai/*values* pada kolom yang sudah didefinisikan dalam *features* dan baris yang ditentukan oleh penulis, yaitu baris 0-242. Nilai pada *x1* ini akan digunakan pada *NearestNeighbors* dengan menggunakan data pada baris 0-242. Data yang digunakan ini merupakan data *sample* untuk menyederhanakan grafik yang didapat sehingga memudahkan penulis dalam menentukan nilai *Eps* yang akan digunakan.

```
#NEARESTNEIGHBORS
nn = NearestNeighbors (n_neighbors =6)
nbrs = nn.fit(df)
distances,indices = nbrs.kneighbors (df)
```

Gambar 4. 4 NearestNeighbors

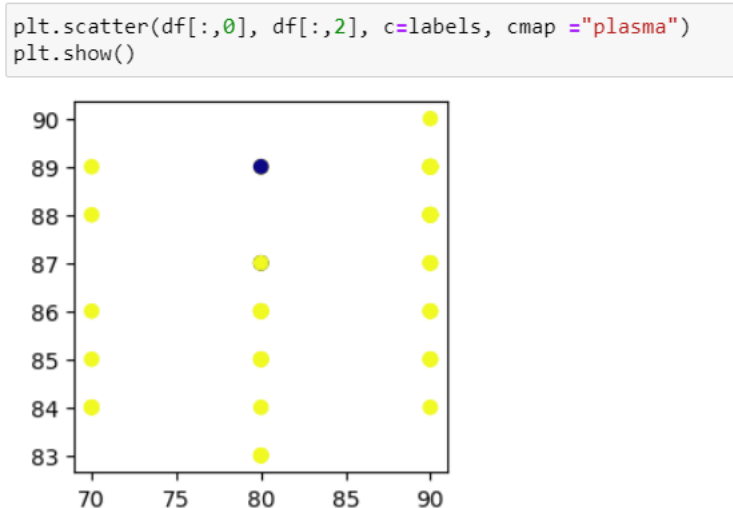
Penulis menampilkan grafik hasil *NearestNeighbors* untuk melihat titik optimal yang dapat digunakan sebagai nilai *Eps*. Penulis menampilkan grafik dengan ukuran 3,3 atau persegi, terlihat bahwa titik jarak optimal yang dapat digunakan berkisar antara 0.3-0.4. Oleh karena itu, penulis akan menggunakan nilai *Eps* 11, dengan *Min_Sample* 100.

```
#MENAMPILKAN GRAFIK HASIL NEARESTNEIGHBORS
distances = np.sort(distances, axis =0)
distances = distances[:,3]
plt.rcParams['figure.figsize'] = (3,3)
plt.plot(distances)
plt.show()
```



Gambar 4. 5 Grafik hasil Nearest Neighbors

Setelah mendapatkan nilai Eps yang optimal, penulis mengimplementasikan algoritma *DBSCAN Clustering*. Setelah itu penulis mendefinisikan array hasil *DBSCAN* dengan nama label. Jumlah kluster yang dihasilkan dan grafik hasil *DBSCAN Clustering* dapat dilihat pada gambar berikut ini.



Gambar 4. 6 Jumlah Kluster *DBSCAN Clustering*

4.3 Uji Validasi *DBSCAN*

Uji validasi algoritma *DBSCAN*, dilakukan menggunakan Silhouette Score. Nilai Silhouette Score antara -1 hingga +1 dan nilai yang lebih tinggi menunjukkan bahwa data poin sangat mirip dengan cluster sendiri dan berbeda dengan cluster tetangga lainnya. Setelah penulis mendefinisikan SI sebagai array hasil Silhouette Score. Uji Silhouette ini menggunakan database yang mendefinisikan database.csv dan label sebagai nilai hasil *DBSCAN Clustering* dalam bentuk array.

```
SI = silhouette_score(df, labels)
print(SI)
```

0.258030877243884

Gambar 4. 7 Implementasi Silhouette Score pada *DBSCAN*

Berdasarkan hasil validasi menggunakan Silhouette Coefficient, metode *DBSCAN* mendapatkan Silhouette Score dengan pembentukan 2 kluster yaitu sebesar 0.258030877243884.

4.4 *K-Means Clustering*

Algoritma *K-Means* adalah algoritma yang mempartisi data ke dalam cluster – cluster sehingga data yang memiliki kemiripan berada pada satu cluster yang sama dan data yang memiliki ketidaksamaan berada pada cluster yang lain. Langkah – Langkah yang dibutuhkan dalam seleksi penentuan siswa terbaik menggunakan rumus algoritma *K-Means* sebagai berikut:

Tabel 4. 8 Contoh Data

| Siswa | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|----|---|------|---|---|----|
| 1 | 90 | 9 | 89,3 | 5 | 5 | 90 |
| 2 | 90 | 8 | 88,7 | 3 | 5 | 80 |
| 3 | 90 | 8 | 89,4 | 5 | 3 | 70 |
| 4 | 90 | 8 | 87,2 | 5 | 3 | 80 |
| 5 | 90 | 9 | 84,9 | 5 | 5 | 95 |
| 6 | 80 | 7 | 85,6 | 3 | 2 | 70 |
| 7 | 80 | 7 | 83,3 | 5 | 5 | 70 |
| 8 | 70 | 8 | 85,7 | 5 | 5 | 70 |
| 9 | 90 | 9 | 88,9 | 5 | 5 | 90 |
| 10 | 80 | 9 | 85,2 | 5 | 5 | 80 |
| 11 | 80 | 8 | 89,1 | 5 | 3 | 70 |
| 12 | 70 | 8 | 84,4 | 3 | 5 | 70 |
| 13 | 90 | 9 | 85,8 | 5 | 5 | 80 |
| 14 | 80 | 7 | 87,4 | 3 | 3 | 70 |
| 15 | 80 | 8 | 89 | 5 | 3 | 70 |
| 16 | 90 | 8 | 85,4 | 5 | 5 | 90 |
| 17 | 90 | 8 | 84,3 | 5 | 5 | 70 |
| 18 | 80 | 7 | 86,1 | 3 | 4 | 70 |
| 19 | 80 | 7 | 85,6 | 3 | 6 | 70 |
| 20 | 80 | 9 | 89,3 | 5 | 3 | 95 |

Sumber: Data sekolah

Dari data diatas jumlah klaster yang digunakan ialah 2 klaster dengan perhitungannya sebagai beriku:

Tabel 4. 9 Centroid Klaster yang Terpilih

| | | | | | | |
|---|----|---|------|---|---|----|
| 1 | 90 | 9 | 89,3 | 5 | 5 | 90 |
| 2 | 90 | 8 | 88,7 | 3 | 5 | 80 |

Iterasi 1**Centroid Cluster 1**

$$\begin{aligned}
&= \text{SQRT}((90-90)^2+(9-9)^2+(89,3-89,3)^2+(5-5)^2+(5-5)^2+(90-90)^2) = 0 \\
&= \text{SQRT}((90-90)^2+(9-8)^2+(89,3-88,7)^2+(5-3)^2+(5-5)^2+(90-80)^2) = 10,2645 \\
&= \text{SQRT}((90-90)^2+(9-8)^2+(89,3-89,4)^2+(5-5)^2+(5-3)^2+(90-70)^2) = 20,12486 \\
&= \text{SQRT}((90-90)^2+(9-8)^2+(89,3-87,2)^2+(5-5)^2+(5-3)^2+(90-80)^2) = 10,45992 \\
&= \text{SQRT}((90-90)^2+(9-9)^2+(89,3-84,9)^2+(5-5)^2+(5-5)^2+(90-95)^2) = 6,66033 \\
&= \text{SQRT}((90-80)^2+(9-7)^2+(89,3-85,6)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 23,03671 \\
&= \text{SQRT}((90-80)^2+(9-7)^2+(89,3-83,3)^2+(5-5)^2+(5-5)^2+(90-70)^2) = 23,2379 \\
&= \text{SQRT}((90-70)^2+(9-8)^2+(89,3-85,7)^2+(5-5)^2+(5-5)^2+(90-70)^2) = 28,52998 \\
&= \text{SQRT}((90-90)^2+(9-9)^2+(89,3-88,9)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 0,4 \\
&= \text{SQRT}((90-80)^2+(9-9)^2+(89,3-85,2)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 14,72447 \\
&= \text{SQRT}((90-80)^2+(9-8)^2+(89,3-89,1)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 22,4731 \\
&= \text{SQRT}((90-17)^2+(9-8)^2+(89,3-84,4)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 75,88155 \\
&= \text{SQRT}((90-90)^2+(9-9)^2+(89,3-85,8)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 10,59481 \\
&= \text{SQRT}((90-80)^2+(9-7)^2+(89,3-87,4)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 22,70705 \\
&= \text{SQRT}((90-80)^2+(9-8)^2+(89,3-89)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 22,47421 \\
&= \text{SQRT}((90-90)^2+(9-8)^2+(89,3-85,4)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 4,026164 \\
&= \text{SQRT}((90-90)^2+(9-8)^2+(89,3-85,4)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 20,63977 \\
&= \text{SQRT}((90-80)^2+(9-7)^2+(89,3-84,3)^2+(5-3)^2+(5-2)^2+(90-70)^2) = 22,78684 \\
&= \text{SQRT}((90-80)^2+(9-7)^2+(89,3-86,1)^2+(5-3)^2+(5-6)^2+(90-70)^2) = 22,86241 \\
&= \text{SQRT}((90-80)^2+(9-9)^2+(89,3-85,6)^2+(5-5)^2+(5-3)^2+(90-95)^2) = 11,35782
\end{aligned}$$

Centroid Cluster 2

$$\begin{aligned}
&= \text{SQRT}((90-90)^2+(9-8)^2+(89,3-88,7)^2+(5-5)^2+(5-3)^2+(90-70)^2) = \\
&20,12486025 \\
&= \text{SQRT}((90-90)^2+(8-8)^2+(89-87)^2+(3-5)^2+(5-3)^2+(80-80)^2) = \\
&3,201562119 \\
&= \text{SQRT}((90-90)^2+(8-9)^2+(89-85)^2+(5-5)^2+(3-5)^2+(70-95)^2) = 25,5 \\
&= \text{SQRT}((90-80)^2+(8-7)^2+(87-86)^2+(5-3)^2+(3-2)^2+(80-70)^2) = \\
&14,44160656
\end{aligned}$$

$$= \text{SQRT}((90-80)^2 + (9-7)^2 + (85-83)^2 + (5-5)^2 + (5-5)^2 + (95-70)^2) = 27,04736586$$

$$= \text{SQRT}((80-70)^2 + (7-8)^2 + (86-86)^2 + (3-5)^2 + (2-5)^2 + (70-70)^2) = 10,67754653$$

$$= \text{SQRT}((80-90)^2 + (7-9)^2 + (83-89)^2 + (5-5)^2 + (5-5)^2 + (70-90)^2) = 23,13784778$$

$$= \text{SQRT}((70-80)^2 + (8-9)^2 + (86-85)^2 + (5-5)^2 + (5-5)^2 + (70-80)^2) = 14,18626096$$

$$= \text{SQRT}((90-80)^2 + (9-8)^2 + (89-89)^2 + (5-5)^2 + (5-3)^2 + (90-70)^2) = 22,47309502$$

$$= \text{SQRT}((80-70)^2 + (9-8)^2 + (85-84)^2 + (5-3)^2 + (5-5)^2 + (80-70)^2) = 14,34015342$$

$$= \text{SQRT}((80-90)^2 + (8-9)^2 + (89-86)^2 + (5-5)^2 + (3-5)^2 + (70-80)^2) = 14,6931957$$

$$= \text{SQRT}((70-80)^2 + (8-7)^2 + (84-87)^2 + (3-3)^2 + (5-3)^2 + (70-70)^2) = 10,67707825$$

$$= \text{SQRT}((90-80)^2 + (9-8)^2 + (86-89)^2 + (5-5)^2 + (5-3)^2 + (80-70)^2) = 14,67105995$$

$$= \text{SQRT}((80-90)^2 + (7-8)^2 + (87-85)^2 + (3-5)^2 + (3-5)^2 + (70-90)^2) = 22,64950331$$

$$= \text{SQRT}((80-90)^2 + (8-8)^2 + (89-84)^2 + (5-5)^2 + (3-5)^2 + (70-70)^2) = 11,22898036$$

$$= \text{SQRT}((90-80)^2 + (8-7)^2 + (85-86)^2 + (5-3)^2 + (5-4)^2 + (90-70)^2) = 22,5053327$$

$$= \text{SQRT}((90-80)^2 + (8-7)^2 + (84-86)^2 + (5-3)^2 + (5-6)^2 + (70-70)^2) = 10,37737925$$

$$= \text{SQRT}((80-80)^2 + (7-9)^2 + (86-89)^2 + (3-5)^2 + (4-3)^2 + (70-95)^2) = 25,3818833$$

$$= \text{SQRT}((80-80)^2 + (7-7)^2 + (86-86)^2 + (3-5)^2 + (6-3)^2 + (70-70)^2) = 136,8260209$$

$$=\text{SQRT}((80-90)^2+(9-8)^2+(89-86)^2+(5-3)^2+(3-6)^2+(95-70)^2)=153,3443511$$

Iterasi 2

Tabel 4. 10 Iterasi 2

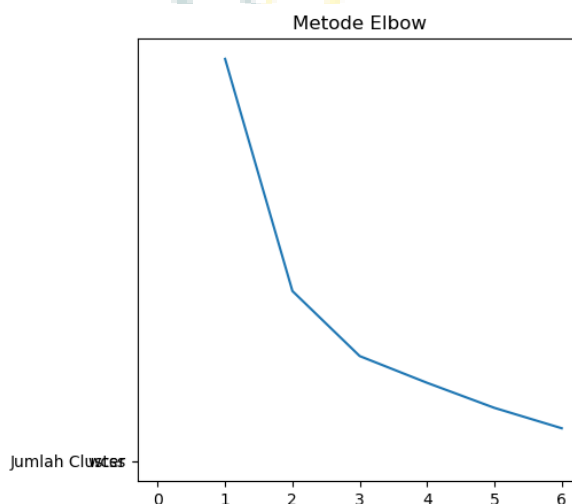
| | |
|-------------|-------------|
| 20,12486025 | 10,45992352 |
| 3,201562119 | 15,63457706 |
| 25,5 | 10,97451593 |
| 14,44160656 | 14,83947438 |
| 27,04736586 | 32,04122345 |
| 10,67754653 | 22,97585689 |
| 23,13784778 | 10,37352399 |
| 14,18626096 | 10,74988372 |
| 22,47309502 | 28,72716484 |
| 14,34015342 | 10,01798383 |
| 14,6931957 | 2,808914381 |
| 10,67707825 | 11,36485812 |
| 14,67105995 | 10,05783277 |
| 22,64950331 | 10,89082182 |
| 11,22898036 | 3,796050579 |
| 22,5053327 | 22,49533285 |
| 10,37737925 | 27,47726333 |
| 25,3818833 | 137,0664437 |
| 136,8260209 | 136,8260209 |
| 153,3443511 | 153,3443511 |
| 20,12486025 | 10,45992352 |

4.5 Implementaasi Algoritma *K-Means*

Setelah itu, mengimplementasikan *K-Means Clustering* dengan menggunakan *library K-Means*. Didefinisikan *K-Means* untuk menjalankan algoritma *K-Mean Clustering*. Penulis melakukan prediksi seperti apa pengelompokan klaster dengan menggunakan $n_cluster = 2$. Penulis menggunakan *K-Means++* agar tidak terkena jebakan *centroid*. Setelah itu, penulis menyiapkan objek *y_K-Means* dengan menggunakan *methodfit_predict* terhadap variabel *x* yang sudah didefinisikan agar mendapatkan index klaster. *y_K-Means* pada adalah objek hasil dari prediksi algoritma *K-Means Clustering* dalam bentuk *array*. Hasil klasterisasi *y_K-Means* yang didapat dari implementasi algoritma *K-Means* ini.

```
In [70]: #METODE KMEANS
wcss = []
for i in range(1, 7):
    kmeans = KMeans(n_clusters = i, init = "k-means+", random_state = 24)
    kmeans.fit(df)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 7), wcss)
plt.title("Metode Elbow")
plt.plot("Jumlah Cluster")
plt.plot("wcss")
plt.show()
```

Gambar 4. 8 Implementasi Algoritma *K-Means*

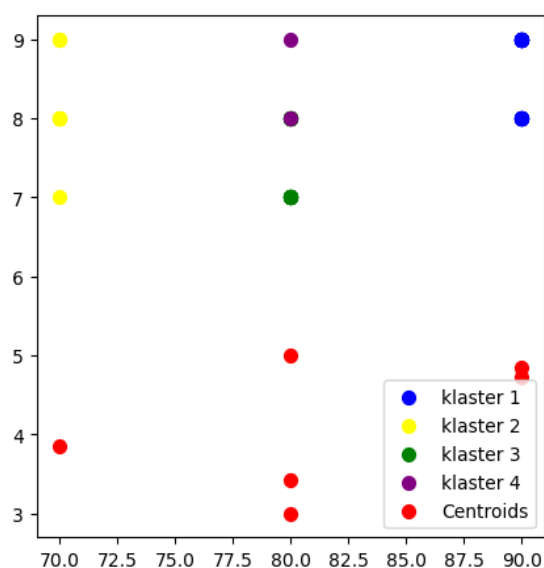


Gambar 4. 9 Grafik Hasil Implementasi

Kemudian untuk memvisualisasikan kluster 1, pada sumbu x, ditulis $df[y_K\text{-Means} == 0, 0]$ artinya data poin nya berasal dari objek x. Penulisan $y_K\text{-Means} == 0$ adalah untuk mewakili baris x yang dipilih, yaitu baris yang clusternya masuk ke cluster 1. Untuk kolomnya, karena ia adalah data pca1, yaitu kolom 1 pada objek x. sehingga penulisannya setelah $y_K\text{-Means} == 0$ adalah koma, kemudian diikuti 0. Sedangkan untuk sumbu y, ditulis $x[y_K\text{-Means} == 0, 1]$ di mana baris yang dipilih adalah kluster yang memiliki nilai 0. Dan untuk kolomnya diisi dengan 1. Untuk menampilkan titik-titik kluster lainnya, maka penulis merubah nilai $y_K\text{-Means}$ pada baris-baris selanjutnya. Dan grafik hasil *K-Means Clustering* dapat dilihat pada Gambar dibawah ini.

```
In [79]: plt.scatter(df[y_kmeans==0,0], df[y_kmeans==0,1], s=50, c="blue", label = "klaster 1")
plt.scatter(df[y_kmeans==1,0], df[y_kmeans==1,1], s=50, c="yellow", label = "klaster 2")
plt.scatter(df[y_kmeans==2,0], df[y_kmeans==2,1], s=50, c="green", label = "klaster 3")
plt.scatter(df[y_kmeans==3,0], df[y_kmeans==3,1], s=50, c="purple", label = "klaster 4")
#plt.scatter(df[y_kmeans==4,0], df[y_kmeans==4,1], s=50, c="black", label = "klaster 5")
#plt.scatter(df[y_kmeans==5,0], df[y_kmeans==5,1], s=242, c="purple", label = "klaster 6")
plt.scatter(kmeans.cluster_centers[:, 0], kmeans.cluster_centers[:, 1], s=50, c = "red", label = "Centroids")
plt.rcParams['figure.figsize'] = (5,5)
plt.legend()
plt.show()
```

Gambar 4. 10 Menampilkan hasil K- Means Clustering



Gambar 4. 11 Grafik hasil K-Means Clustering

4.6 Uji Validasi DBSCAN K-Means

Tahap terakhir pada Improved *K-Means* adalah uji validasi pada implementasi hasil pengembangan metode *K-Means Clustering* dengan *DBSCAN Clustering*. Uji validasi yang dilakukan oleh penulis menggunakan Uji Validasi *Silhouette* yang memiliki rentang nilai -1 hingga 1 Seperti sub bab *K-Means Clustering*, tahap ini juga dilakukan dengan menggunakan *tools* Jupyter Notebook pada file *K-Means_clustering.ipynb*

```
In [80]: from sklearn.metrics import silhouette_samples, silhouette_score
SI = silhouette_score(df,y_kmeans)
print(SI)
```

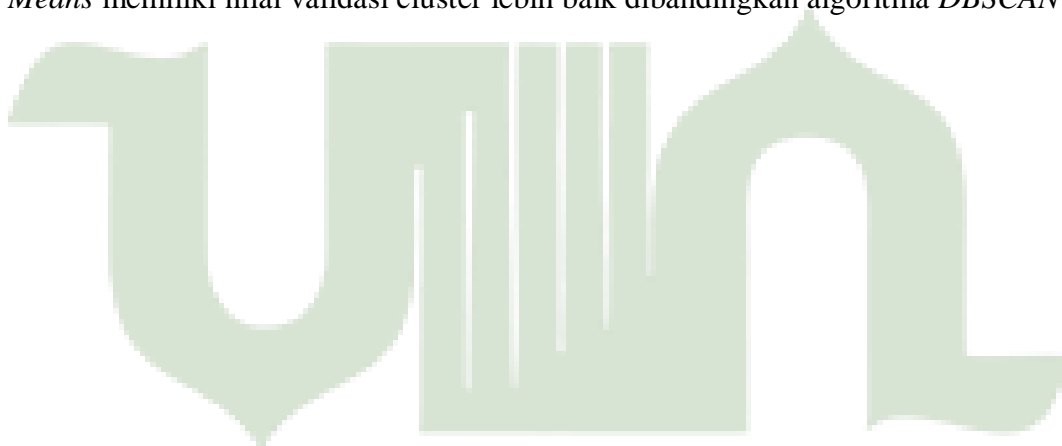
0.5697019340266847

Gambar 4. 12 Uji Validasi Silhouette pada *K-Means Clustering*

Berdasarkan hasil validasi menggunakan Silhouette Coefficient, metode *K-Means* mendapatkan Silhouette Score dengan pembentukan 4 kluster yaitu sebesar 0.5697019340266847.

4.7 Perbandingan Algoritma *DBSCAN* dan *K-Means*

Perbandingan algoritma *DBSCAN* dan *K-Means* dilakukan dengan menentukan jumlah cluster yang paling optimal dilihat dari nilai *Silhouette Index* (SI). Grafik perbandingan nilai SI tertinggi pada setiap algoritma. Hasil pengujian validitas cluster terhadap algoritma *DBSCAN* dan *K-Means* menggunakan nilai *Silhouette Index* (SI) sudah dilakukan, klasterisasi menggunakan *K-Means* yang memiliki nilai SI terbaik yaitu 0.5697019340266847. Sementara dari hasil klasterisasi menggunakan algoritma *DBSCAN* diperoleh nilai 0.258030877243884 dengan nilai epsilon 11 dan nilai MinPts 11. Maka pada penelitian ini, algoritma *K-Means* memiliki nilai validasi cluster lebih baik dibandingkan algoritma *DBSCAN*



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN