

BAB IV

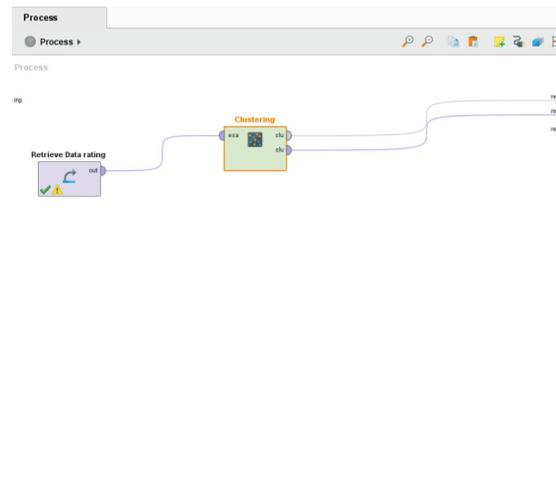
HASIL DAN PEMBAHASAN

4.1 Hasil

Berdasarkan hasil identifikasi masalah yang telah dilakukan oleh peneliti, maka ditemukan bahwa penggunaan algoritma k-means dan DBSCAN dapat mengatasi permasalahan yang ada pada *cold star* dengan menggunakan rentang umur sebagai *indicator* pemilihan rekomendasi film kepada pengguna baru

4.1.1 Pengaplikasian algoritma K-means pada data rating Film

Pada pengaplikasian algoritma K-Means pada data rating yang dilakukan oleh peneliti, ditemukan bahwa penggunaan K optimal menurut uji David bouldin index ialah sebesar 2.

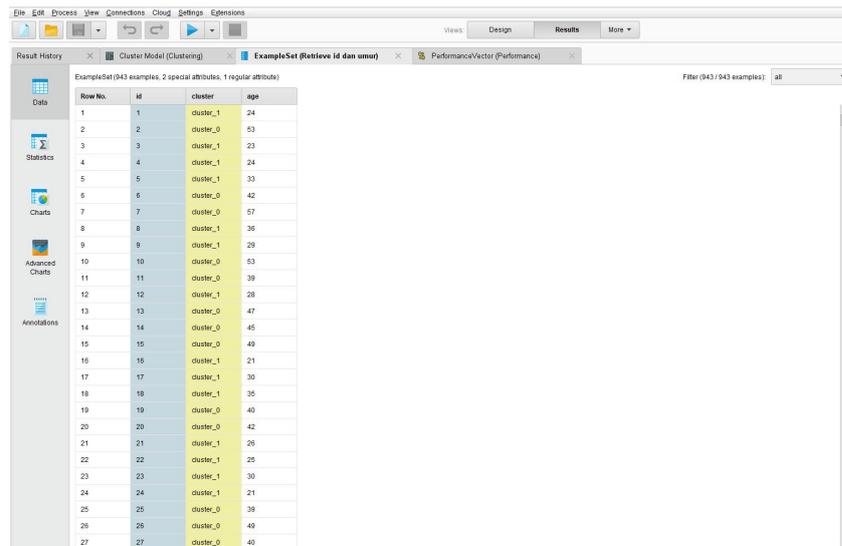


Gambar 4.1 Proses *clustering* data rating menggunakan algoritma K-Means

Dibawah merupakan hasil *cluster* model yang didapatkan

Cluster 0: 332 items
 Cluster 1: 611 items
 Total number of items: 943

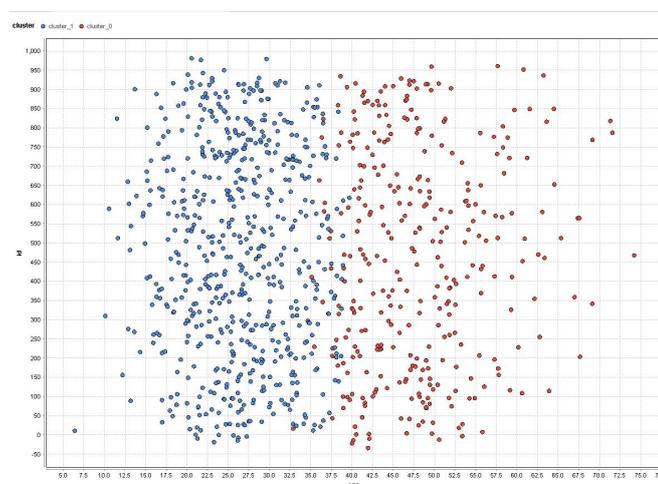
Dengan hasil penyebaran *cluster* dapat di lihat pada gambar 4.2



Row No.	id	cluster	age
1	1	cluster_1	24
2	2	cluster_0	53
3	3	cluster_1	23
4	4	cluster_1	24
5	5	cluster_1	33
6	6	cluster_0	42
7	7	cluster_0	57
8	8	cluster_1	36
9	9	cluster_1	29
10	10	cluster_0	53
11	11	cluster_0	39
12	12	cluster_1	28
13	13	cluster_0	47
14	14	cluster_0	45
15	15	cluster_0	49
16	16	cluster_1	21
17	17	cluster_1	30
18	18	cluster_1	35
19	19	cluster_0	40
20	20	cluster_0	42
21	21	cluster_1	26
22	22	cluster_1	25
23	23	cluster_1	30
24	24	cluster_1	21
25	25	cluster_0	39
26	26	cluster_0	49
27	27	cluster_0	40

Gambar 4.2 Rincian penyebaran data *cluster* K-Means

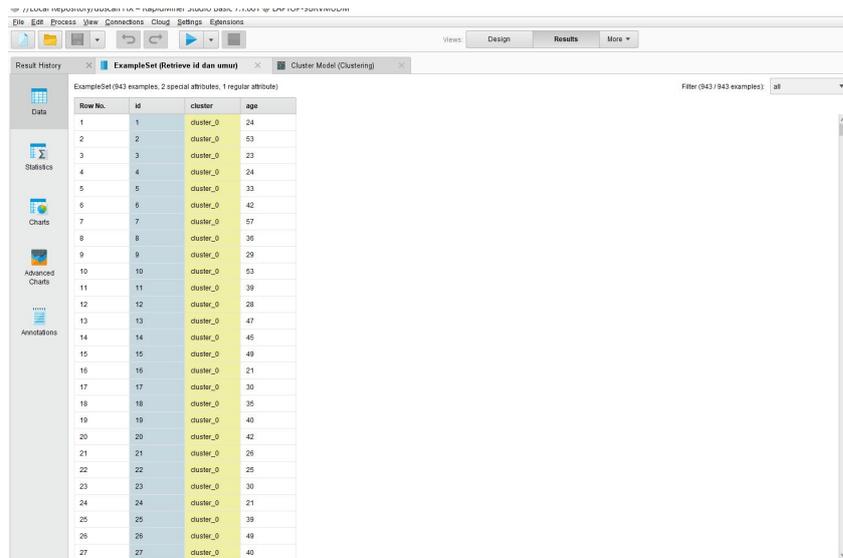
Dengan visualisasi data penyebaran dapat disimak pada gambar 4.3



Gambar 4.3 visualisasi data *cluster* menggunakan K-means

4.1.2 Pengaplikasian algoritma DBSCAN pada data rating Film

Setelah dilakukan penelitian menggunakan algoritma KNN ditemukan bahwa nilai eps sebesar 0,7. Nilai 0,7 didapatkan dari dari posisi “*knee*” yang terlihat pada plot seperti pada gambar 4.4



Row No.	id	cluster	age
1	1	cluster_0	24
2	2	cluster_0	53
3	3	cluster_0	23
4	4	cluster_0	24
5	5	cluster_0	33
6	6	cluster_0	42
7	7	cluster_0	57
8	8	cluster_0	36
9	9	cluster_0	29
10	10	cluster_0	53
11	11	cluster_0	39
12	12	cluster_0	28
13	13	cluster_0	47
14	14	cluster_0	45
15	15	cluster_0	49
16	16	cluster_0	21
17	17	cluster_0	30
18	18	cluster_0	35
19	19	cluster_0	40
20	20	cluster_0	42
21	21	cluster_0	26
22	22	cluster_0	25
23	23	cluster_0	30
24	24	cluster_0	21
25	25	cluster_0	39
26	26	cluster_0	43
27	27	cluster_0	40

Gambar 4.4 Rincian penyebaran data *cluster* DBSCAN

4.1.2 Pengaplikasian Cosine similarty

Pada gambar 4.5 dan gambar 4.5 dibawah merupakan hasil perhitungan *cosine similarty* pada setiap objek pada data. *cosine* menghitung kemiripan-kemiripan antar data dengan rentang nilai 0,00 – 1,0. Semakin besar nilai yang didapatkan, menandakan semakin mirip data tersebut. sebagai contoh pada tabel 4.1, dijelaskan pada baris pertama bahwa data pertama dihitung dengan data kedua menghasilkan nilai 1, yang berarti data ke-1 dan data ke-2 memiliki kemiripan yang autentik. Selanjutnya, dicoba pada data ke-1 dengan data ke-3 pada baris kedua memiliki kemiripan data yang autentik.

First	Second	Similarity
1.0	2.0	1
1.0	3.0	1
1.0	4.0	1
1.0	5.0	1
1.0	6.0	1
1.0	7.0	1
1.0	8.0	1
1.0	9.0	1
1.0	10.0	1
1.0	11.0	1
1.0	12.0	1
1.0	13.0	1
1.0	14.0	1
1.0	15.0	1
1.0	16.0	1
1.0	17.0	1
1.0	18.0	1
1.0	19.0	1
1.0	20.0	1
1.0	21.0	1
1.0	22.0	1
1.0	23.0	1
1.0	24.0	1
1.0	25.0	1

Gambar 4.5 Hasil Uji Cosine Similarity *Cluster 1* K-Means

First	Second	Similarity
1.0	2.0	1
1.0	3.0	1
1.0	4.0	1
1.0	5.0	1
1.0	6.0	1
1.0	7.0	1
1.0	8.0	1
1.0	9.0	1
1.0	10.0	1
1.0	11.0	1
1.0	12.0	1
1.0	13.0	1
1.0	14.0	1
1.0	15.0	1
1.0	16.0	1
1.0	17.0	1
1.0	18.0	1
1.0	19.0	1
1.0	20.0	1
1.0	21.0	1
1.0	22.0	1
1.0	23.0	1
1.0	24.0	1
1.0	25.0	1

Gambar 4.5 Hasil Uji Cosine Similarity *Cluster 2* K-Means

Dari hasil uji cosine similarity kita dapat melihat bahwa persamaan antara dokumen satu dan yang lain sebesar 1 dimana mengartikan sudut vektor yang dibuat oleh dokumen A dan B semakin kecil. Semakin besar nilai yang didapatkan, menandakan semakin mirip data tersebut dengan rentang 0-1.

4.2 PEMBAHASAN

Setelah dilakukan penelitian yang dilakukan peneliti dengan menggunakan dua algoritma yakni K-means dan DBSCAN ditemukan bahwa K-means didapati bahwa 2 *cluster* merupakan pengelompokan yang paling baik. Namun, pada algoritma DBSCAN ditemukan bahwa kurang tepatnya penggunaan algoritma DBSCAN pada data rating.

4.2.1 K-Means Pada Data Rating

Setelah dilakukan penelitian oleh peneliti ditemukan bahwa *cluster* yang terbentuk pada data set yakni 2 *cluster*. *Cluster* tersebut didapatkan dengan menggunakan uji davies bouldin index yang telah di lakukan uji coba dengan angka 1-30. Cara kerja davies bouldin index adalah menemukan nilai terkecil mendekati 0 untuk menemukan nilai K optimal pada proses *clusterisasi* menggunakan algoritma K-means. Hasil dari implementasi algoritma K-means dapat dilihat pada tabel 4.1 berdasarkan umur pengguna

Tabel 4.1 Hasil *clustering* menggunakan K-means

Row No.	id	cluster	age
1	1	cluster_1	24
2	2	cluster_0	53
3	3	cluster_1	23
4	4	cluster_1	24
5	5	cluster_1	33
6	6	cluster_0	42
7	7	cluster_0	57
8	8	cluster_1	36
9	9	cluster_1	29
10	10	cluster_0	53
11	11	cluster_0	39
12	12	cluster_1	28
13	13	cluster_0	47
14	14	cluster_0	45
15	15	cluster_0	49
16	16	cluster_1	21
17	17	cluster_1	30
18	18	cluster_1	35

Dimana umur ≤ 38 masuk kedalam *cluster* 1 dan ≤ 39 masuk kedalam *cluster* 2.

4.2.2 DBSCAN Pada Data Rating

Proses penelitian pengimplementasian DBSCAN dalam proses mencari nilai Eps yakni menggunakan algoritma KNN dengan cara mencari “knee” pada plot kedekatan jarak. Eps yang didapatkan yakni 0,7. Namun, sangat disayangkan hasil dari proses tersebut ditemukan bahwa *cluster* tidak terbentuk. Untuk pengujian yang lebih dalam penulis mencoba berbagai parameter eps untuk membuktikan kebenaran proses DBSCAN. Berikut tabel hasil uji coba nilai berbagai eps pada tabel 4.2

Tabel 4.2 Uji coba nilai Eps

Eps	MinPts	Number of <i>cluster</i>	Noise
0.3	5	1	0
0.3	6	1	0
0.3	7	1	0
0.3	8	1	0
0.3	9	1	0
0.3	10	1	0
0.4	5	1	0
0.4	6	1	0
0.4	7	1	0
0.4	8	1	0
0.4	9	1	0
0.4	10	1	0
0.5	5	1	0

0.5	6	1	0
0.5	7	1	0
0.5	8	1	0
0.5	9	1	0

Percobaan terus dilakukan sampai dengan nilai Eps 3. Namun, nilai *cluster* yang ditemukan tetap 1. Hal ini sesuai dengan teori yang ditulis oleh Dr. Suyanto dalam bukunya yang berjudul Data mining untuk klasifikasi dan klasterisasi yang ditulis pada tahun 2019. Pada buku tersebut dijelaskan bahwa penggunaan DBSCAN akan lebih efektif digunakan pada data yang berbentuk spasial.

4.2.3 Proses Validation

Pada proses validation menggunakan algoritma KNN ditemukan bahwa

1-Nearest Neighbour model for classification.
The model contains 660 examples with 2 dimensions of the following classes:

```
cluster_0
cluster_1
```

dengan hasil performance sebesar 100%

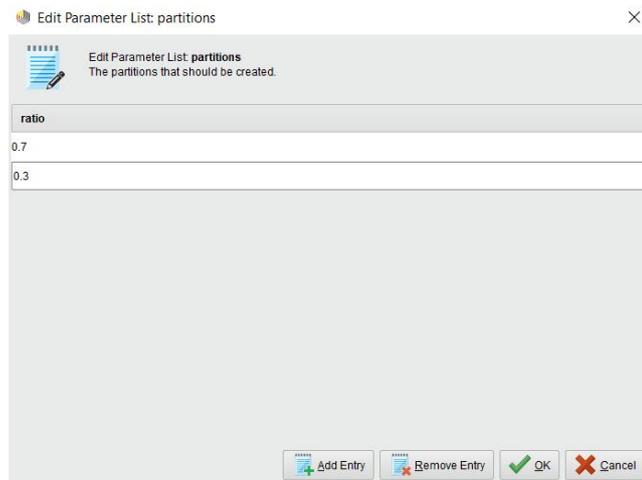
Table View Plot View

accuracy: 100.00% +/- 0.00% (mikro: 100.00%)

	true cluster_0	true cluster_1	class precision
pred. cluster_0	611	0	100.00%
pred. cluster_1	0	332	100.00%
class recall	100.00%	100.00%	

Gambar 4.6 Performance data set *cluster* hasil K-Means

Proses penggunaan data menggunakan split data sebesar 70:30. Dimana *data training* sebesar 70% dari keseluruhan data dan *data testing* sebesar 30% dari keseluruhan data yang dipisahkan oleh parameter *split data* pada rapid miner.



Gambar 4.7 Proses Split data

Sedangkan performance dari dari algoritma DBSCAN sebesar 98,52% yang diuji dengan algoritma K-NN dengan perlakuan yang sama pada data K-means di atas. Berikut gambar 4.8 merupakan visualisasi hasil performance dari data DBSCAN dan pada gambar 4.9 merupakan hasil performa data yang belum di *cluster* menggunakan algoritma K-Means dan DBSCAN.

accuracy: 98.52% +/- 0.70% (mikro: 98.52%)

	true cluster 1	true noise	true cluster 2	class precision
pred. cluster 1	929	4	7	98.83%
pred. noise	0	0	0	0.00%
pred. cluster 2	3	0	0	0.00%
class recall	99.88%	0.00%	0.00%	

Gambar 4.8 Performance data set *cluster* hasil DBSCAN

accuracy: 74.56% +/- 4.38% (mikro: 74.55%)

	true remaja akhir	true lansia awal	true dewasa awal	true dewasa ak...	true lansia akhir	true anak-anak	true remaja awal	true manula	true remaja akhir	class precision
pred. remaja ak...	201	0	36	0	0	2	18	0	1	77.91%
pred. lansia awal	0	103	0	20	23	0	0	2	0	69.59%
pred. dewasa a...	42	0	244	25	0	0	0	0	0	78.46%
pred. dewasa a...	0	26	19	137	0	0	0	0	0	75.27%
pred. lansia akhir	0	9	0	0	16	0	0	6	0	51.61%
pred. anak-anak	0	0	0	0	0	0	0	0	0	0.00%
pred. remaja a...	5	0	0	0	0	1	1	0	0	14.29%
pred. manula	0	0	0	0	4	0	0	1	0	20.00%
pred. remaja ak...	1	0	0	0	0	0	0	0	0	0.00%
class recall	80.72%	74.64%	81.61%	75.27%	37.21%	0.00%	5.26%	11.11%	0.00%	

Gambar 4.9 performa data set sebelum *clustering*

4.2.4 Komparasi K-Means dan DBSCAN

Hasil implementasi K-Means dan DBSCAN pada penelitian ini ditemui perbedaan dimana pada algoritma K-Means ditemukan 2 *cluster* sedangkan penggunaan DBSCAN ditemukan 1 *cluster*. Untuk melihat apakah *clusterisasi* dapat menyelesaikan permasalahan *cold star* pada penelitian ini, maka penulis menguji apakah data yang telah di *cluster* dan setelah di *cluster* mengalami nilai akurasi yang jauh lebih tinggi menggunakan algoritma K-NN. Berikut pada tabel 4.3

Tabel 4.3 validasi KNN data setelah dan sebelum *clustering*

	Akurasi setelah <i>clustering</i>	Akurasi sebelum <i>clustering</i>
K-Means	100%	74,55%
DBSCAN	98,52%	