BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Klasterisasi K-Means Pada *Dataset* Demografi Fiks

Pada ***Dataset*** **Demografi Fiks** dilakukan klasterisasi dengan algoritma K-Means dari k = 2 hingga k = 30 untuk mencari *Davies Bouldin Index* (DBI) terkecil. *Davies Bouldin Index (DBI)* terkecil merupakan salah satu indikator jumlah klaster terbaik pada suatu *dataset*. Hasilnya disajikan pada tabel 4.1.

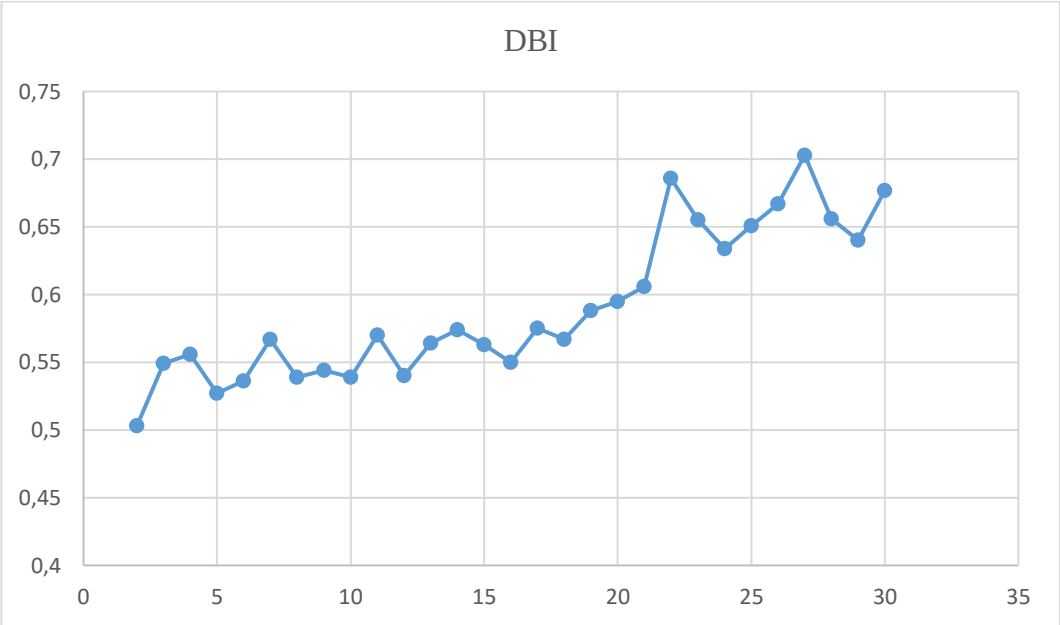
Tabel 4.1 *Davies Bouldin Index* (DBI) k = 2 Hingga k = 30

Jumlah klastering (k)	*Davies Bouldin Index* (DBI)
2	0,503
3	0,549
4	0,556
5	0,527
6	0,536
7	0,567
8	0,539
9	0,544
10	0,539
11	0,570
12	0,540
13	0,564
14	0,574
15	0,563

Jumlah klastering (k)	*Davies Bouldin Index* (DBI)
16	0,550
17	0,575
18	0,567
19	0,588
20	0,595
21	0,606
22	0,686
23	0,655
24	0,634
25	0,651
26	0,667
27	0,703
28	0,656
29	0,640
30	0,677

Ketika dilakukan percobaan dengan k = 5 dan k = 6, nilai DBI terjadi penurunan dibandingkan dengan nilai DBI ketika k = 4. Tetapi nilai DBI nya tidak lebih kecil dari nilai DBI k = 2. Maka percobaan K-Means terus dilanjutkan dengan kenaikan nilai k = 7 dan seterusnya hingga mencapai k = 30.

Dari percobaan klasterisasi K-Means dengan k = 2 hingga k = 30, maka nilai DBI memiliki kecenderungan semakin naik. Grafik DBI yang semakin besar nilainya ditunjukkan pada gambar 4.1.



Gambar 4.1 Grafik pergerakan nilai DBI dari k = 2 hingga k = 30

Pada gambar 4.1, nilai DBI beberapa kali turun daripada nilai DBI sebelumnya. Akan tetapi, nilai DBI yang turun tidak lebih rendah daripada DBI pada k = 2 yaitu di atas 0,503. Dari percobaan dengan k = 2 hingga k = 30, k = 27 memiliki DBI terbesar yaitu 0,703. Artinya, terdapat selisih 0,200 terhadap DBI terendah.

Pada penelitian Elly Muningsih dkk [23] dan Azzahra dkk [24], dilakukan percobaan dengan K-Means dari k = 2 hingga k = 10. Elly Muningsih dkk [23] melakukan penelitian dengan K-Means pada k = 2 hingga k = 10 dan mendapatkan nilai DBI 0,175 hingga 0,720. DBI terendah yaitu 0,175 berada pada k = 3. Azzahra dkk [24] melakukan penelitian dengan K-Means pada k = 2 hingga 10 dan mendapatkan nilai DBI 1,3427 hingga 2,0794. DBI terendah yaitu 1,3427 berada pada k = 9. Kedua penelitian ini melakukan percobaan

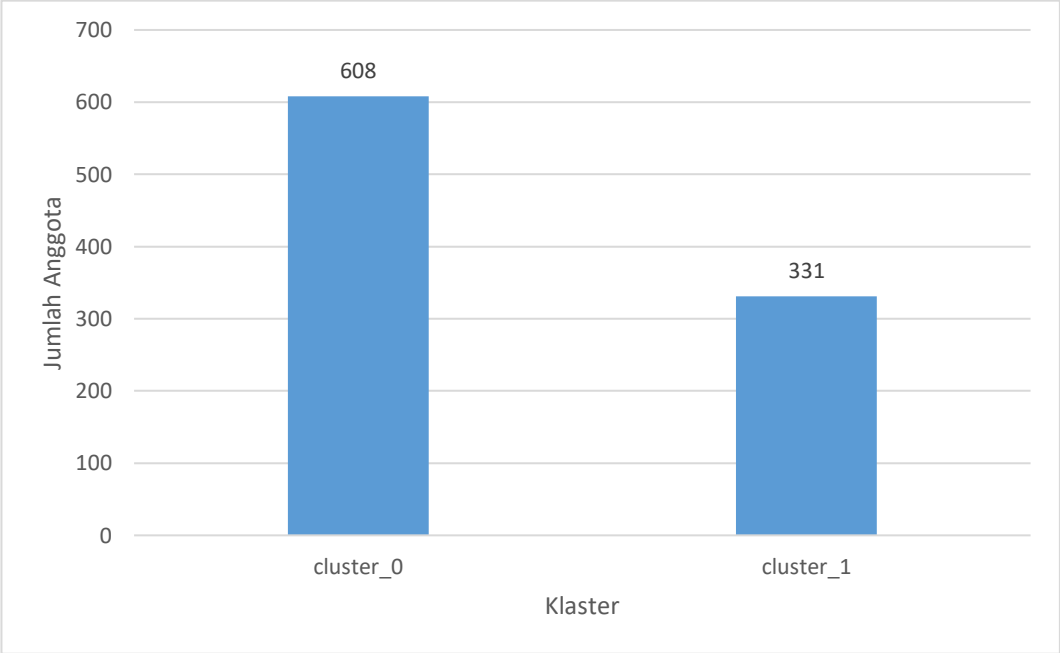
dengan algoritma K-Means pada nilai k = 2 hingga k = 10. Hasilnya yaitu klaster terbaik berada pada k di bawah 10 yaitu k = 3 [23] dan k = 9 [24].

Dari seluruh DBI pada penelitian ini (tabel 4.1 dan gambar 4.1), klaster k = 2 memiliki DBI terendah di antara seluruh DBI dengan nilai k lainnya yaitu 0,503. Karena DBI terendah menunjukkan jumlah klaster terbaik, maka penelitian ini menunjukkan bahwa *dataset* demografi fiks ini sebaiknya diklaster menjadi 2 klaster.

Pada hasil klasterisasi K-Means dengan k = 2, jumlah anggota kedua klaster ternyata berbeda. Jumlah anggota tiap klasternya ditunjukkan pada tabel 4.2 dan visualisasinya ditunjukkan pada gambar 4.2.

Tabel 4.2 Jumlah anggota tiap klaster pada k = 2

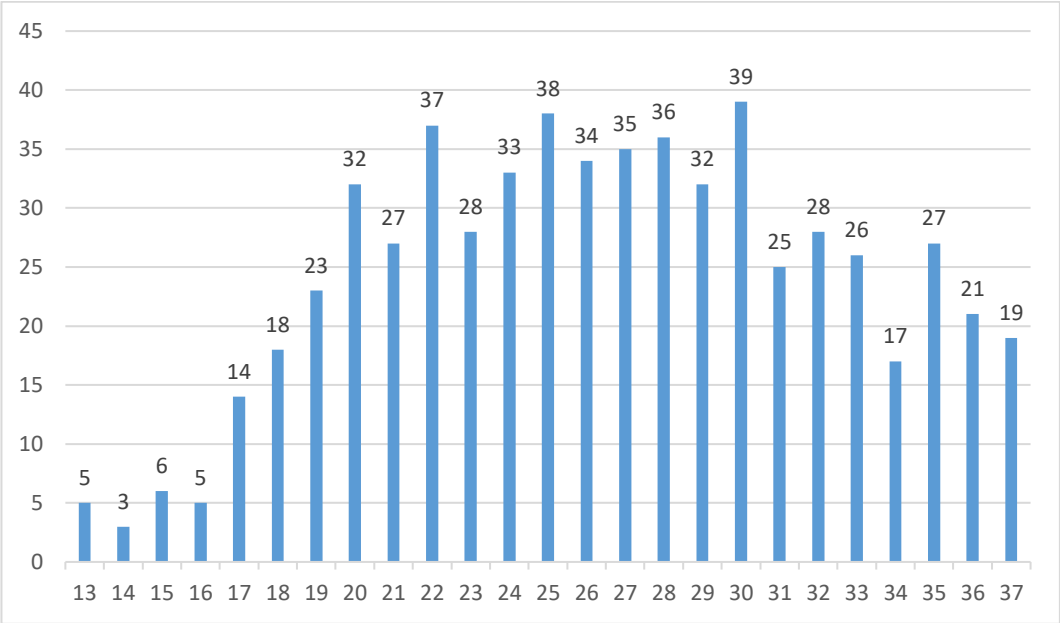
Klaster	Jumlah Anggota
Klaster 0	608
Klaster 1	331
Jumlah	939



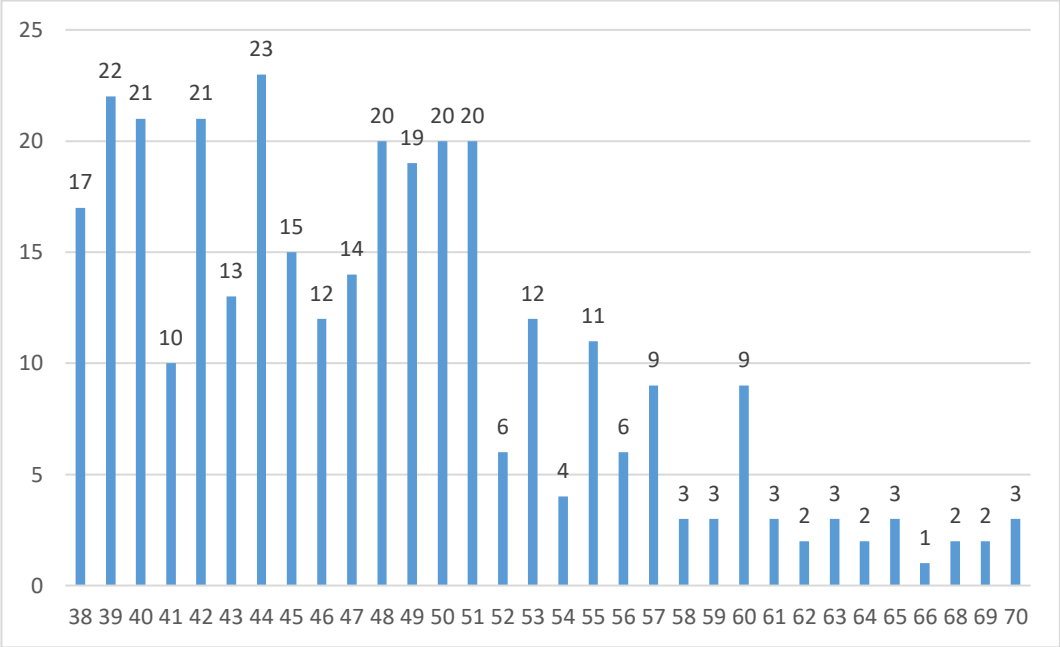
Gambar 4.2 Jumlah anggota tiap klaster pada k = 2

Pada data tersebut, klaster yang memiliki anggota terbanyak yaitu klaster 0 sebesar 608 anggota, sedangkan klaster 1 berjumlah 331 anggota. Hal ini menunjukkan bahwa jumlah anggota klaster 0 1,83 kali lipat lebih banyak daripada jumlah anggota klaster 1.

Jika keanggotaan klaster 0 dan 1 dianalisis umurnya, maka didapatkan data bahwa klaster 0 merupakan kelompok usia 13 tahun hingga 37 tahun dan klaster 1 merupakan kelompok usia 38 tahun hingga 70 tahun seperti ditunjukkan pada gambar 4.3 dan 4.4.



Gambar 4.3 Persebaran usia anggota klaster 0 K-Means

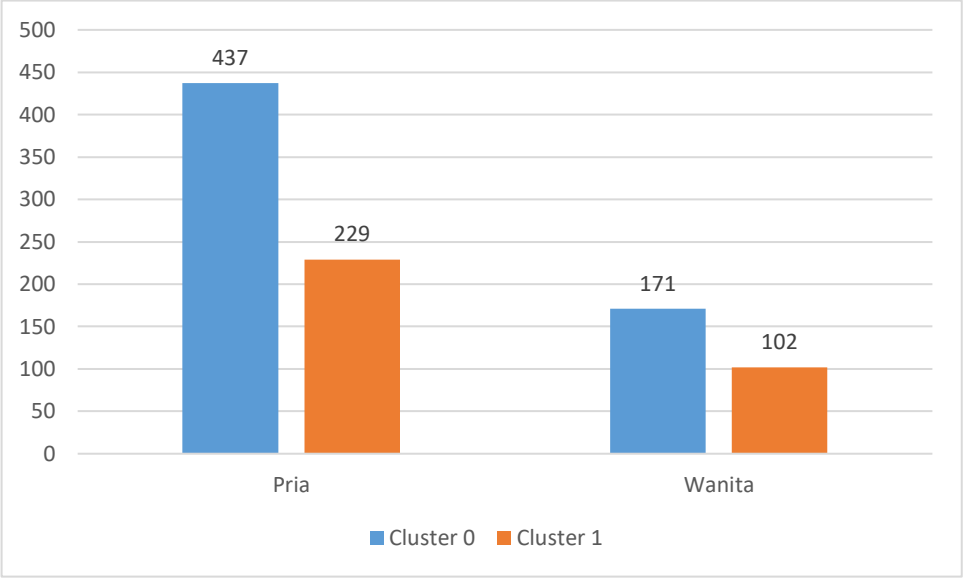


Gambar 4.4 Persebaran usia anggota klaster 1 K-Means

Pada klaster 0, pengguna dengan kelompok usia 30 tahun memiliki jumlah yang lebih banyak daripada kelompok usia lainnya yaitu 39 orang, sedangkan kelompok usia 14 tahun memiliki jumlah paling sedikit daripada kelompok usia lainnya yaitu 3 orang. Kelompok usia pengguna yang jumlahnya lebih dari 10 orang berada pada rentang umur 17 tahun hingga 37 tahun.

Pada klaster 1, pengguna dengan kelompok usia 44 tahun memiliki jumlah yang lebih banyak daripada kelompok usia lainnya yaitu 23 orang, sedangkan kelompok usia 66 tahun memiliki jumlah paling sedikit daripada kelompok usia lainnya yaitu 1 orang. Kelompok usia pengguna yang lebih besar dari 10 orang berada pada rentang umur 38 hingga 55 tahun.

Selanjutnya, data ini dianalisis jumlahnya berdasarkan jenis kelamin. Visualisasi data ini disajikan pada gambar 4.5.



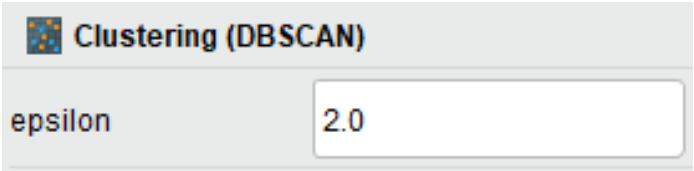
Gambar 4.5 Jumlah jenis kelamin pengguna tiap klaster

Jumlah pria pada masing – masing klaster yaitu 437 *user* (pada klaster 0) dan 171 *user* (pada klaster 1). Berarti jumlah pria di klaster 0 sebesar 1,91 kali lipat lebih banyak daripada jumlah pria di klaster 1. Jumlah wanita pada masing – masing klaster yaitu 229 *user* (pada klaster 0) dan 102 *user* (pada klaster 1). Berarti jumlah wanita di klaster 0 sebesar 1,68 kali lipat lebih banyak daripada jumlah wanita di klaster 1.

Kemudian, *dataset* anggota klaster 0 dan 1 digabung dengan ***Dataset Rating* Fiks** sehingga terbentuk *dataset* baru. *Dataset* ini diberi nama ***Dataset* K-Means Klaster 0 dan *Dataset* K-Means Klaster 1**.

4.2 Hasil Klasterisasi DBSCAN Pada *Dataset* Demografi Fiks

Selain menggunakan metode K-Means, ***Dataset Demografi* Fiks** juga diklasterisasi menggunakan algoritma DBSCAN pada RapidMiner. *Parameter* yang diatur dalam operator DBSCAN adalah *epsilon*. Pengaturan parameter tersebut ditunjukkan pada gambar 4.6.



Gambar 4.6 Pengaturan *parameter* pada DBSCAN

Jika *epsilon*nya berbeda, maka jumlah klasternya bisa sama atau berbeda. Dari beberapa percobaan dari *epsilon* 0,1 hingga 4,0 didapatkan jumlah klaster DBSCAN yang disajikan pada tabel 4.3 dan tabel 4.4.

Tabel 4.3 Jumlah klaster DBSCAN *Dataset* Demografi *Epsilon* 0,1 - 2,0

Epsilon	Jumlah Klaster
0,1	70
0,2	70
0,3	70
0,4	70
0,5	70
0,6	70
0,7	70
0,8	70
0,9	70
1,0	70

Epsilon	Jumlah Klaster
1,1	70
1,2	2
1,3	2
1,4	2
1,5	2
1,6	2
1,7	2
1,8	2
1,9	2
2,0	2

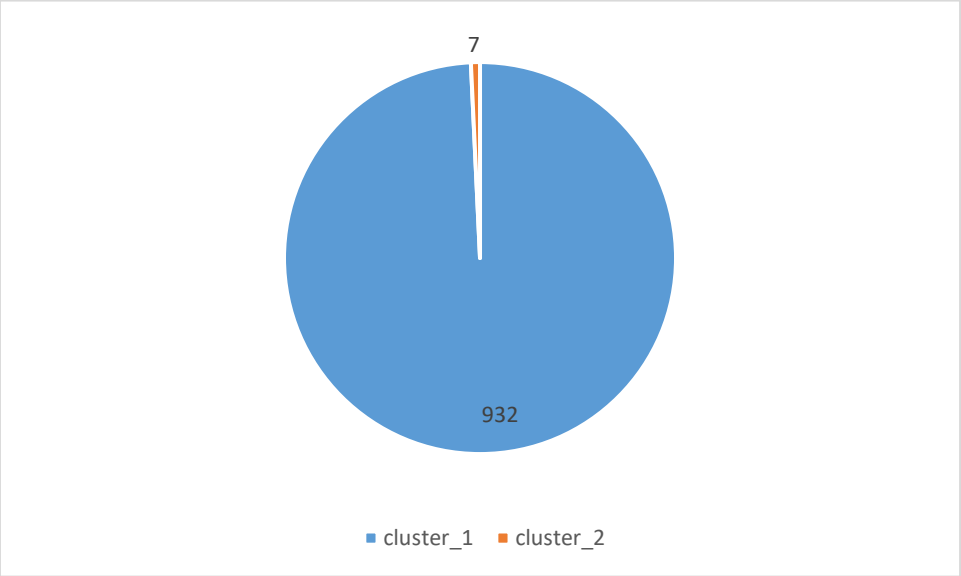
Tabel 4.4 Jumlah klaster DBSCAN *Dataset* Demografi Fiks *Epsilon* 2,1 - 3,0

Epsilon	Jumlah Klaster
2,1	2
2,2	1
2,3	1
2,4	1
2,5	1
2,6	1
2,7	1
2,8	1
2,9	1
3,0	1

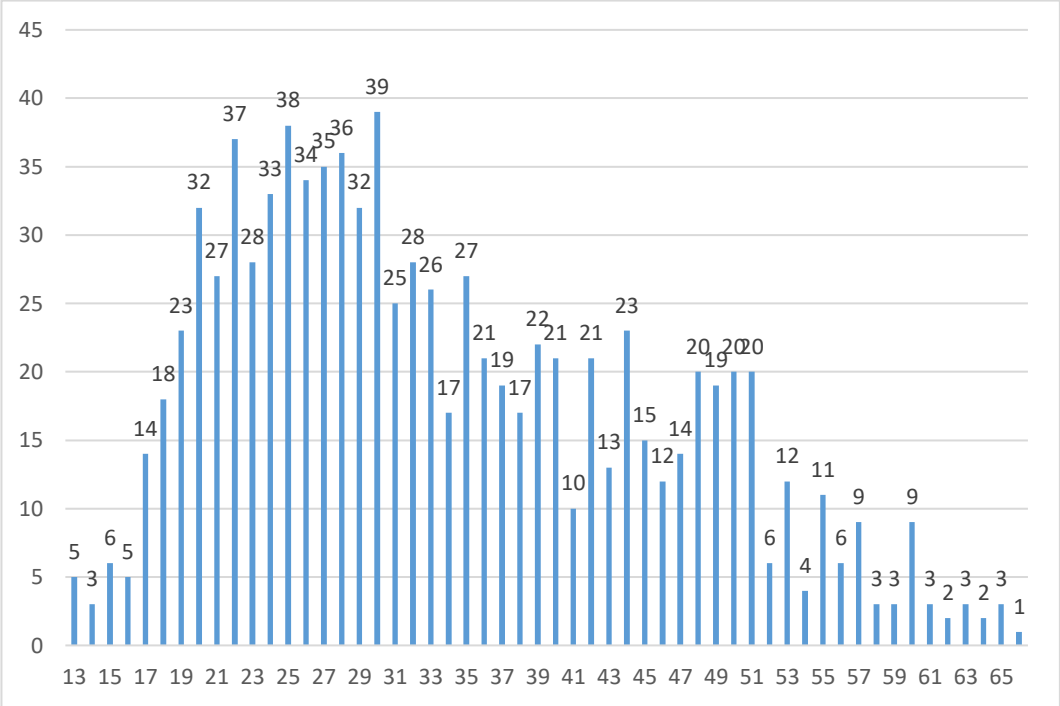
Epsilon	Jumlah Klaster
3,1	1
3,2	1
3,3	1
3,4	1
3,5	1
3,6	1
3,7	1
3,8	1
3,9	1
4,0	1

Pada hasil percobaan DBSCAN di atas (tabel 4.3 dan 4.4), jika *epsilon*nya 0,1 hingga 1,1, maka ***Dataset Demografi* Fiks** mendapatkan 70 klaster. Jika *epsilon*nya diubah menjadi 1,2 hingga 2,1 maka *dataset* ini mendapatkan 2 klaster. Jika *epsilon*nya diubah menjadi 2,2 hingga 4,0, maka *dataset* ini mendapatkan 1 klaster.

Jika dianalisis jumlah anggota 2 klaster, maka didapatkan jumlah anggota klaster yang sangat berbeda dan persebaran umurnya beragam seperti ditunjukkan pada gambar 4.7 dan 4.8.



Gambar 4.7 Jumlah anggota klaster DBSCAN *dataset* demografi lengkap



Gambar 4.8 Persebaran umur pengguna klaster 1 DBSCAN

Pada hasil berjumlah 2 klaster, jumlah anggota klaster 1 adalah 932 anggota dan jumlah anggota klaster 1 adalah 7 anggota. Jadi, jumlah anggota klaster 1 133,143 kali lebih banyak daripada jumlah anggota klaster 2. Klaster 1 memiliki anggota pengguna dengan rentang usia 13 – 66 tahun, sedangkan klaster 2 memiliki anggota pengguna dengan rentang usia 68 – 70 tahun.

Kemudian, data anggota klaster 1 dan 2 digabung dengan ***Dataset Rating* Fiks** sehingga terbentuk *dataset* baru. *Dataset* ini diberi nama ***Dataset* DBSCAN Klaster 1** dan ***Dataset* DBSCAN Klaster 2**.

4.3 Hasil *Weight Point Rank* (WP-Rank) dan *Normalized Discounted Cumulative Gain* (NDCG) Pada Hasil Klaster K-Means, Hasil Klaster DBSCAN Serta *Dataset Rating* Fiks

Selanjutnya hasil poin 4.1 dan 4.2 digabungkan dengan *rating* film yang telah dipilih oleh *user* tersebut dari *movie*1 hingga *movie*1682. Karena metode K-Means dan DBSCAN masing – masing menghasilkan 2 klaster, maka ada 4

hasil klaster yang akan digabungkan dengan *rating*nya sesuai *user* tersebut. Selain itu, *rating* sebelum klasterisasi juga digabungkan dengan data *rating*nya. Kemudian data demografinya dihapus karena yang diolah dengan metode WP-Rank hanyalah nilai *rating*nya. Dengan demikian, terdapat 5 *dataset* baru yang hanya berisi *rating* seluruh *movie*. Contoh tampilan 5 *dataset* tersebut ditunjukkan pada tabel. 4.2. *Dataset* ini yang akan diolah dengan metode WP-Rank dan dievaluasi dengan metode NDCG.

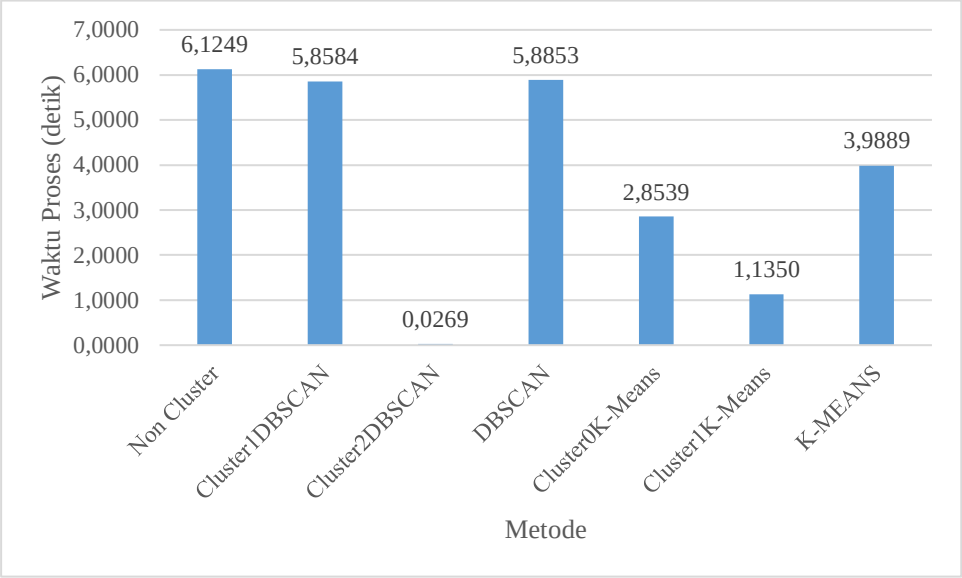
Tabel 4.5 Contoh salah satu dari 5 *dataset* yang akan diolah dengan metode WP-Rank dan dievaluasi dengan NDCG

5	3	4	3	3	5
0	0	0	0	0	0
0	0	0	0	0	0

Hasil WP-Rank dan evaluasi NDCG disajikan pada tabel 4.3. Persebaran rata – rata waktu proses ditunjukkan pada gambar 4.9.

Tabel 4.6 Perbandingan waktu proses

Percobaan	Waktu Proses (detik)				
	Sebelum Klasterisasi	Klaster1 DBSCAN	Klaster2 DBSCAN	Klaster0 K-Means	Klaster1 K-Means
1	6,2782	5,7156	0,0265	2,9547	1,0945
2	6,34977	5,7524	0,0277	2,8120	1,1520
3	6,03554	5,9657	0,0277	2,9084	1,1557
4	5,83519	5,8077	0,0247	2,8237	1,1066
5	5,95676	5,7603	0,0227	2,9296	1,1391
6	6,31154	5,7148	0,0264	2,8627	1,1327
7	6,09765	5,7306	0,0250	2,8071	1,1543
8	6,27581	6,0428	0,0280	2,7553	1,1213
9	6,01965	6,2622	0,0256	2,8299	1,1595
10	6,08906	5,8314	0,0347	2,8559	1,1345
Rata2 Waktu Proses	6,1249	5,8584	0,0269	2,8539	1,1350



Gambar 4.9 Persebaran rata – rata waktu proses

Nilai rata – rata waktu yang diperlukan untuk memproses seluruh *rating* sebelum klasterisasi sebesar 6,1249 detik. Ini adalah waktu proses dari 939 x 1682 = 1.579.398 *rating*. Setelah klasterisasi, klaster 1 DBSCAN (932 x 1682 = 1.567.624 *rating*) memiliki waktu proses rata – rata sebesar 5,8584 detik, sedangkan rata – rata waktu proses klaster 2 DBSCAN (7 x 1682 = 11.774 rating) hanya 0,0269 detik. Jika rata – rata waktu proses kedua klaster DBSCAN dihitung, maka hasilnya 5,8853 detik atau lebih cepat 0,2397 detik daripada sebelum klasterisasi. Jika dibandingkan keduanya, maka DBSCAN lebih cepat 1,0407 kali lipat daripada waktu proses data yang belum diklaster. Maka bisa disimpulkan bahwa klasterisasi DBSCAN dapat mempercepat waktu proses seluruh *rating* / mengatasi *scalability*.

Setelah klasterisasi, klaster 0 K-Means (memiliki 608 x 1682 = 1.022.656 *rating*) memiliki waktu proses rata – rata sebesar 2,8539 detik, sedangkan rata – rata waktu proses klaster 1 K-Means (331 x 1682 = 556.742 *rating*) hanya 1,1350 detik. Jika rata – rata waktu proses kedua klaster K-Means dihitung, maka hasilnya 3,9889 detik atau lebih cepat 2,1360 detik daripada sebelum klasterisasi. Jika dibandingkan keduanya, maka K-Means lebih cepat 1,5355 kali lipat daripada waktu proses data yang belum diklaster. Maka bisa

disimpulkan bahwa klasterisasi mengunakan K-Means dapat mempercepat waktu proses seluruh *rating* / mengatasi *scalability*.

Jika dibandingkan ke 3 waktu rata – rata prosesnya, maka K-Means lebih unggul dibandingkan dengan DBSCAN dan tanpa klasterisasi. Hal ini menunjukkan bahwa K-Means dapat mengatasi *scalability* dalam *Collaborative Filtering*.

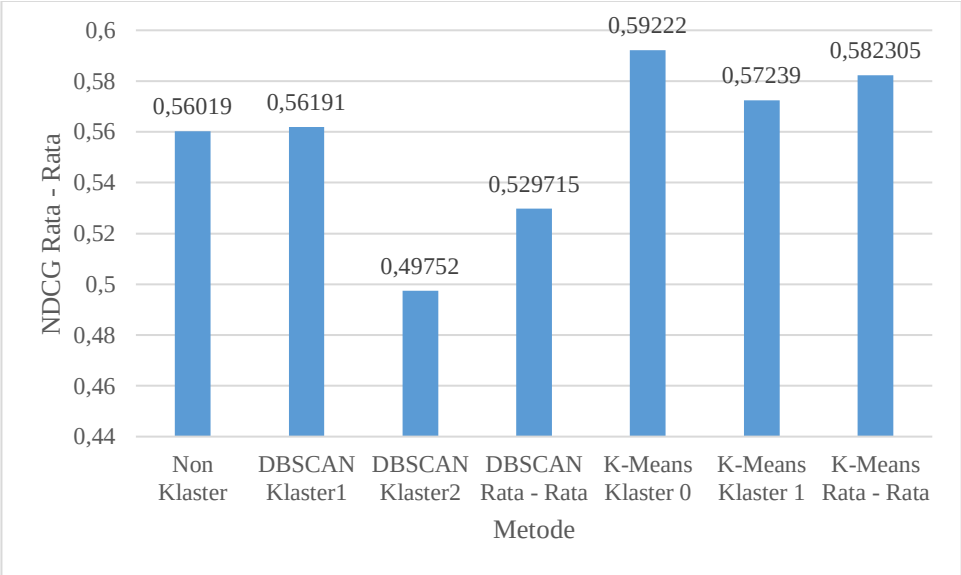
Selanjutnya, semua metode dalam penelitian ini dievaluasi dengan metode NDCG. Hasil evaluasi NDCG berdasarkan klaster ditunjukkan pada tabel 4.7. Nilai rata – rata hasil NDCG berdasarkan klaster tersebut ditunjukkan pada tabel 4.8 serta gambar 4.10.

Tabel 4.7 Hasil evaluasi NDCG berdasarkan klaster

Proses Ke	NDCG				
	Non Klaster	DBSCAN Klaster 1	DBSCAN Klaster 2	K-Means Klaster 0	K-Means Klaster 1
1	0,4925	0,4951	0,2396	0,5425	0,4101
2	0,4737	0,4751	0,348	0,4983	0,4537
3	0,4712	0,4729	0,3877	0,5054	0,485
4	0,4994	0,5012	0,4492	0,5173	0,52
5	0,5276	0,5293	0,4959	0,5516	0,5581
6	0,5591	0,5606	0,5405	0,5926	0,5948
7	0,5893	0,5908	0,5829	0,6262	0,6228
8	0,6242	0,6259	0,619	0,6597	0,6573
9	0,6583	0,6602	0,6429	0,6964	0,6968
10	0,7066	0,708	0,6695	0,7322	0,7253
Rata – Rata	0,56019	0,56191	0,49752	0,59222	0,57239

Tabel 4.8 Nilai rata – rata NDCG berdasarkan klaster

Metode	NDCG
Non Klaster	0,56019
DBSCAN Klaster1	0,56191
DBSCAN Klaster2	0,49752
DBSCAN Rata - Rata	0,529715
K-Means Klaster 0	0,59222
K-Means Klaster 1	0,57239
K-Means Rata - Rata	0,582305



Gambar 4.10 Perbandingan nilai NDCG berdasarkan klaster

Nilai rata – rata NDCG Metode DBSCAN ternyata menurun dibandingkan dengan NDCG non klaster sebesar 0,030475 yaitu dari 0,56019 menjadi 0,529715. Sedangkan nilai rata – rata NDCG Metode K-Means Klaster lebih baik / meningkat daripada NDCG non klaster sebesar 0,022115 yaitu dari 0,56019 menjadi 0,582305. Ini menunjukkan bahwa *dataset* ini lebih cocok diklaster menggunakan metode K-Means daripada metode DBSCAN. Dengan demikian, K-Means dan WP-Rank terbukti dapat mengatasi *scalability*. K-Means lebih unggul daripada DBSCAN dalam hasil penelitian ini.

4.4 Hasil Imputasi Pada Hasil Klaster K-Means dan Hasil Klaster DBSCAN

Pada hasil 4.3, *dataset* penggabungan hasil klasterisasi dengan *rating* film, selanjutnya dihilangkan s*parsity*nya dengan menggunakan metode imputasi (sesuai metode 3.9) kemudian dibandingkan hasilnya. Hasilnya ditunjukkan pada tabel 4.9.

Tabel 4.9 Hasil olah data *sparsity* menggunakan RapidMiner

User Id	***movie1***	***movie2***	...	...	***movie1681***	***movie1682***
1	5	3			3	3
2	4	3			3	3
3	4	3			3	3
…						
…						
941	3	3			3	3
942	4	3			3	3
943	4	3			3	3

Pada tabel 4.9, data yang kosong / tidak memiliki nilai, telah diubah menjadi nilai rata – rata pada kolom tersebut. Hasilnya terdapat satu macam bilangan yaitu bilangan bulat. Selanjutnya, hasil olah data ini dibandingkan dengan perhitungan nilai rata – rata kolom tersebut menggunakan Microsoft Excel. Perbandingannya ditunjukkan pada tabel 4.7.

Tabel 4.10 Perbandingan hasil olah data menggunakan RapidMiner dengan olah data menggunakan Microsoft Excel

	5		
3,917	3,210	3,076	3,552
4	5	3	4

Baris pertama merupakan baris nilai *dataset* sebelum diolah. Baris kedua merupakan perhitungan rata – rata nilai masing – masing nilai *rating movie Id* menggunakan Microsoft Excel. Baris ketiga berisi nilai olah data menggunakan RapidMiner untuk mengatasi *sparsity*.

Pada kolom pertama, *dataset* aslinya kosong. Dengan *RapidMiner* dihasilkan angka 4. Dengan *Microsoft Excel*, dihasilkan angka 3,917. Berarti, nilai 3,917 dibulatkan menjadi 4 oleh *RapidMiner*. Ini merupakan pembulatan ke atas. Maka menurut *RapidMiner*, *dataset* yang kosong pada kolom ini dapat diisi dengan angka 4.

Pada kolom kedua, *dataset* aslinya memiliki nilai rating 5. Dengan *Microsoft Excel* dihasilkan 3,210. Dengan *RapidMiner* dihasilkan angka 5. Berarti *RapidMiner* tidak mengubah nilai yang telah ada.

Selanjutnya, pada kolom ketiga, *dataset* aslinya kosong. Dengan *RapidMiner* dihasilkan angka 3. Dengan *Microsoft Excel*, dihasilkan angka 3,076. Berarti, nilai 3,076 dibulatkan menjadi 3 oleh *RapidMiner*. Ini merupakan pembulatan ke bawah. Maka menurut *RapidMiner*, dataset yang kosong pada kolom ini dapat diisi dengan angka 3.

Selanjutnya pada kolom ke 4, *dataset* aslinya kosong. Dengan *RapidMiner* dihasilkan angka 4. Dengan *Microsoft Excel*, dihasilkan angka 3,552. Berarti, nilai 3,552 dibulatkan menjadi 4 oleh *RapidMiner*. Ini merupakan pembulatan ke atas. Maka menurut *RapidMiner*, *dataset* yang kosong pada kolom ini dapat diisi dengan angka 4.

Maka, dari semua kolom tersebut dapat disimpulkan bahwa *RapidMiner* akan menghasilkan bilangan bulat yang merupakan pembulatan terdekat dari rata – rata nilai di kolom tersebut. Pembulatannya bisa ke atas atau ke bawah yaitu pembulatan ke bilangan bulat terdekat.

Selanjutnya, pada beberapa kolom, *dataset* asli tidak memiliki nilai *rating* alias kosong atau *sparsity*.

Pada ***Dataset* K-Means Klaster 0 Imputasi**, sebelum *sparsity* diatasi, terdapat 953.795 data yang kosong. Setelah diatasi *sparsity* tersebut menggunakan *RapidMiner*, terdapat 56 kolom yang masih terdapat *sparsity*nya. Jika 1 kolom terdapat 608 data, maka kolom yang masih kosong sebanyak 56 kolom x 608 data = 34.048 data. Dengan demikian, data yang terisi sebesar 953. 795 – 34.048 = 919.747 data. Persentase data yang masih kosong sebesar 34.048 / 953. 795 x 100 % = 3,570 %. Sedangkan persentase data yang telah terisi sebesar 919.140 / 953. 795 x 100% = 96,43 %.

Pada ***Dataset* K-Means Klaster 1 Imputasi**, sebelum *sparsity* diatasi, terdapat 525.760 data yang kosong. Setelah diatasi *sparsity* tersebut menggunakan *RapidMiner*, terdapat 207 kolom yang masih terdapat *sparsity*nya. Jika 1 kolom terdapat 331 data, maka kolom yang masih kosong sebanyak 207 kolom x 331 data = 68.517 data. Dengan demikian, data yang terisi sebesar 525.760 – 68.517 = 457.243 data. Persentase data yang masih kosong sebesar 68.517 / 525.760 x 100% = 13,032%. Sedangkan persentase data yang telah terisi sebesar 68.517 / 525.760 x 100% = 86,968%.

Pada ***Dataset* DBSCAN Klaster 1 Imputasi**, sebelum *sparsity* diatasi, terdapat 1.468.170 data yang kosong. Setelah diatasi *sparsity* tersebut menggunakan *RapidMiner*, tidak terdapat kolom yang masih terdapat *sparsity*nya. Dengan demikian, data yang terisi sebesar 1.468.170 – 0 = 1.468.170 data. Persentase data yang masih kosong sebesar 0 / 1.468.170 x 100 % = 0%. Sedangkan persentase data yang telah terisi sebesar 1.468.170 / 1.468.170 x 100% = 100%.

Pada ***Dataset* DBSCAN Klaster 2 Imputasi**, sebelum *sparsity* diatasi, terdapat 11.385 data yang kosong. Setelah diatasi *sparsity* tersebut menggunakan *RapidMiner*, terdapat 1.380 kolom yang masih terdapat *sparsity*nya. Jika 1

kolom terdapat 7 data, maka kolom yang masih kosong sebanyak 1.380 kolom x 7 data = 9.660 data. Dengan demikian, data yang terisi sebesar 11.385 – 9.660 = 1.725 data. Persentase data yang masih kosong sebesar 9.660 / 11.385 x 100% = 84,848%, sedangkan persentase data yang telah terisi sebesar 1.725 / 11.385 x 100% = 15,152%.

Ke-empat data di atas dapat dilihat juga pada tabel 4.11.

Tabel 4.11 Perbandingan hasil klasterisasi dan *sparsity*

Dataset	**K-Means Klaster 0 Imputasi**	**K-Means Klaster 1 Imputasi**	**DBSCAN Klaster 1 Imputasi**	**DBSCAN Klaster 2 Imputasi**
Jumlah Null	953.795	525.760	1.468.170	11.385
Jumlah Null Teratasi	919.747	457.243	1.468.170	1.725
Persentase Null Teratasi	96,43	86,968	100	15,152
Jumlah Null Tidak Teratasi	34.048	68.517	0	9.660
Persentase Null Tidak Teratasi	3,570	13,032	0	84,848

Dari data tersebut, *RapidMiner* dapat mengatasi 100% *sparsity* pada DBSCAN Klaster 1 akan tetapi *RapidMiner* hanya dapat mengatasi 15,152% *sparsity* pada DBSCAN Klaster 2. RapidMiner dapat mengatasi K-Means Klaster sebesar 86,968% dan 96,43%. Karena itu, *dataset* ini diklasterisasi kemudian dihilangan *sparsity*nya lebih baik dengan algoritma K-Means daripada DBSCAN.

Hasil ini menunjukkan bahwa *sparsity* pada *dataset* Movielens 100k dapat diatasi dengan memanfaatkan fitur-fitur dan algoritma yang tersedia di *RapidMiner* yaitu *Replace Operator Missing Value* dan K-Means.

4.5 Hasil *Weight Point Rank* (WP-Rank) dan *Normalized Discounted Cumulative Gain* (NDCG) Pada Hasil Imputasi Klaster K-Means dan Hasil Imputasi Klaster DBSCAN

Data kosong pada hasil 4.4 selanjutnya digantikan dengan angka nol agar nantinya dapat diketahui rankingnya dengan metode WP-Rank dan NDCG menggunakan Matlab. Hasilnya disajikan pada tabel 4.12.

Tabel 4.12 Perbandingan waktu proses **DBSCAN Klaster Imputasi** dan **K-Means Klaster Imputasi** dan Non Klaster Non Imputasi

Percobaan	Waktu Proses (detik)				
	Non Klaster & Non Imputasi	DBSCAN Klaster 1 Imputasi	DBSCAN Klaster 2 Imputas	K-Means Klaster 0 Imputasi	K-Means Klaster 1 Imputasi
1	6,071	6,505	0,345	3,022	1,172
2	5,999	6,413	0,060	3,062	1,271
3	5,794	6,109	0,041	2,987	1,237
4	5,844	6,132	0,036	2,902	1,212
5	5,852	6,217	0,045	2,925	1,216
6	5,794	6,265	0,033	2,984	1,189
7	6,572	6,279	0,031	3,053	1,193
8	5,901	6,304	0,035	3,004	1,180
9	6,001	6,175	0,037	2,968	1,200
10	5,871	6,163	0,036	3,011	1,222
Rata2 Waktu Proses	5,970	6,256	0,070	2,992	1,209

Dari data di atas, dapat diketahui bahwa DBSCAN Klaster 1 memiliki waktu proses lebih lama daripada waktu proses klaster lainnya. DBSCAN Klaster 2 memiliki waktu proses lebih cepat daripada waktu proses lainnya. Akan tetapi,

jika dihitung rata – rata waktu proses kedua klaster, rata – rata waktu proses K-Means klaster adalah (2,992 detik + 1,209 detik) / 2 = 2,101 detik. Sedangkan DBSCAN memiliki rata – rata waktu proses sebesar (6,256 detik + 0,070 detik) / 2 = 3,163 detik. Maka, rata – rata waktu proses K-Means lebih cepat daripada DBSCAN.

Jika waktu proses non klaster dan non imputasi dibandingkan dengan rata – rata waktu proses klaster DBSCAN imputasi, maka terdapat selisih sebesar 5,970 detik - 3,163 detik = 2,807 detik. Berarti klaster DBSCAN imputasi mampu mempercepat waktu proses sebesar 2,807 detik.

Jika waktu proses non klaster dan non imputasi dibandingkan dengan rata – rata waktu proses klaster K-Means imputasi, maka terdapat selisih sebesar 5,970 detik - 2,101 detik = 3,869 detik. Berarti klaster DBSCAN imputasi mampu mempercepat waktu proses sebesar 3,869 detik.

Selanjutnya, 4 *dataset* klaster imputasi dievalusi serta dibandingkan dengan *dataset* non klaster non imputasi. Hasilnya disajikan pada tabel 4.13.

Tabel 4.13 Perbandingan hasil evaluasi NDCG berdasarkan imputasi dan non klaster non imputasi

Proses Ke	NDCG				
	Non Klaster & Non Imputasi	DBSCAN Klaster 1	DBSCAN Klaster 2	K-Means Klaster 0	K-Means Klaster 1
1	0,493	1	1	1	1
2	0,474	1	1	1	1
3	0,471	1	1	1	1
4	0,499	1	1	1	1
5	0,528	1	1	1	1
6	0,559	1	1	1	1
7	0,589	1	1	1	1
8	0,624	1	1	1	1
9	0,658	1	1	1	1

10	0,707	1	1	1	1
Rata – Rata	0,560	1	1	1	1

Dataset non klaster dan non imputasi memiliki nilai – rata NDCG sebesar 0,560. Sedangkan seluruh klaster imputasi memiliki NDCG sebesar 1. Berarti terdapat peningkatan NDCG sebesar 1 – 0,560 = 0,44. Hal ini menunjukkan bahwa klasterisasi dan imputasi dapat meningkatkan nilai NDCG.