BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil K-Means Clustering pada Dataset Movielens OK

Pada Dataset Movielens OK, klasterisasi dilakukan menggunakan algoritma K-Means dengan nilai k mulai dari 2 hingga 5 untuk mencari nilai Davies Bouldin Index (DBI) terkecil. DBI terkecil merupakan salah satu indikator jumlah klaster terbaik pada suatu dataset. Hasilnya dapat dilihat dalam tabel berikut.

Tabel 4.1 Nilai DBI k = 2 hingga k = 5

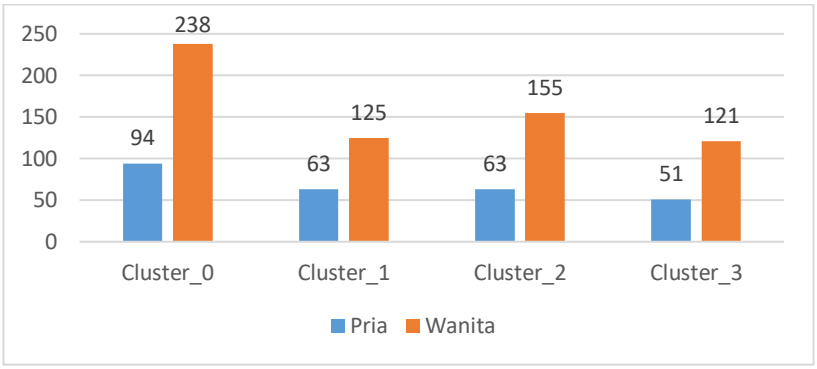
k	DBI
2	0.803
3	0.846
4	0.788
5	0.855

Berdasarkan tabel di atas, nilai Davies Bouldin Index terkecil terdapat pada k = 4, yakni 0.788. Berdasarkan semua nilai Davies-Bouldin Index (DBI) dalam penelitian ini, klaster dengan nilai k = 4 menunjukkan nilai DBI paling rendah dibandingkan dengan nilai k lainnya, yakni sebesar 0,788. Karena DBI yang lebih rendah menandakan jumlah klaster yang optimal, langkah berikutnya dalam penelitian ini merekomendasikan pengelompokan dataset Movielens OK ini menjadi 4 klaster. Data jumlah masing-masing anggota cluster dapat dilihat pada tabel di bawah.

Tabel 4.2 Jumlah Anggota Cluster

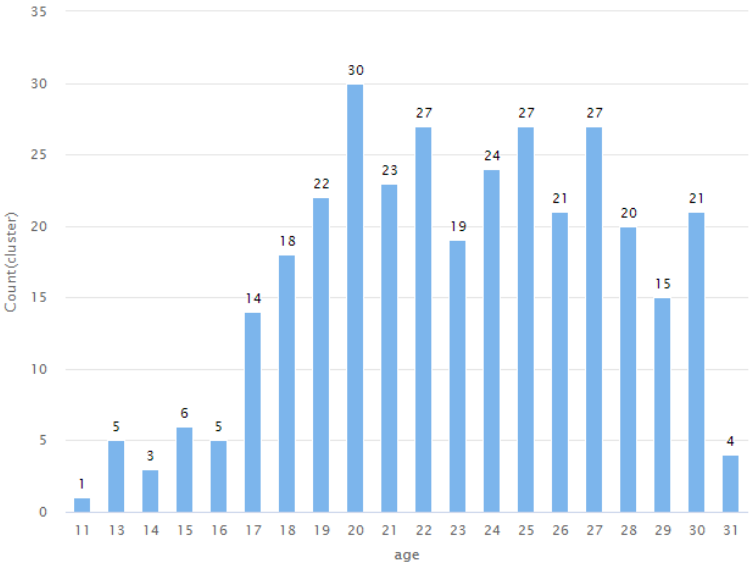
Cluster	Jumlah Anggota
Cluster_0	332
Cluster_1	188
Cluster_2	218
Cluster_3	172

Pada data tersebut, klaster dengan anggota terbanyak yaitu pada cluster_0 sebanyak 332 anggota, sedangkan klaster dengan anggota terkecil yaitu pada cluster_3 dengan 172 anggota.



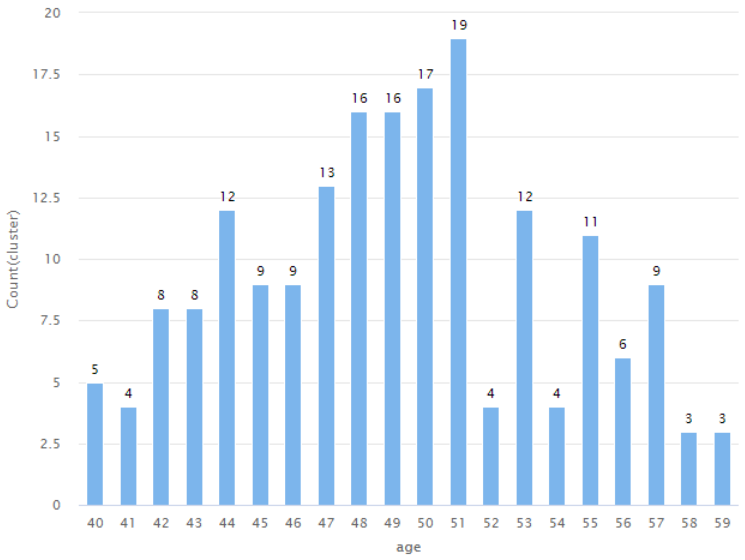
Gambar 4.1 Jenis Kelamin pada Setiap Cluster

Jumlah keseluruhan pria yaitu 271, dengan rincian 94 (cluster_0), 63 (cluster_1 dan cluster_2), serta 51 (cluster_3). Berdasarkan gambar di atas dapat terlihat bahwa pada jumlah wanita selalu lebih banyak dibandingkan pria pada setiap cluster. Persebaran usia masing-masing klaster dapat dilihat pada gambar di bawah.



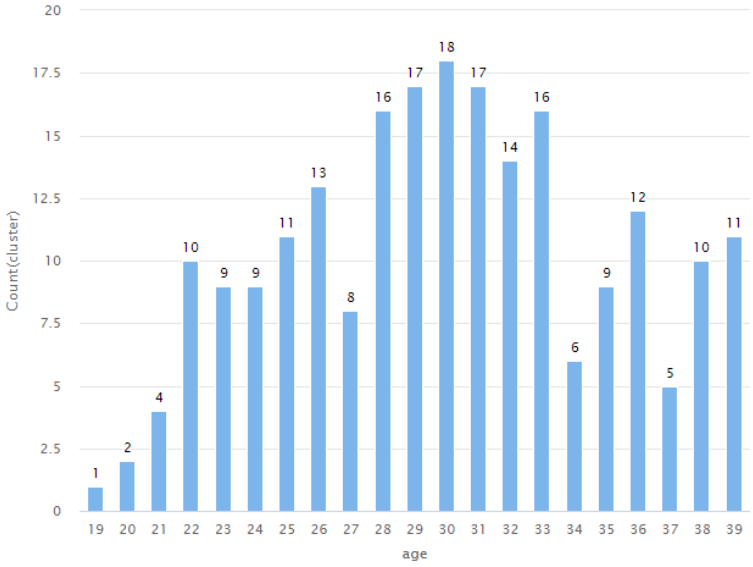
Gambar 4.2 Persebaran Usia Cluster_0

Berdasarkan gambar di atas, dapat terlihat bahwa pada cluster_0 terdapat 30 *user* berusia 20 dan merupakan yang terbanyak pada kelompok ini. Klaster ini berisi anggota dengan kelompok usia dengan rentang 11 hingga 31 tahun. Apabila dilakukan analisis lebih lanjut, data pada cluster_0 mayoritas berisi anggota dengan usia 18 sampai dengan 30 tahun.



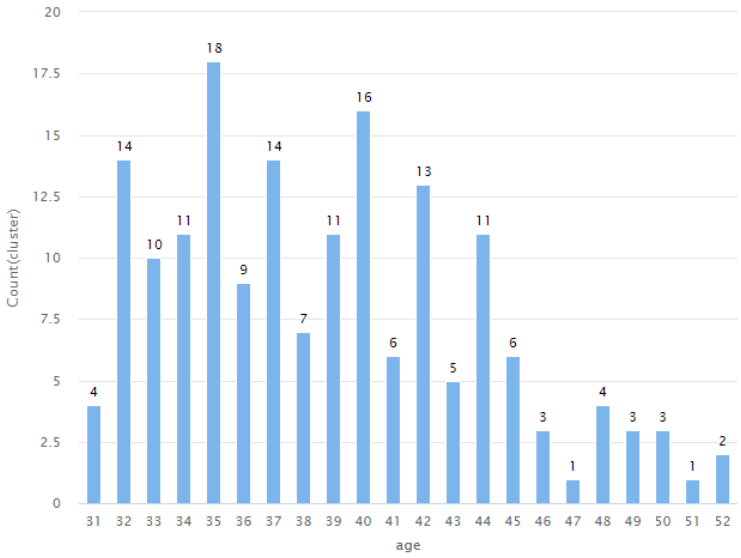
Gambar 4.3 Persebaran Usia Cluster_1

Berdasarkan gambar di atas, terlihat bahwa di cluster_1 terdapat 19 pengguna yang berusia 51 tahun, yang merupakan jumlah terbesar dalam kelompok ini. Cluster ini terdiri dari individu dengan usia antara 40 hingga 59 tahun. Jika diteliti lebih lanjut, mayoritas data dalam cluster_1 menunjukkan anggota dengan usia berkisar antara 44 hingga 55 tahun.



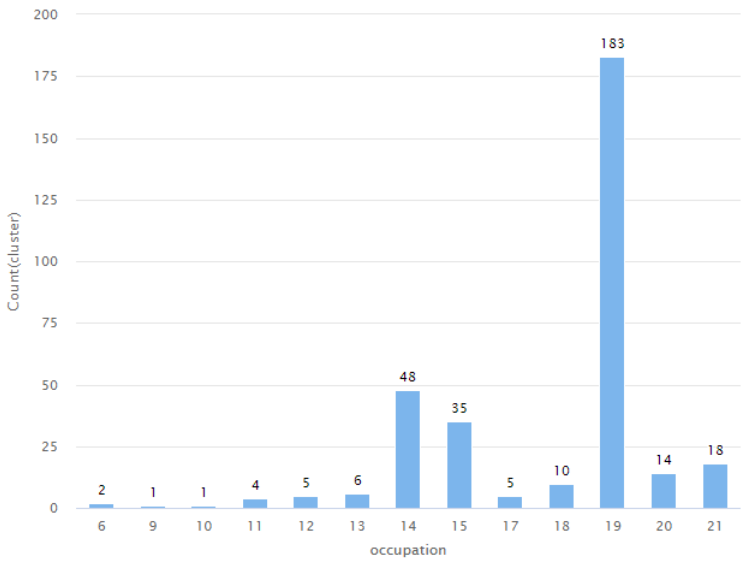
Gambar 4.4 Persebaran Usia Cluster_2

Dari gambar di atas, terlihat bahwa di cluster_2 terdapat 18 pengguna yang berusia 30 tahun, jumlah terbanyak dalam kelompok tersebut. Cluster ini terdiri dari individu dengan usia antara 19 hingga 39 tahun. Jika dianalisis lebih lanjut, mayoritas data dalam cluster_2 menunjukkan anggota dengan usia berkisar antara 26 hingga 33 tahun.



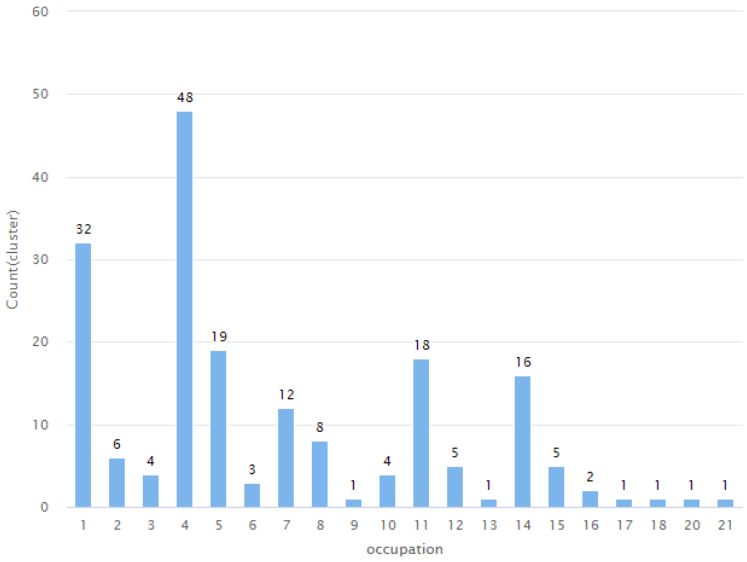
Gambar 4.5 Persebaran Usia Cluster_3

Dari gambar di atas, terlihat bahwa di cluster_3 terdapat 18 pengguna yang berusia 35 tahun, jumlah terbanyak dalam kelompok tersebut. Cluster ini terdiri dari individu dengan usia antara 31 hingga 52 tahun. Jika dianalisis lebih lanjut, mayoritas data dalam cluster_3 menunjukkan anggota dengan usia berkisar antara 32 hingga 40 tahun.



Gambar 4.6 Persebaran Pekerjaan Cluster_0

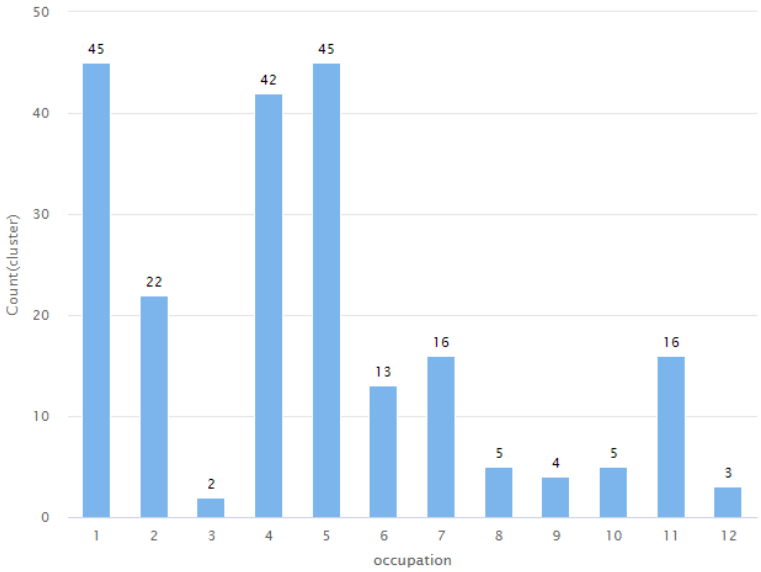
Pada gambar di atas, dapat dilihat bahwa pada cluster_0 terdapat 183 pengguna sebagai pelajar, yang merupakan jumlah terbanyak pada cluster ini. Cluster ini terdiri dari berbagai pekerjaan dengan kode nomor 6, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21 (entertainment, homemaker, lawyer, librarian, marketing, none, other, programmer).



Gambar 4.7 Persebaran Pekerjaan Cluster_1

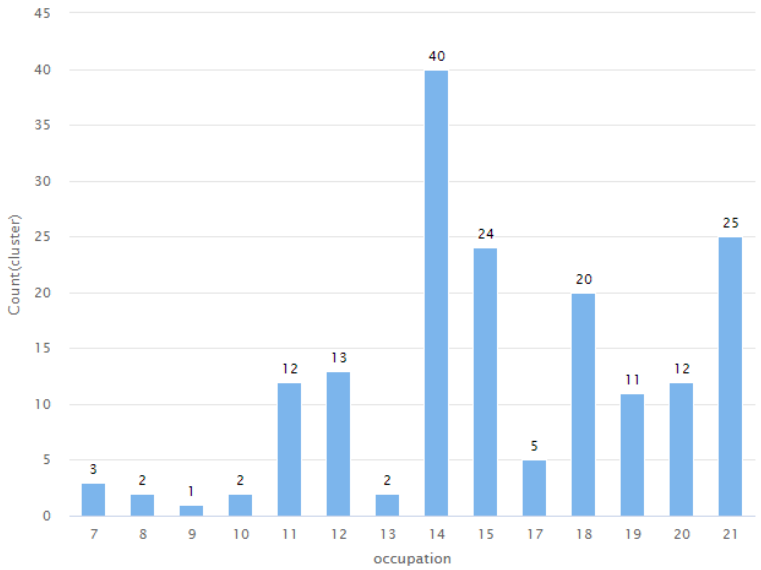
Pada gambar di atas, dapat dilihat bahwa pada cluster_1 terdapat 48 pengguna sebagai educator (pengajar), yang merupakan jumlah terbanyak pada cluster ini. Cluster ini terdiri dari berbagai pekerjaan dengan kode nomor 1, 2, 3, 4,

5, 6, 7, 8, 9, 10, 11, 12,13, 14, 15, 16, 17, 18, 20, 21 (administrator, artist, doctor, educator, engineer, entertainment, executive, healthcare, homemaker, lawyer, librarian, marketing, none, other, programmer, retired, salesman, scientist, technician, writer).



Gambar 4.8 Persebaran Pekerjaan Cluster_2

Pada gambar di atas, dapat dilihat bahwa pada cluster_2 terdapat 45 pengguna sebagai administrator dan engineer, yang merupakan jumlah terbanyak pada cluster ini. Cluster ini terdiri dari berbagai pekerjaan dengan kode nomor 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 (administrator, artist, doctor, educator, engineer, entertainment, executive, healthcare, homemaker, lawyer, librarian, marketing).



Gambar 4.9 Persebaran Pekerjaan Cluster_3

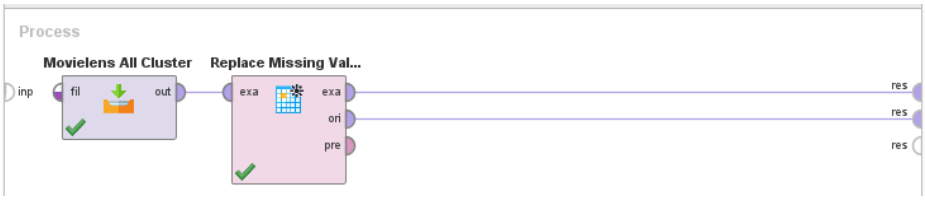
Pada gambar di atas, terlihat bahwa mayoritas pekerjaan pada cluster_3 yaitu other (pekerjaan selain kode 1 sampai dengan 21). Pada cluster ini mencakup berbagai pekerjaan, yaitu dari kode nomor 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21 (executive, healthcare, homemaker, lawyer, librarian, marketing, none, other, programmer, salesman, scientist, technician, writer).

Selanjutnya, data anggota dari setiap kelompok cluster disatukan dengan Dataset Movielens OK, menciptakan dataset baru dan diberi penamaan Dataset Movielens All Cluster. Selain itu, dibuat juga dataset baru yang berisi data masing-masing cluster, dengan penamaan (Data cluster_0), (Data Cluster_1), (Data Cluster_2), (Data Cluster_3) untuk kebutuhan pada tahapan selanjutnya.

4.2 Imputation Data Cluster

Setelah klasterisasi sebanyak 4 cluster, langkah berikutnya adalah melakukan imputation pada rating-rating yang masih kosong dengan menggunakan fungsi average pada tools RapidMiner. Terdapat dua pendekatan yang akan diterapkan pada tahap ini. Pendekatan pertama adalah melakukan imputation secara keseluruhan ketika data klaster masih bercampur antara cluster satu dan cluster lainnya. Sementara itu, pendekatan kedua dilakukan imputation nilai pada setiap cluster secara terpisah.

1. Imputation pada semua Cluster secara Menyeluruh
   Pada tahap ini, dilakukan imputation secara menyeluruh menggunakan dataset Movielens All Cluster. Imputation ini dilakukan menggunakan fungsi average yang ada pada tools RapidMiner.



Gambar 4.10 Model Imputation pada Dataset Movielens All Cluster

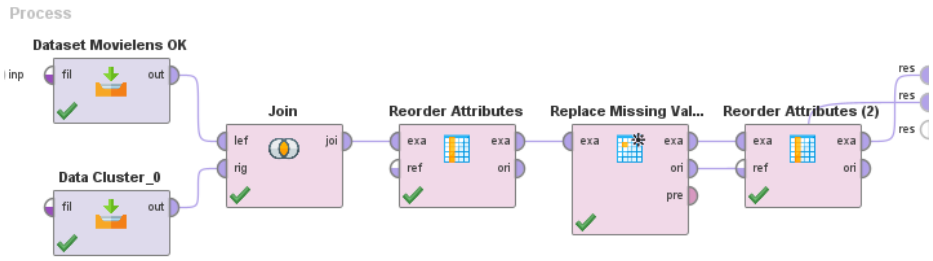
Setelah dilakukan imputation, maka nilai yang sebelumnya kosong sudah terisi sepenuhnya. Dataset Movielens All Cluster yang telah dilakukan imputation dapat dilihat pada tabel berikut.

Tabel 4.3 Dataset Movielens All Cluster + Imputation

Id	*age*	*gender*	*occupation*	*cluster*	*movie1*	*movie2*	*...*	*...*	*movie1681*	*movie1682*
1	24	2	20	Cluster_0	5	3	…	…	3	3
2	53	1	14	Cluster_1	4	4	…	…	4	4
…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…
909	48	1	11	Cluster_1	4	4	…	…	3	3
910	22	2	19	Cluster_0	4	5	…	…	3	3

Proses imputation ini dilakukan untuk mengatasi masalah *sparsity* dan *cold start.* Selanjutnya, dataset yang sudah melalui proses imputation disimpan ke dalam file baru dan diberi nama Movielens All Cluster Imputation untuk diproses pada tahap selanjutnya.

2. Imputation Per Cluster
   Pada tahap ini, dilakukan imputation pada masing-masing cluster secara terpisah.



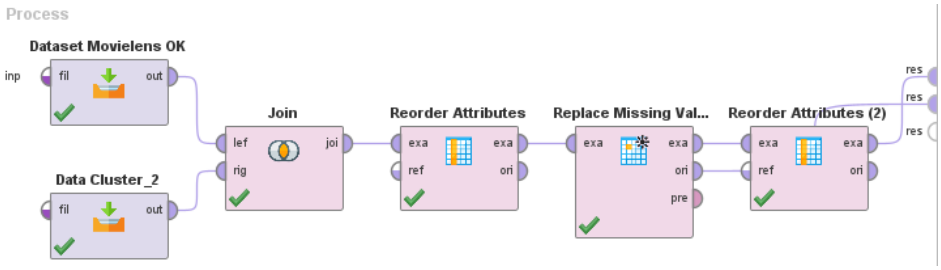
Gambar 4.11 Model Imputation pada Data Cluster_0

Pada gambar di atas, dilakukan penggabungan antara Data Cluster_0 yang berisi data demografi user dengan Dataset Movielens OK yang berisi data user beserta rating menggunakan operator Join. Selanjutnya dilakukan proses imputation menggunakan fungsi *average* untuk mengatasi *missing value*. Data Cluster_0 yang sudah dilakukan imputation dapat dilihat pada tabel berikut.

Tabel 4.4 Data Cluster_0 + Imputation

Id	*age*	*gender*	*occupation*	*cluster*	*movie1*	*movie2*	*...*	*...*	*movie1681*	*movie1682*
1	24	2	20	Cluster_0	5	3	…	…	4	3
3	23	2	21	Cluster_0	4	4	…	…	4	2
…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…
908	20	2	19	Cluster_0	5	4	…	…	4	3
910	22	2	19	Cluster_0	4	5	…	…	4	3

Setelah dilakukan imputation, maka nilai yang sebelumnya kosong sudah terisi sepenuhnya. Selanjutnya, dataset yang sudah melalui proses imputation disimpan ke dalam file baru dan diberi nama Cluster_0 Imputation untuk diproses pada tahap selanjutnya. Proses yang sama dilakukan pada klaster berikutnya.



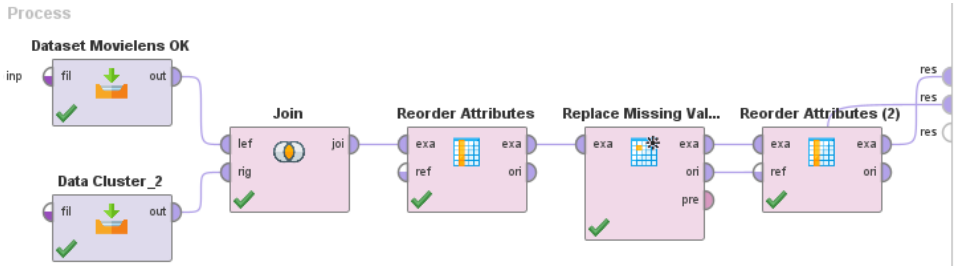
Gambar 4.12 Model Imputation pada Data Cluster_1

Pada gambar di atas, dilakukan penggabungan antara Data Cluster_1 yang berisi data demografi user dengan Dataset Movielens OK yang berisi data user beserta rating menggunakan operator Join. Selanjutnya dilakukan proses imputation menggunakan fungsi *average* untuk mengatasi *missing value*. Data Cluster_1 yang sudah dilakukan imputation dapat dilihat pada tabel berikut.

Tabel 4.5 Data Cluster_1 +Imputation

Id	*age*	*gender*	*occupation*	*cluster*	*movie1*	*movie2*	*...*	*...*	*movie1681*	*movie1682*
2	53	1	14	Cluster_1	4	4	…	…	4	4
6	23	2	21	Cluster_1	4	4	…	…	3	3
…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…
904	48	2	4	Cluster_1	5	4	…	…	3	3
909	48	1	11	Cluster_1	3	4	…	…	3	3

Setelah dilakukan imputation, maka nilai yang sebelumnya kosong sudah terisi sepenuhnya. Selanjutnya, dataset yang sudah melalui proses imputation disimpan ke dalam file baru dan diberi nama Cluster_1 Imputation untuk diproses pada tahap selanjutnya. Proses yang sama dilakukan pada klaster berikutnya.



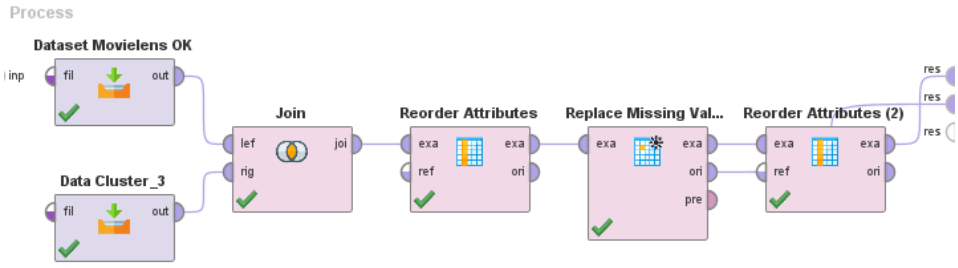
Gambar 4.13 Model Imputation pada Data Cluster_2

Pada gambar di atas, dilakukan penggabungan antara Data Cluster_2 yang berisi data demografi user dengan Dataset Movielens OK yang berisi data user beserta rating menggunakan operator Join. Selanjutnya dilakukan proses imputation menggunakan fungsi *average* untuk mengatasi *missing value*. Data Cluster_2 yang sudah dilakukan imputation dapat dilihat pada tabel berikut.

Tabel 4.6 Data Cluster_2 + Imputation

Id	*age*	*gender*	*occupation*	*cluster*	*movie1*	*movie2*	*...*	*...*	*movie1681*	*movie1682*
8	36	2	1	Cluster_2	4	4	…	…	1	3
16	21	2	6	Cluster_2	4	3	…	…	1	3
…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…
891	29	1	1	Cluster_2	5	3	…	…	1	3
907	32	2	1	Cluster_2	4	4	…	…	1	3

Setelah dilakukan imputation, maka nilai yang sebelumnya kosong sudah terisi sepenuhnya. Selanjutnya, dataset yang sudah melalui proses imputation disimpan ke dalam file baru dan diberi nama Cluster_2 Imputation untuk diproses pada tahap selanjutnya. Proses yang sama dilakukan pada klaster berikutnya.



Gambar 4.14 Model Imputation pada Data Cluster_3

Pada gambar di atas, dilakukan penggabungan antara Data Cluster_3 yang berisi data demografi user dengan Dataset Movielens OK yang berisi data user beserta rating menggunakan operator Join. Selanjutnya dilakukan proses imputation menggunakan fungsi *average* untuk mengatasi *missing value*. Data Cluster_3 yang sudah dilakukan imputation dapat dilihat pada tabel berikut.

Tabel 4.7 Data Cluster_3 + Imputation

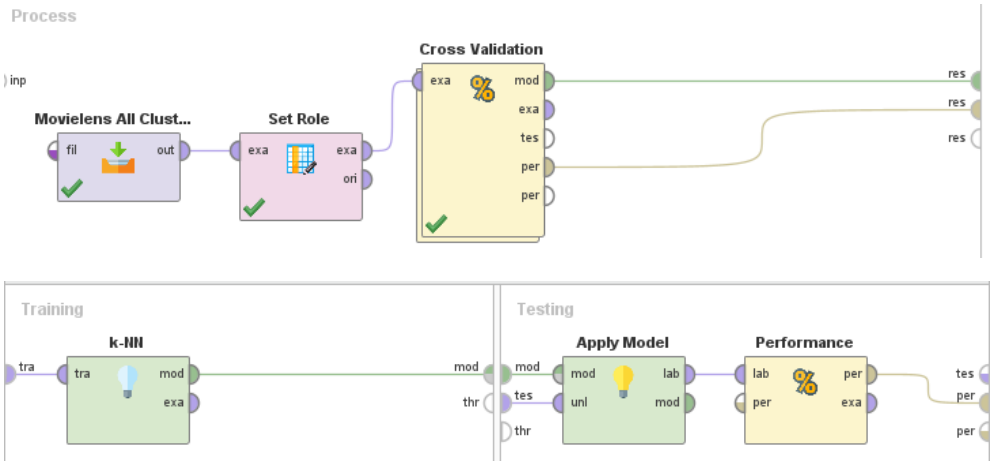
Id	*age*	*gender*	*occupation*	*cluster*	*movie1*	*movie2*	*...*	*...*	*movie1681*	*movie1682*
5	33	1	14	Cluster_3	4	3	…	…	1	3
11	39	1	14	Cluster_3	2	5	…	…	1	3
…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…
898	44	2	18	Cluster_3	3	4	…	…	1	3
905	38	1	20	Cluster_3	4	4	…	…	1	3

Setelah dilakukan imputation, maka nilai yang sebelumnya kosong sudah terisi sepenuhnya. Selanjutnya, dataset yang sudah melalui proses imputation disimpan ke dalam file baru dan diberi nama Cluster_3 Imputation untuk diproses pada tahap selanjutnya.

Setelah masing-masing cluster selesai dilakukan imputation, maka data setelah imputation tersebut dijadikan satu file excel untuk diproses pada tahapan selanjutnya. Data ini diberi nama Data Cluster Imputation.

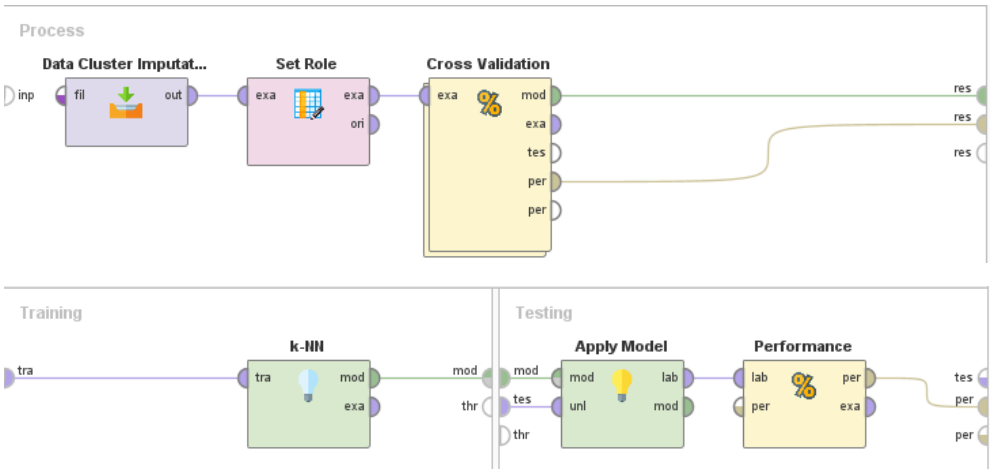
4.3 Klasifikasi KNN

Setelah melalui tahap clustering dan imputation dengan dua pendekatan yang berbeda, langkah selanjutnya yaitu melakukan klasifikasi menggunakan algoritma KNN untuk mengetahui performa dan akurasi yang didapatkan. Dalam hal ini, label yang digunakan yaitu Cluster. Pada penelitian ini digunakan Cross Validation dengan 10 folds untuk menguji tingkat keberhasilan klasifikasi. Penggunaan 10 folds dikarenakan 10 folds merupakan nilai yang optimal dan memiliki kesalahan klasifikasi paling kecil [26]. Model klasifikasi menggunakan KNN dapat dilihat pada gambar di bawah.



Gambar 4.15 Model Klasifikasi Movielens All Cluster Imputation menggunakan KNN

Pada gambar di atas, dilakukan pembuatan model menggunakan dataset Movielens All Cluster. Selanjutnya, dilakukan pengaturan menggunakan operator Set Role untuk mengatur atribut cluster sebagai label. Lalu dilakukan pengujian menggunakan operator Cross Validation dengan 10 fold yang di dalamnya terdapat operator KNN sebagai algoritma yang digunakan untuk dilihat performanya.



Gambar 4.16 Model Klasifikasi Data Cluster Imputation menggunakan KNN

Pada gambar di atas, dilakukan pembuatan model menggunakan dataset Data Cluster Imputation. Selanjutnya, dilakukan pengaturan menggunakan operator Set Role untuk mengatur atribut cluster sebagai label. Lalu dilakukan pengujian menggunakan operator Cross Validation dengan 10 fold yang di dalamnya terdapat operator KNN sebagai algoritma yang digunakan untuk dilihat performanya.

Selanjutnya, hasil akurasi yang didapatkan pada masing-masing pendekatan yang telah dilakukan dapat dilihat pada tabel berikut.

Tabel 4.8 Perbandingan Akurasi menggunakan KNN

No	Status	Akurasi
1	Imputation Cluster secara Menyeluruh	80.11%
2	Imputation Per Cluster	100%

Tabel 4.8 menunjukkan hasil perbandingan performa akurasi yang diuji menggunakan algoritma KNN terkait model imputation pada data klaster secara menyeluruh dan model imputation pada data klaster secara terpisah. Pada percobaan tersebut didapatkan hasil performa akurasi sebesar 80.11% saat imputation dilakukan pada data klaster secara menyeluruh, dan didapatkan hasil akurasi sebesar 100% saat imputation dilakukan pada masing-masing klaster secara terpisah. Imputation yang dilakukan pada masing-masing cluster secara terpisah mendapatkan akurasi yang lebih tinggi dan sempurna dibandingkan imputation yang dilakukan saat semua cluster masih menjadi satu. Hal ini dikarenakan imputation pada masing-masing cluster akan memberikan rekomendasi imputation yang lebih spesifik sesuai dengan preferensi anggota cluster.

Selain menggunakan algoritma KNN, dilakukan juga pengujian klasifikasi menggunakan algoritma naïve bayes sebagai pembanding. Adapun pendekatan yang dilakukan yaitu dilakukan percobaan imputation nilai yang kosong berdasarkan nilai rata-rata pada data klaster secara menyeluruh dan juga pada masing-masing klaster untuk dilihat perbandingan hasil dan performa akurasinya menggunakan algoritma naïve bayes. Hasil perbandingan performa akurasi yang diuji menggunakan algoritma naïve bayes terkait model imputation pada data klaster secara menyeluruh dan model imputation pada data klaster secara terpisah ditunjukkan pada tabel di bawah.

Tabel 4.9 Perbandingan Akurasi menggunakan Naïve Bayes

No	Status	Akurasi
1	Imputation Cluster secara Menyeluruh	30.99%
2	Imputation Per Cluster	100%

4.4 Evaluasi Confusion Matrix

Berdasarkan langkah-langkah yang telah dilakukan, didapatkan Confusion Matrix dari pengujian menggunakan Cross Validation 10 fold dari dua pendekatan. Confusion Matrix dari masing-masing pendekatan dapat dilihat pada tabel di bawah.

Tabel 4.10 Confusion Matrix Imputation Cluster secara Menyeluruh

	true cluster_0	true cluster_1	true cluster_3	true cluster_2	class precision
pred. cluster_0	308	0	26	19	87.25%
pred. cluster_1	0	152	24	10	81.72%
pred. cluster_3	10	26	103	23	63.58%
pred. cluster_2	14	10	19	166	79.43%
class recall	92.77%	80.85%	59.88%	76.15%	

Berdasarkan tabel di atas, dapat dilihat pada Pred. cluster_0 menunjukkan presisi tertinggi sebesar 87.25% dan juga memiliki recall tertinggi sebesar 92.77% untuk True cluster_0, menandakan bahwa model sangat efektif dalam mengidentifikasi data yang termasuk dalam cluster_0.

Di sisi lain, Pred. cluster_3 memiliki presisi terendah sebesar 63.58%, menunjukkan bahwa prediksi untuk cluster ini kurang akurat. Selain itu, True cluster_3 memiliki recall terendah yaitu 59.88%, menunjukkan bahwa model kesulitan dalam mengenali data yang seharusnya berada dalam cluster ini. Secara keseluruhan, model ini memiliki akurasi sebesar 80.11%.

Tabel 4.11 Confusion Matrix Imputation Per Cluster

	true cluster_0	true cluster_1	true cluster_2	true cluster_3	class precision
pred. cluster_0	332	0	0	0	100.00%
pred. cluster_1	0	188	0	0	100.00%
pred. cluster_2	0	0	218	0	100.00%
pred. cluster_3	0	0	0	172	100.00%
class recall	100.00%	100.00%	100.00%	100.00%	

Pada tabel di atas dapat dilihat bahwa Presisi untuk setiap cluster mencapai 100%, yang berarti bahwa setiap data yang diprediksi masuk dalam suatu cluster memang benar-benar berada di cluster tersebut. Recall untuk setiap cluster juga mencapai 100%, menunjukkan bahwa model berhasil memprediksi dengan benar semua data

yang sebenarnya termasuk dalam setiap cluster. Model clustering menunjukkan kinerja yang sempurna dengan precision dan recall mencapai 100% untuk setiap cluster. Ini menunjukkan bahwa model dapat mengelompokkan data dengan akurasi yang sangat tinggi, tanpa membuat kesalahan dalam prediksi.

Berdasarkan hasil yang diperoleh, imputasi yang dilakukan secara terpisah pada setiap cluster menghasilkan akurasi yang lebih tinggi dan sempurna dibandingkan dengan imputasi yang dilakukan saat semua cluster digabung menjadi satu. Hal ini disebabkan karena imputasi pada masing-masing cluster memberikan rekomendasi yang lebih spesifik sesuai dengan preferensi anggota cluster.