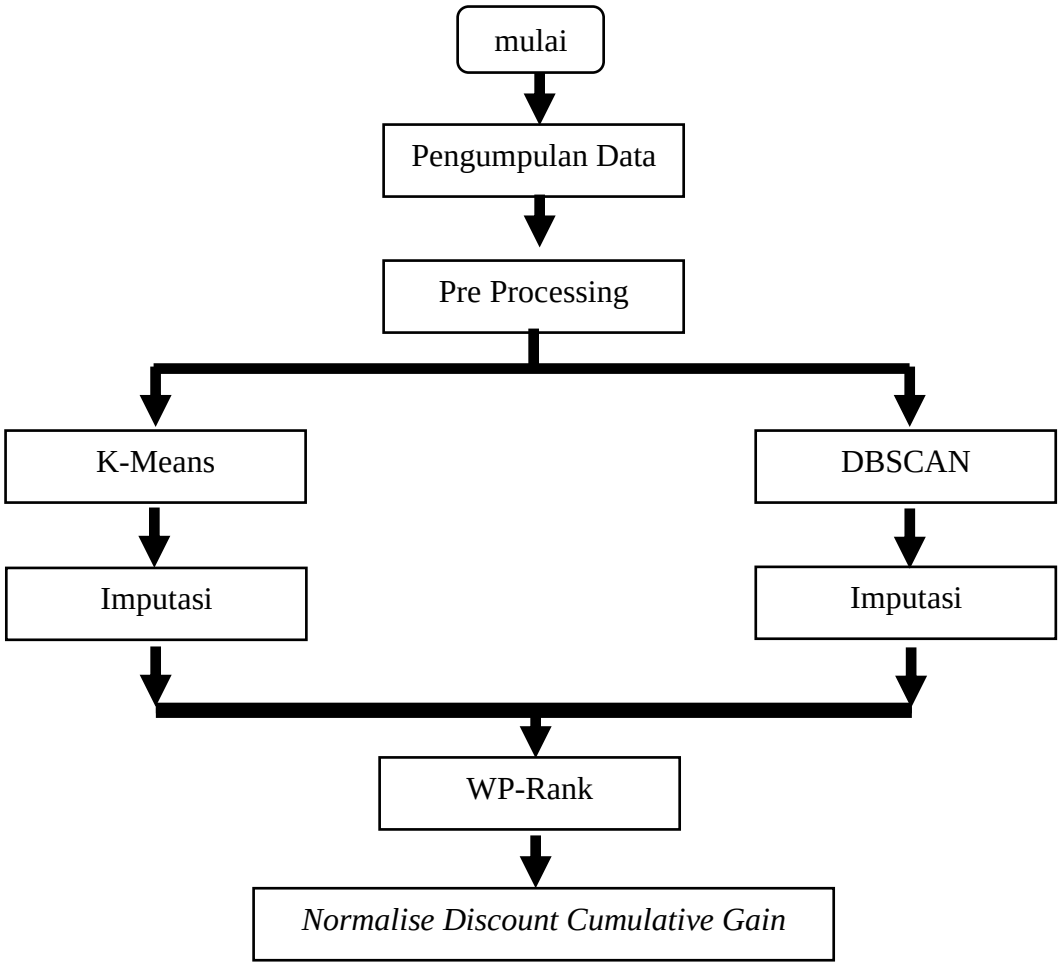
BAB III METODOLOGI PENELITIAN

Penelitian ini bertujuan untuk mengatasi *scalability* dan *sparsity* pada *Collaborative Filtering* dengan pendekatan *clustering* (K-Means dan DBSCAN) dan *ranking based* (WP-Rank). Hasil dari proses ini yang selanjutnya sebagai dasar pemberikan rekomendasi ke *user* / pengguna yang sesuai dengan preferensinya. Untuk lebih jelasnya dapat dilihat pada Gambar 3.1.



Gambar 3.1 Langkah - langkah penelitian

3.1 Pengumpulan Data

Penelitian ini menggunakan data publik film dari website movielens.org khususnya *dataset* 100k yang memiliki 100.000 data. *Dataset* ini dapat didownload melaluli link : https://grouplens.org/datasets/movielens/100k/. *Dataset* ini dirilis pada bulan April 1998. *Dataset* tersebut dikumpulkan melalui situs web MovieLens (movielens.umn.edu) selama tujuh bulan sejak tanggal 19 September 1997 sampai 22 April 1998. Data ini telah dibersihkan dari data pengguna / *user* yang memiliki *rating* kurang dari 20 buah film atau tidak memiliki demografi yang lengkap. Kumpulan data ini terdiri dari 100.000 rating dari angka 1 - 5 dari 943 pengguna / *user* terhadap 1682 buah film / *movie*. Data yang digunakan pada penelitian ini yaitu data demografi pengguna dan data *rating* terhadap 1682 film. Deskripsi, atribut dan jumlah *record*-nya disajikan pada tabel 3.1.

Tabel 3.1 Deskripsi, atribut dan jumlah *record dataset*

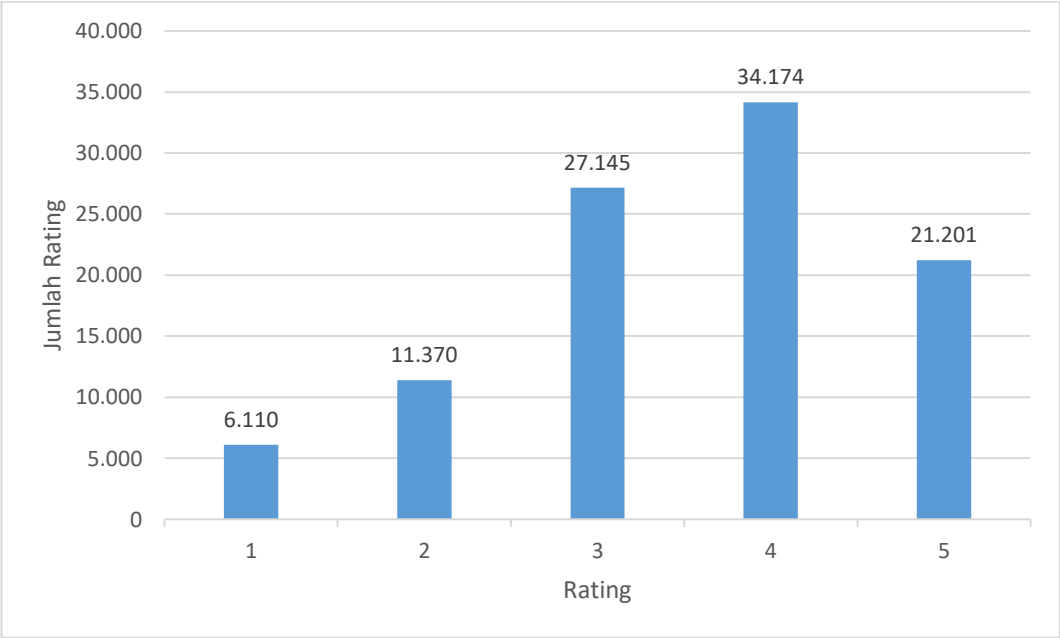
Nama *File Dataset*	Deskripsi *File Dataset*	Atribut	Jumlah *Record*
u.data	Kumpulan data 100.000 *rating* film oleh 943 pengguna pada 1682 film	Id Pengguna (*User Id*) 1 - 943 Id Film (*Movie Id*) 1 - 1692 Rating Film 1 – 5	943 1.682 100.000
u.user	Informasi demografis tentang 943 pengguna yang memberikan rating terhadap 1682 film	Id Pengguna (*User Id*) 1 - 943 Umur User (*Age*) 7 - 73 Jenis Kelamin User (Pria / *Male* dan Wanita (*Female*) Pekerjaan User (*Occupation*) 21 macam pekerjaan *Zipcode* (Kode Pos Rumah)	943 943 943 943 943 923

Dataset rating film berisi 100.000 *rating* film dari *rating* bernilai 1 hingga 5. Semakin besar nilainya menunjukkan semakin besar kesukaan / minat pengguna terhadap film tersebut. Dari *dataset* ini, masing – masing *rating* memiliki jumlah yang bervariasi antara 6.110 - 34.174. Jumlah *rating* paling sedikit adalah 6.110 pada *rating* 1 dan jumlah *rating* paling banyak adalah

34.174 pada *rating* 4. Data lengkap jumlah masing – masing *rating* disajikan pada tabel 3.2. Visualisasinya disajikan pada gambar 3.2.

Tabel 3.2 Jumlah persebaran *rating*

Rating	Jumlah *Rating*
1	6.110
2	11.370
3	27.145
4	34.174
5	21.201
Jumlah *Rating*	100.000



Gambar 3.2 Persebaran *rating* 1682 film oleh 943 pengguna pada *Dataset Movielens* 100k

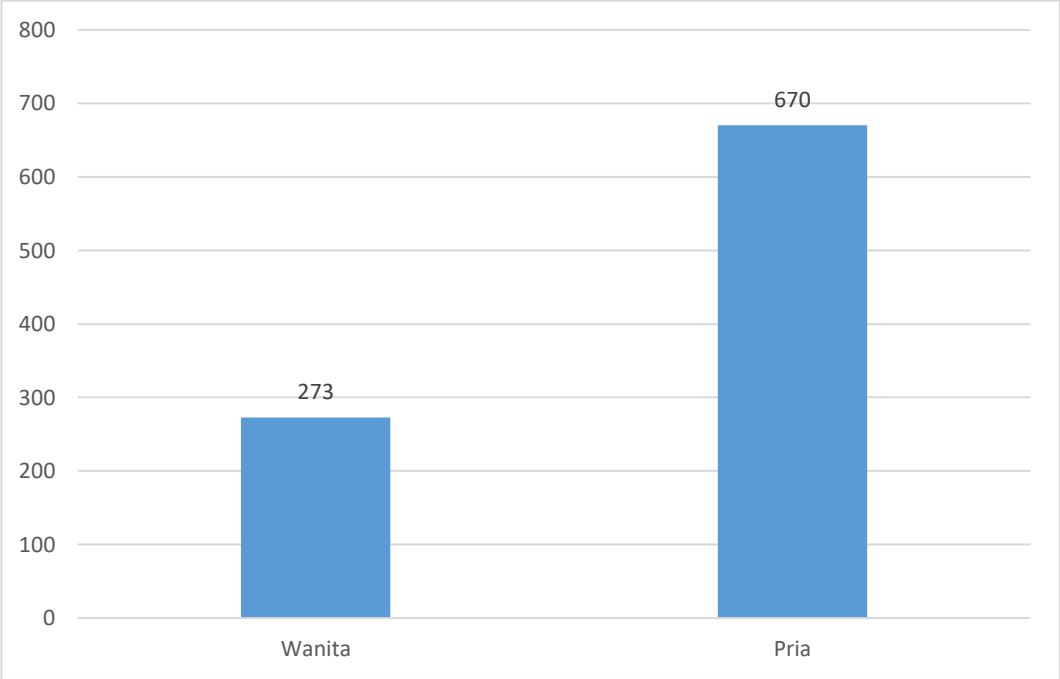
Pengguna (*user*) yang memberikan *rating* terhadap 1682 film, memiliki umur yang beragam. Umur terendah / termuda pengguna adalah 7 tahun, sedangkan umur tertingginya / tertua adalah 73 tahun. *User* yang memiliki umur yang sama terbanyak adalah user dengan umur 30 tahun sejumlah 39 user. *User* yang hanya berjumlah 1 pada umur tersebut yaitu pada umur 7, 10, 11, 66 dan 73 tahun. Persebaran umur (*age*) dari 943 pengguna disajikan pada tabel 3.3. Dari

943 pengguna yang memberikan rating terhadap 1682 film, terdapat 670 pria dan 273 wanita. Visualisasi jumlahnya disajikan pada gambar 3.3

Tabel 3.3 Persebaran umur 943 pengguna pada *Dataset Movielens* 100k

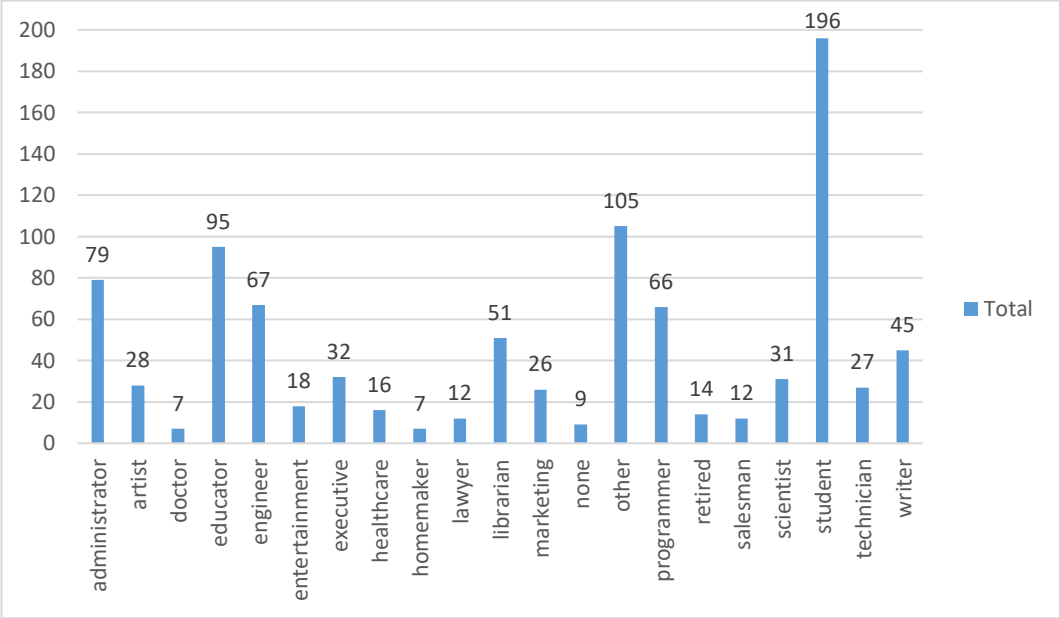
Umur	Jumlah
7	1
10	1
11	1
13	5
14	3
15	6
16	5
17	14
18	18
19	23
20	32
21	27
22	37
23	28
24	33
25	38
26	34
27	35
28	36
29	32
30	39
31	25
32	28
33	26
34	17
35	27
36	21
37	19
38	17
39	22
40	21

Umur	Jumlah
41	10
42	21
43	13
44	23
45	15
46	12
47	14
48	20
49	19
50	20
51	20
52	6
53	12
54	4
55	11
56	6
57	9
58	3
59	3
60	9
61	3
62	2
63	3
64	2
65	3
66	1
68	2
69	2
70	3
73	1



Gambar 3.3 Persebaran jenis kelamin 943 *user* pada *Dataset Movielens* 100k

Seluruh pengguna (943 *user*) yang memberikan *rating* terhadap 1682 film, memiliki pekerjaan (*occupation*) yang beragam. Terdapat 21 macam pekerjaan semua pengguna. Pengguna terbanyak memiliki pekerjaan sebagai pelajar (*student*) sebanyak 196 orang. Pengguna paling sedikit yaitu 7 orang memiliki pekerjaan dokter (*doctor*) dan pembuat rumah (*home maker*). Visualisasi persebaran pekerjaan pengguna ditunjukkan pada gambar 3.4.



Gambar 3.4 Persebaran pekerjaan 943 *user* pada *Dataset Movielens* 100k

Dari 943 pengguna yang memberikan *rating* terhadap 1682 film pada *dataset* ini, terdapat 2 *user* yang tidak memberikan kode pos rumahnya / tempat tinggalnya dan 18 *user* memberikan kode pos yang salah yaitu berupa huruf. Selain itu yaitu 923 pengguna memberikan kode pos rumahnya dengan benar. Dari 923 pengguna tersebut, terdapat 13 pengguna yang berasal dari wilayah yang sama atau memiliki kode pos yang sama.

3.2 Alat Penelitian

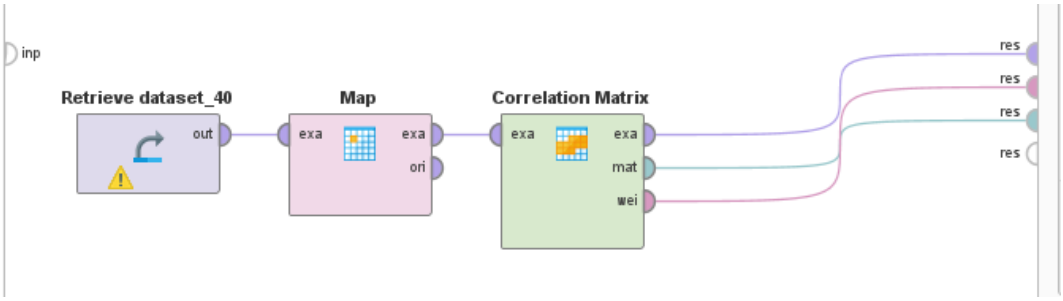
Penelitian ini menggunakan beberapa alat yang terdiri atas *hardware* dan *software* yang disajikan pada tabel 3.4.

Tabel 3.4 Alat penelitian

No	Nama Alat	Spesifikasi	Fungsi
1	Laptop Merk Hawlett Packard	Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz 2.00 GHz	Membuat serta menguji desain penelitian
2	Operating System Windows 11	Windows 11 Pro Version 21H2	Menjalankan aplikasi penelitian
3	Microsoft Excel	2019	Membuat grafik perbandingan terhadap data
4	Rapidminer	9.9	Melakukan klastering (K-Means dan DBSCAN)
5	Matlab	R2022a	Melakukan perangkingan dengan WP-Rank dan evaluasi NDCG

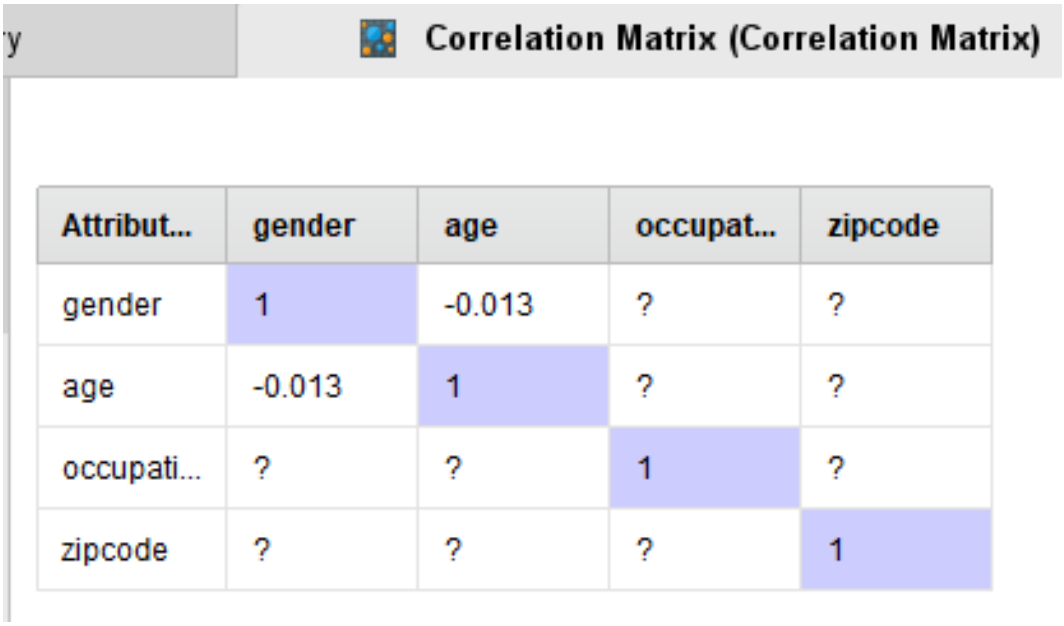
3.3 Pre-*Processing Dataset* Demografi

Collaborative Filtering berbasis pengguna (*User based*) bekerja berdasarkan asumsi bahwa setiap pengguna merupakan bagian dari kelompok yang memiliki kesamaan dengan pengguna lainnya sehingga pengguna yang memiliki kesamaan hubungan (atribut) akan tertarik terhadap item yang sama [28]. Data demografi pengguna pada penelitian ini terdiri atas umur (*age*), jenis kelamin (*gender*), pekerjaan (*occupation*) dan kode pos tempat tinggalnya (*zipcode*). Data demografi ini selanjutnya diperiksa kesamaan hubungan (atributnya). Agar kesamaan hubungannya dapat dicari maka ke 4 atribut tersebut (*age, gender, occupation* dan *zipcode*) diperiksa keterkaitannya / korelasinya (*correlation*). Salah satu *tool* yang dapat digunakan untuk memeriksa keterkaitan ini adalah Rapidminer. Desain proses keterkaitannya ditunjukkan pada gambar 3.5.



Gambar 3.5 Desain keterkaitan demografi 943 pengguna pada *Dataset Movielens* 100k

Maka didapatkan korelasi (*correlation*) antara *age* (usia) dan *gender* (jenis kelamin) sebesar -0.013. *Occupation* dan *zipcode* tidak memiliki korelasi terhadap *age* dan *gender*. Hasil korelasi ini ditunjukkan pada gambar 3.6.



Attribut...	gender	age	occupat...	zipcode
gender	1	-0.013	?	?
age	-0.013	1	?	?
occupati...	?	?	1	?
zipcode	?	?	?	1

Gambar 3.6 Korelasi / keterkaitan *Dataset* Demografi pada *Dataset Movielens* 100k

Karena hanya umur dan jenis kelamin yang berkaitan, maka 2 atribut ini saja yang digunakan dalam pengolahan *dataset* selanjutnya. Atribut pekerjaan dan kode pos selanjutnya tidak digunakan dalam pengolahan *dataset* selanjutnya. Selain itu, umur pengguna yang termuda dan tertua masing - masing hanya 1 orang yaitu umur 7, 10, 11 dan 73 tahun. Umur ini tidak dimasukkan ke

pengolahan *dataset* selanjutnya. Jumlah data yang sebelumnya 943 data berkurang 4 data sehingga menjadi 939 data. Kemudian dibuat *dataset* baru yang terdiri atas *UserId*, umurnya dan jenis kelaminnya. *Dataset* ini diberikan nama / istilah ***Dataset* Demografi Fiks**.

3.4 *Pre-Processing Dataset Rating*

Dataset rating terdiri atas 100.000 penilaian (*rating*) bernilai 1 hingga 5 dari 943 pengguna terhadap 1682 buah film. Jika seluruh pengguna (943 *user*) memberikan *rating* terhadap seluruh film (1682 film), maka seharusnya akan ada 1.586.126 rating. Jumlah ini didapat dari perkalian jumlah *user* dan jumlah film yaitu 943 x 1682 = 1.586.126 *rating*. Akan tetapi, hanya ada 100.000 rating yang berhasil dikumpulkan. Kekurangan *ratingnya* sebesar 1.586.126 – 100.000 = 1.486.126 rating. Jika dipersentase, maka jumlah *ratingnya* yang terisi hanya sebesar 100.000 / 1.586.126 x 100% = 6,0346%. Jika dipersentase, maka jumlah *rating* yang kosong sebesar 1.486.126 / 1.586.126 x 100% = 93,695%. Data yang kosong ini disebut *sparsity*. *Dataset rating* terdiri atas atribut *user id, movie id* dan *rating*. Contoh *dataset*nya ditunjukkan pada tabel 3.5.

Tabel 3.5 Contoh *Dataset Rating* 100k *Movielens* (data bagian atas)

User Id	*Movie Id*	*Rating*
196	242	3
186	302	3
22	377	1
244	51	2
166	346	1

Selanjutnya dilakukan *transpose* terhadap *dataset* tersebut atau diubah menjadi bentuk kolom *User Id*, *movie*1, *movie*2 hingga *movie*1682 pada sebuah tabel baru. *User Id* berisi data nomor pengguna dari 1 hingga 943. *Movie*1 hingga *movie*1682 berisi nilai *rating* yang telah dipilih *user* / pengguna terhadap film tersebut. Setelah dilakukan *transpose*, ternyata banyak data yang kosong

(*sparsity*). Data ini tidak bisa diproses jika kosong. Maka *rating* yang kosong diganti dengan angka nol agar dapat diproses dalam pengolahan data selanjutnya. Contoh *dataset* yang telah di*transpose* ditunjukkan pada tabel 3.6.

Tabel 3.6 *Pre-processing Dataset Rating*

User Id	*movie1*	*movie2*	*movie3*	*...*	*...*	*movie1681*	*movie1682*
1	5	3	4	…	…	0	0
2	4	0	0	…	…	0	0
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
942	0	0	0	…	…	0	0
943	0	5	0	…	…	0	0

Selanjutnya pada *dataset* ini, dihilangkan data yang berisi rating yang berasal dari *user* berumur 7, 10, 11 dan 73 tahun agar sama dengan jumlah user pada ***Dataset* Demografi Fiks** yaitu 939 user. Kemudian, jumlah *rating*nya dihitung kembali dan didapatkan jumlah *rating* dari 939 user sebesar 99.843 *rating*. Berarti terdapat selisih sebesar 157 *rating*. Jika 939 user memberikan penilaian terhadap seluruh film, maka akan didapatkan 1.579.398 *rating*. Ini merupakan perkalian 939 user x 1682 film = 1.579.398 *rating*. Berarti jumlah *rating* yang kosong (*sparsity*) sebesar 1.579.398 - 99.843 = 1.479.555 *rating*.

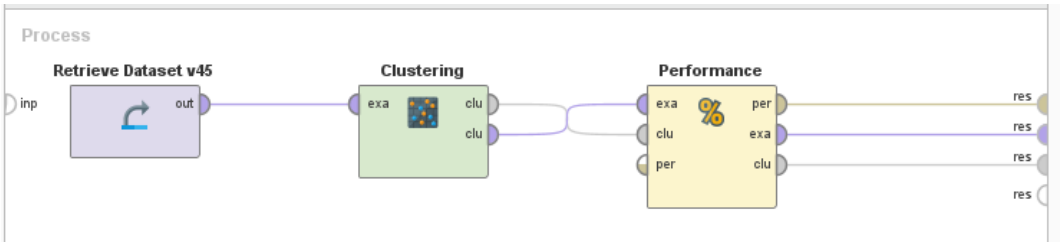
Jika dipersentase dari 939 *user* tersebut, maka persentase *rating* yang terisi hanya sebesar 99.843 / 1.579.398 x 100% = 6,32% dan persentase *rating* yang kosong (*sparsity*) sebesar 1.479.555 / 1.579.398 x 100% = 93,68%.

Selanjutnya *dataset* ini diberikan nama / istilah ***Dataset Rating* Fiks**. *Dataset* ini akan digunakan pada pengolahan data selanjutnya.

3.5 Klasterisasi *Dataset* Demografi Fiks dengan Algoritma K-Means

Pada proses ini, terdapat beberapa langkah yaitu :

1. Lakukan proses klasterisasi dengan algoritma K-Means terhadap ***Dataset* Demografi Fiks** dimulai dari k = 2. Proses klasterisasi ini dapat dilakukan menggunakan *software Rapidminer*. Desain prosesnya ditunjukkan pada gambar 3.7.



Gambar 3.7 Desain proses klasterisasi K-Means Pada ***Dataset* Demografi Fiks**

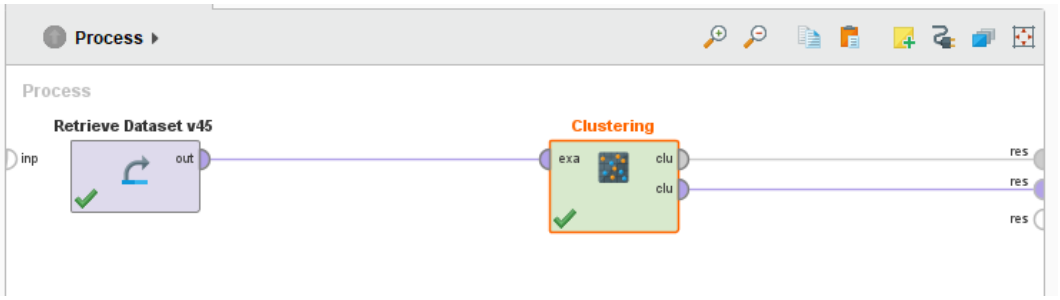
Operator *Retrieve* adalah operator yang dapat mengambil *dataset* yang dibutuhkan. Operator *clustering* digunakan untuk menghitung *Davies Bouldin Index (DBI)* dari masing – masing klaster yang ditetapkan. Parameter yang diinput pada operator ini adalah nilai k yang bisa diubah – ubah. Operator *performance* digunakan untuk mengetahui jumlah anggota tiap klaster, anggota tiap klaster, nilai *Davies Bouldin Index (DBI)* masing – masing klaster, visualisasi klaster yang ada dan lainnya.

2. Mencari *Davies Bouldin Index* (DBI) pada k = 2
3. Lakukan proses nomor 1 dan ubah nilai k nya dari k = 3 hingga k = 30 serta simpan *Davies Bouldin Index*nya.
4. Tentukan nilai DBI terkecil dari klaster k = 2 hingga k = 30.
5. Pada data hasil klaster dengan DBI terkecil, anggota masing – masing klaster digabungkan dengan ***Dataset Rating* Fiks**. *Dataset* penggabungan ini diberi istilah : ***Dataset* K-Means Klaster**.
6. *Dataset* ini selanjutnya akan diolah dengan metode WP-Rank dan NDCG.

3.6 Klasterisasi *Dataset* Demografi Fiks dengan Algoritma DBSCAN

Pada proses ini, terdapat beberapa langkah yaitu :

1. Lakukan proses klasterisasi dengan algoritma DBSCAN terhadap ***Dataset* Demografi Fiks**. Proses klasterisasi ini dapat dilakukan menggunakan *software* Rapidminer. Desain prosesnya ditunjukkan pada gambar 3.7.



Gambar 3.8 Desain proses klasterisasi DBSCAN Pada ***Dataset* Demografi Fiks**

Operator *clustering* DBSCAN memiliki parameter yang bisa diubah yang merupakan ciri khas dari DBSCAN yaitu nilai *epsilon*. Penelitian melakukan klasterisasi dimulai dari *epsilon* 0,1 hingga 4.

2. Mencari jumlah klaster tiap proses di atas.
3. Menentukan klaster terbaik.
4. Mencari keanggotaan pada klaster terbaik.
5. Pada data hasil klaster terbaik, anggota masing – masing klaster digabungkan dengan ***Dataset Rating* Fiks**. *Dataset* penggabungan ini diberi istilah : ***Dataset* DBSCAN Klaster**.
6. *Dataset* ini selanjutnya akan diolah dengan metode WP-Rank dan NDCG.

3.7 *Weight Point Rank* (WP-Rank) *dan Normalized Discounted Cumulative Gain* (NDCG) *Dataset* Hasil Klasterisasi

Pada proses ini terdapat beberapa langkah yaitu :

1. Perangkingan terhadap *dataset* hasil klasterisasi (hasil pengolahan metode 3.5 dan 3.6 langkah ke 5) yaitu ***Dataset* K-Means Klaster** dan ***Dataset* DBSCAN Klaster** dengan metode *Weight Point Rank* (WP-Rank).
2. Pencarian Nilai *Normalized Discounted Cumulative Gain* (NDCG) terhadap ***Dataset* K-Means Klaster** dan ***Dataset* DBSCAN Klaster.**

3. Pencarian waktu proses pada ***Dataset* K-Means Klaster** dan ***Dataset* DBSCAN Klaster.**
4. Langkah 1 – 3 dilakukan masing – masing 10 kali perulangan.
5. Lakukan perbandingan waktu proses dan nilai NDCG 10 kali perulangan.

3.8 *Weight Point Rank* (WP-Rank) dan *Normalized Discounted Cumulative Gain* (NDCG) *Dataset* Sebelum Klasterisasi

Pada proses ini terdapat beberapa langkah yaitu :

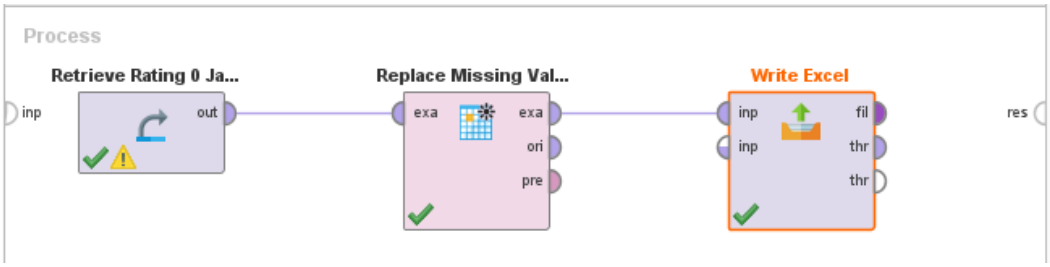
1. Perangkingan terhadap *Dataset* sebelum klasterisasi (***Dataset Rating* Fiks**) dengan metode *Weight Point Rank* (WP-Rank).
2. Pencarian Nilai *Normalized Discounted Cumulative Gain* (NDCG) terhadap ***Dataset Rating* Fiks.**
3. Pencarian waktu proses terhadap ***Dataset Rating* Fiks**.
4. Langkah 1 – 3 dilakukan masing – masing 10 kali perulangan.
5. Lakukan perbandingan waktu proses dan nilai NDCG 10 kali perulangan
6. Lakukan perbandingan dengan poin 3.7.5.

3.9 Mengatasi *Sparsity* Pada *Dataset* Hasil Klasterisasi K-Means

Pada metode 3.5 Langkah ke 5, *dataset* hasil klasterisasi K-Means digabungkan dengan ***Dataset Rating* Fiks** dan diberikan istilah ***Dataset* K-Means Klaster**. Selanjutnya, *dataset* ini dihilangkan *sparsity*nya dengan operator *replace missing value* pada *RapidMiner*.

Pada proses ini terdapat beberapa langkah yaitu :

1. *Dataset* tersebut diolah dengan menggunakan operator *replace missing value*.
2. Pada bagian pengaturan *operator replace missing value*, nilai kosong digantikan dengan nilai rata – rata (*average*) pada kolom tersebut.
3. Hasilnya diinput ke sebuah file Microsoft excel kosong. *Dataset* ini diberi nama ***Dataset* K-Means Klaster Imputasi (KKI).** Desain Langkah metode 3.9 ini ditunjukkan pada gambar 3.9.



Gambar 3.9 Desain RapidMiner untuk mengatasi *sparsity* pada *Dataset* Hasil Klasterisasi K-Means

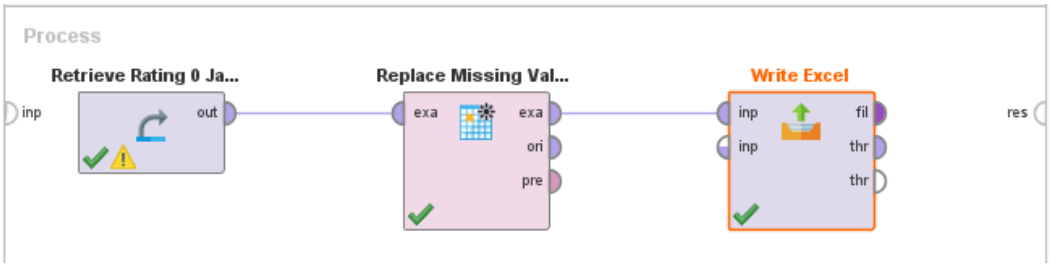
4. Selanjutnya hasilnya dibandingkan dengan perhitungan nilai rata – rata setiap *movie id* menggunakan file excel. Hal ini bertujuan untuk mengetahui perbedaan antara hasil perhitungan nilai rata – rata dengan software Micorsoft Excel dengan hasil olah data menggunakan RapidMiner.

3.10 Mengatasi *Sparsity* Pada *Dataset* Hasil Klasterisasi DBSCAN

Pada metode 3.6 Langkah ke 5, *dataset* hasil klasterisasi DBSCAN digabungkan dengan ***Dataset Rating* Fiks** dan diberikan istilah ***Dataset* DBSCAN Klaster**. Selanjutnya, *dataset* ini dihilangkan *sparsity*nya dengan operator *replace missing value* pada *RapidMiner*.

Pada proses ini terdapat beberapa langkah yaitu :

1. *Dataset* tersebut diolah dengan menggunakan operator *replace missing value.*
2. Pada bagian pengaturan operator *replace missing value*, nilai kosong digantikan dengan nilai rata – rata (average) pada kolom tersebut.
3. Hasilnya diinput ke sebuah file excel kosong. *Dataset* ini diberi nama ***Dataset* DBSCAN Klaster Imputasi (DKI).** Desain Langkah metode 3.10 ini ditunjukkan pada gambar 3.10.



Gambar 3.10 Desain RapidMiner untuk mengatasi *sparsity* pada *Dataset* Hasil Klasterisasi DBSCAN

4. Selanjutnya hasilnya dibandingkan dengan perhitungan nilai rata – rata setiap *movie id* menggunakan file excel. Hal ini bertujuan untuk mengetahui perbedaan antara hasil perhitungan nilai rata – rata dengan software Micorsoft Excel dengan hasil olah data menggunakan RapidMiner.

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

Pada proses ini terdapat beberapa langkah yaitu :

1. Perangkingan terhadap ***Dataset*** **KKI** dengan metode *Weight Point Rank* (WP-Rank).
2. Pencarian Nilai *Normalized Discounted Cumulative Gain* (NDCG) pada ***Dataset*** **KKI**.
3. Pencarian waktu proses pada ***Dataset*** **KKI**.
4. Langkah 1 – 3 dilakukan masing – masing 10 kali perulangan.
5. Lakukan perbandingan waktu proses dan nilai NDCG 10 kali perulangan.

3.12 *Weight Point Rank (WP-Rank)* dan *Normalized Discounted Cumulative Gain* (NDCG) Imputasi Hasil Klaster *Dataset*

Pada proses ini terdapat beberapa langkah yaitu :

1. Perangkingan terhadap ***Dataset*** **DKI** dengan metode *Weight Point Rank* (WP-Rank).

2. Pencarian Nilai *Normalized Discounted Cumulative Gain* (NDCG) pada ***Dataset* DKI**.
3. Pencarian waktu proses pada ***Dataset* DKI**.
4. Langkah 1 – 3 dilakukan masing – masing 10 kali perulangan.
5. Lakukan perbandingan waktu proses dan nilai NDCG 10 kali perulangan.