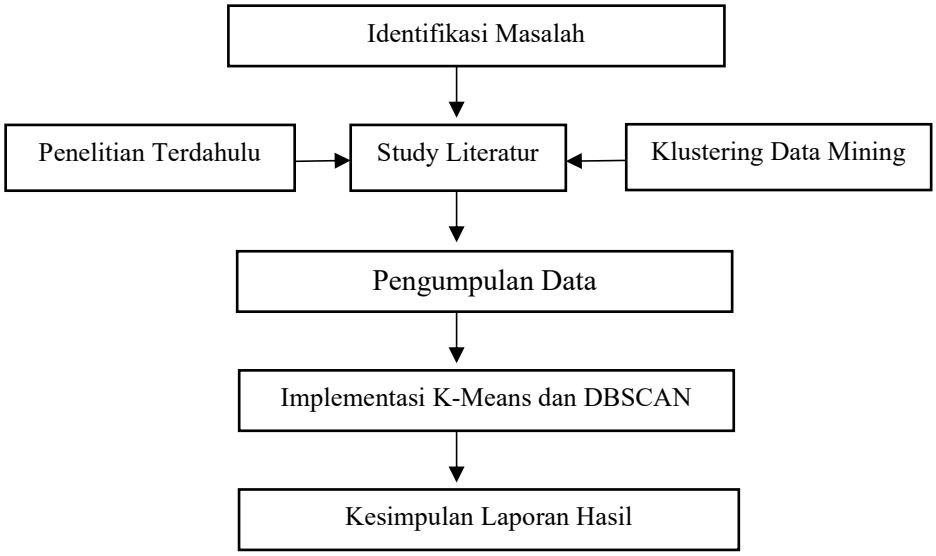
BAB III

METODELOGI PENELITIAN

3.1 Metedologi Penelitian

Metodelogi penelitian yang digunakan penulis pada penelitian dapat dilihat pada alur penelitian. Alur penelitian ini digambar sesuai tahapan yang dilakuka saat penelitian ini berlangsung dari awal penelitian hingga selesainya penelitian. Berikut merupakan beberapa tahapan pada gambar 3.1

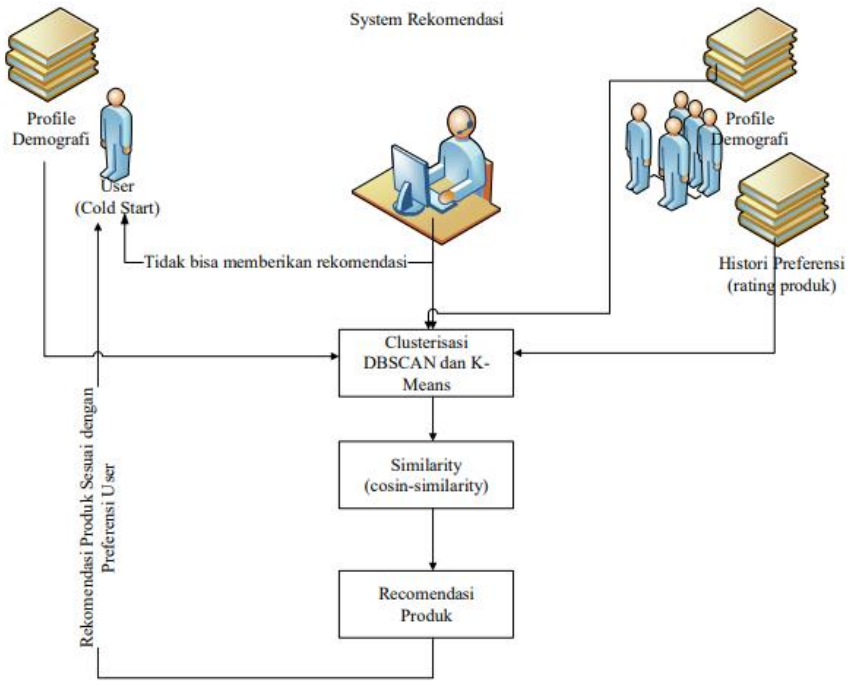


Gambar 3.1 Alur Penelitian

3.2 Kerangka Penelitian

Data yang digunakan merupakan data yang dimiliki oleh grouplens.org yang dikumpulkan oleh Proyek Penelitian GroupLens di Universitas Minnesota. Kumpulan data ini terdiri dari 100.000 (peringkat 1-5) dari 943 pengguna di 1682 film, dimana setiap pengguna telah menilai setidaknya 20 film. Data yang digunakan telah dibersihkan dari pengguna yang tidak memiliki demograpi yang

lengkap dan pengguna yang miliki peringkat kurang dari 20 film. Baik Universitas Minnesota maupun peneliti mana pun yang terlibat dapat menjamin kebenaran data kesesuaiannya untuk tujuan tertentu, atau keabsahan hasil berdasarkan penggunaan kumpulan data. penelitian dimulai dengan melakukan pengunduhan data pada link https://grouplens.org/datasets/movielens/100k/ setelah data didapatkan, tahap selanjutnya yakni menganalisa data dengan melihat, memperbaiki ataupun mentransformasi data atau dapat di sebut *preprocessing data*. setelah dilakukan *preprocessing data* selanjutnya dapat memasukan data tersebut kedalam algoritma K-Means dan DBSCAN untuk menentukan rekomendasi film kepada penonton. Adapun kerangka penelitian yang dilakukan dapat dilihat pada gambar 3.2



Gambar 3.2 Kerangka Penelitian

3.3 Analisis Pengolahan Data

Data yang digunakan pada penelitiam ini dapat diunduh pada link https://grouplens.org/datasets/movielens/100k/. Setelah di unduh, kita dapat melakukan preprocessing data seperti yang dapat kita lihat pada tabel 3.1 dan tabel 3.2

Tabel 3.1 Data sebelum *preprocessing*

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id	age	gender	occupation	movie1	movie2	movie3	movie4	movie5	movie6	movie7	movie8	movie9	movie10
2	1	24	M	technician	5	3	4	3	3	5	4	1	5	3
3	2	53	F	other	4	0	0	0	0	0	0	0	0	2
4	3	23	M	writer	0	0	0	0	0	0	0	0	0	0
5	4	24	M	technician	0	0	0	0	0	0	0	0	0	0
6	5	33	F	other	4	3	0	0	0	0	0	0	0	0
7	6	42	M	executive	4	0	0	0	0	0	2	4	4	0
8	7	57	M	administrato	0	0	0	5	0	0	5	5	5	4
9	8	36	M	administrato	0	0	0	0	0	0	3	0	0	0
10	9	29	M	student	0	0	0	0	0	5	4	0	0	0
11	10	53	M	lawyer	4	0	0	4	0	0	4	0	4	0
12	11	39	F	other	2	5	5	5	5	5	3	4	5	1
13	12	28	F	other	1	4	4	3	4	3	2	4	1	4
14	13	47	M	educator	3	5	4	2	1	2	2	3	4	0
15	14	45	M	scientist	2	5	4	5	5	4	4	5	3	0
16	15	49	F	educator	4	4	3	3	5	4	5	4	5	0
17	16	21	M	entertainme	4	3	2	5	4	4	3	4	3	0

Tabel 3.2 Data setelah *preprocessing*

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id	age	gender	occupation	movie1	movie2	movie3	movie4	movie5	movie6	movie7	movie8	movie9	movie10
2	1	24	2	20	5	3	4	3	3	5	4	1	5	3
3	2	53	1	14	4	0	0	0	0	0	0	0	0	2
4	3	23	2	21	0	0	0	0	0	0	0	0	0	0
5	4	24	2	20	0	0	0	0	0	0	0	0	0	0
6	5	33	1	14	4	3	0	0	0	0	0	0	0	0
7	6	42	2	7	4	0	0	0	0	0	2	4	4	0
8	7	57	2	1	0	0	0	5	0	0	5	5	5	4
9	8	36	2	1	0	0	0	0	0	0	3	0	0	0
10	9	29	2	19	0	0	0	0	0	5	4	0	0	0
11	10	53	2	10	4	0	0	4	0	0	4	0	4	0
12	11	39	1	14	2	5	5	5	5	5	3	4	5	1
13	12	28	1	14	1	4	4	3	4	3	2	4	1	4
14	13	47	2	4	3	5	4	2	1	2	2	3	4	0
15	14	45	2	18	2	5	4	5	5	4	4	5	3	0
16	15	49	1	4	4	4	3	3	5	4	5	4	5	0
17	16	21	2	6	4	3	2	5	4	4	3	4	3	0

Pada tabel 3.3, tabel 3.4 dan tabel 3.5 merupakan penjabaran mengenai transformasi data yang berawal merupakan data berbentuk teks menjadi data dengan menggunakan angka.

Tabel 3.3 Transformasi data pekerjaan penonton

Pekerjaan	Notasi angka
Administration	1
Artist	2
Doctor	3
Educator	4
Engineer	5
Entertainment	6
Executive	7
Healthcare	8
Homemaker	9
Lawyer	10
Librarian	11
Marketing	12
None	13
Other	14
Programmer	15
Retired	16
Salesman	17
Scientist	18
Student	19
Technician	20
Writer	21

Tabel 3.4 Tranformasi data jenis kelamin penonton

Jenis kelamin	Notasi angka
Female	1
Male	2

Tabel 3.5 Tranformasi data judul film

Judul film	Notasi angka
Toy story	Movie 1
Golden eye	Movie 2
Four rooms	Movie 3
Get shorty	Movie 4
Copy cat	Movie 5
Dst..	

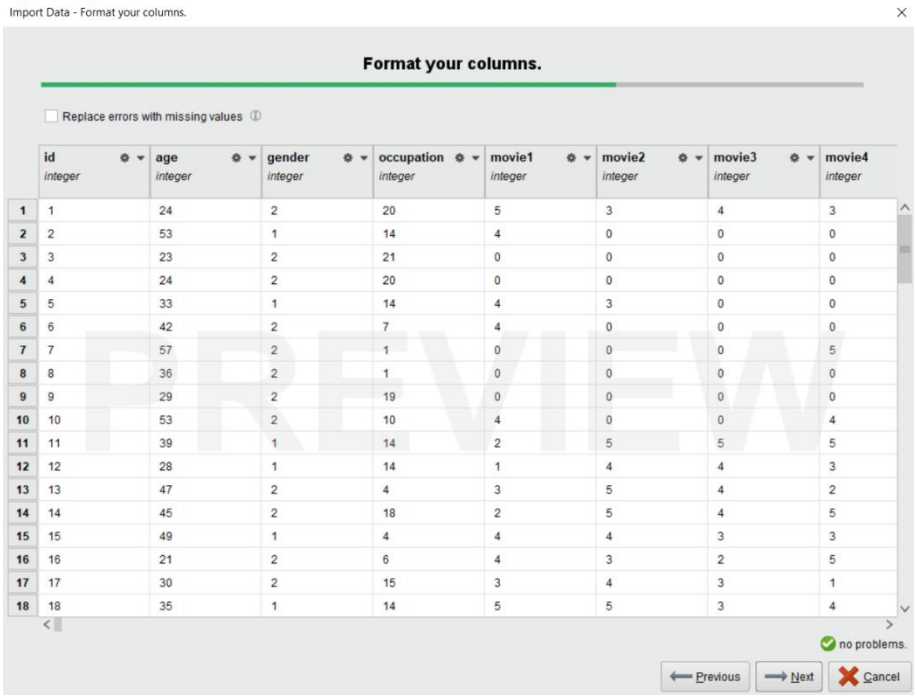
Setelah semua data telah dilakukan proses pre-processing data, maka kita dapat langsung memasukannya kedalam rapid miner untuk dapat menggali informasi penyebaran data dengan menggunakan algoritma ke-means dan juga DBSCAN.

3.4 Rapid Miner

Selanjutnya data yang telah dilakukan transformasi data, data tersebut dapat di proses menggunakan *platform* rapid miner. Pada penelitian ini, penulis menggunakan rapid miner 9.10.

3.4.1 Pengaplikasian K-means pada rapid minner

Berikut merupakan proses *clustering* menggunakan k-means yang dapat dilihat pada gambar 3.3 yang merupakan gambar import data yang ada pada rapid miner



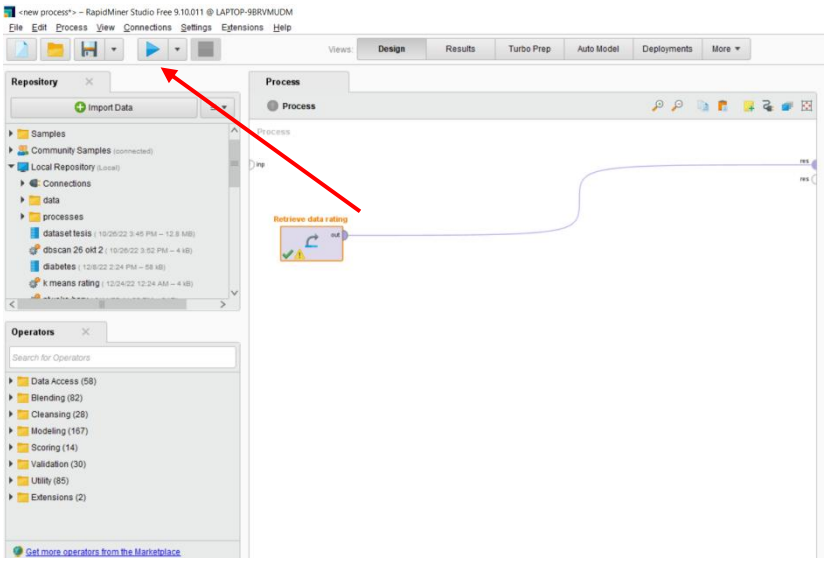
	id integer	age integer	gender integer	occupation integer	movie1 integer	movie2 integer	movie3 integer	movie4 integer
1	1	24	2	20	5	3	4	3
2	2	53	1	14	4	0	0	0
3	3	23	2	21	0	0	0	0
4	4	24	2	20	0	0	0	0
5	5	33	1	14	4	3	0	0
6	6	42	2	7	4	0	0	0
7	7	57	2	1	0	0	0	5
8	8	36	2	1	0	0	0	0
9	9	29	2	19	0	0	0	0
10	10	53	2	10	4	0	0	4
11	11	39	1	14	2	5	5	5
12	12	28	1	14	1	4	4	3
13	13	47	2	4	3	5	4	2
14	14	45	2	18	2	5	4	5
15	15	49	1	4	4	4	3	3
16	16	21	2	6	4	3	2	5
17	17	30	2	15	3	4	3	1
18	18	35	1	14	5	5	3	4

Gambar 3.3 Import data ke dalam rapid miner

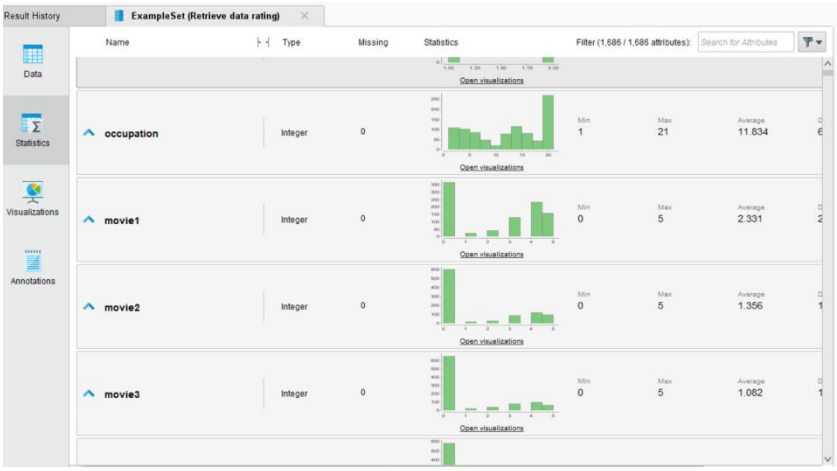
Setelah dilakukan pengimportan data, kita dapat mulai untuk proses data mining dalam rapid miner dengan cara import data kedalam halaman proses yang dapat dilihat pada gambar 3.4

Gambar 3.4 Import data ke halaman proses

Pada gambar 3.5 kita dapat melihat data yang kita import dengan tombol play yang ada pada pojok kanan atas untuk melihat statistic data yang telah kita import.

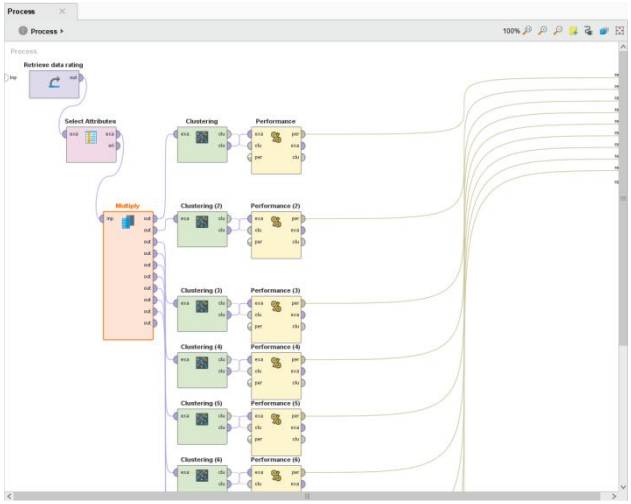


Gambar 3.5 Tombol play untuk melihat data yang telah di import



Gambar 3.6 Statistik data

Dengan melihat *statistic* pada gambar 3.6, kita dapat melihat data mana saja yang memiliki kekosongan data (*missing data*) agar dapat dilakukan proses *replace missing value* atau langsung dapat masuk kedalam algoritma k-means. Setelah melihat data satu persatu, penulis tidak menemukan adanya kekosongan data. Sehingga, kita dapat secara langsung memasukan algoritma k-means ke dalam data rating tersebut. Dalam menentukan k optimal pada penelitian ini, penulis menggunakan Davies Bouldin index untuk menentukan nilai K mana yang paling optimal yang merupakan sebuah metode untuk memvalidasi cluster eveluasi kuantitatif dari hasil *clustering*. Adapun pemilihan nilai yang digunakanan adalah nilai yang paling kecil yang diperoleh oleh test DBI. Semakin kecil nilai DBI yang digunakan, maka semakin baik juga performa cluster yang didapatkan yang dapat dilihat pada gambar 3.7



Gambar 3.7 Proses percobaan mencari nilai K optimal

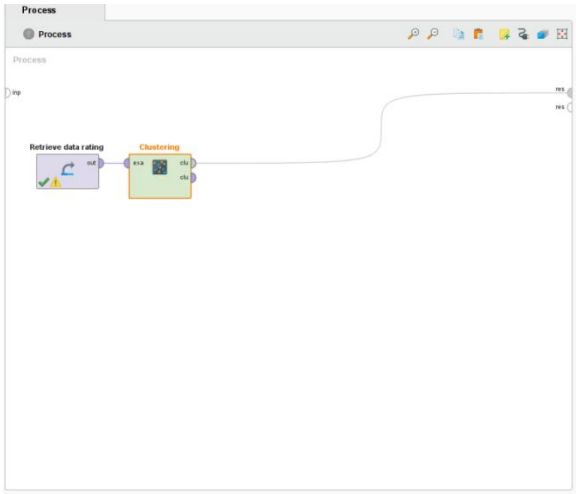
Setelah dilakukan percobaan nilai K, maka di dapati nilai uji DBI yang dapat dilihat pada tabel 3.7

Tabel 3.7 Nilai uji DBI

Jumlah Cluster	Nilai DBI K-Means
2	-2.980
3	-3.402
4	-3.528
5	-3.670
6	-3.673
7	-3.712
8	-4.071
9	-3.806
10	-3.731
11	-4.206
12	-3.677
13	-3.563
14	-3.366
15	-3.346
16	-3.237
17	-3.795
18	-3.376
19	-3.330
20	-3.190
21	-3.693
22	-3.151
23	-3.172

24	-2.976
25	-3.537
26	-2.999
27	-3.224
28	-3.299
29	-3.145
30	-3.093
31	-3.393
32	-3.046
33	-3.095
34	-3.029
35	-3.304

Dengan melihat nilai DBI yang terus naik kembali, maka proses percobaan berhenti pada nilai 35 dengan K optimal sebesar 2 *clustering*. Setelah mendapatkan nilai K-optimal, maka proses clustering dapat diimplementasikan seperti pada gambar 3.8



Gambar 3.8 Konfigurasi rapid miner menggunakan algoritma K-Means

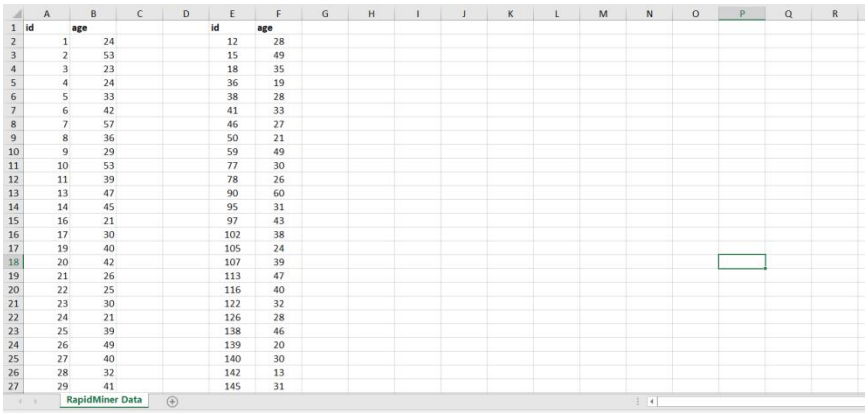
3.4.2 Pengaplikasian DBSCAN Pada Rapid Minner

DBSCAN merupakan algoritma yang di rancang untuk menemukan jumlah *cluster* dan juga *noise* yang ada pada data. Sama halnya dengan pencarian K

optimal pada *clustering* K-means, pemilihan nilai eps dibutuhkan pemilihan yang tepat agar bentuk *clustering* yang di dapatkan berjalan dengan optimal. Pada penelitian kali ini, penulis menggunakan *Euclidean distance* untuk menemukan nilai epsilon. Proses perhitungan nilai eps terbaik meggunakan algoritma K-NN.

Dalam proses penggunaan algoritma K-NN dibutuhkan data latih dan data testing. Pada proses penelitian ini, data dibagi menjadi 70:30. Dimana, 70% dari keseluruhan data merupakan data latih dan 30% merupakan data testing. Data set tersebut dapat dilihat pada tabel 3.8 dibawah ini.

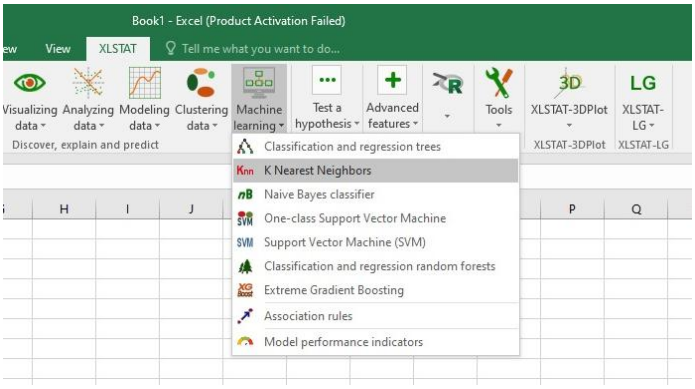
Tabel 3.8 dataset split data algoritma K-NN



id	age		id	age
1	24		12	28
2	53		15	49
3	23		18	35
4	24		36	19
5	33		38	28
6	42		41	33
7	57		46	27
8	36		50	21
9	29		59	49
10	53		77	30
11	39		78	26
13	47		90	60
14	45		95	31
16	21		97	43
17	30		102	38
19	40		105	24
20	42		107	39
21	26		113	47
22	25		116	40
23	30		122	32
24	21		126	28
25	39		138	46
26	49		139	20
27	40		140	30
28	32		142	13
29	41		145	31

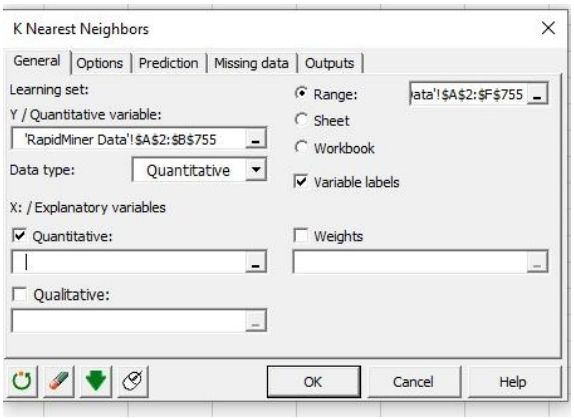
RapidMiner Data

Setelah itu kita dapat pilih pada *toolbar* menu *mechine learning* K-nearest neighbors untuk mengimplementasikan K-NN pada penelitian ini yang ditunjukan pada gambar 3.9



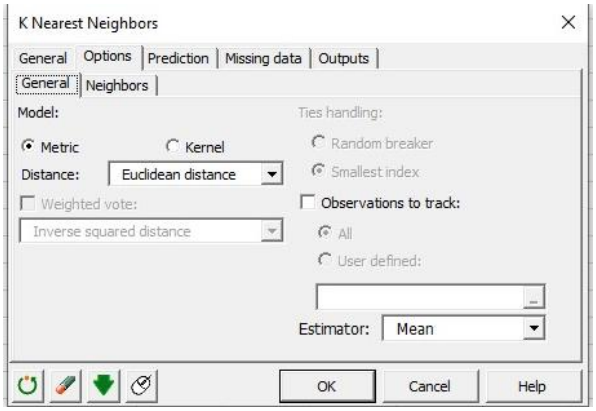
Gambar 3.9 proses pemilihan algoritma K-NN pada XLSTAT

Selanjutnya, kita dapa mengklik algoritma K-NN dan menginput data latih kedalam box data latih yang ditunjukan pada gambar 3.10



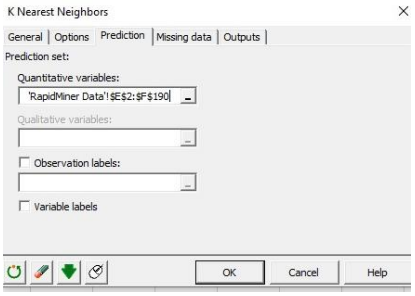
Gambar 3.10 Input data latih pada K-NN

Setelah memasukan data latih kita dapat memasukan memilih secara manual ataupun memodifikasi kedekatan jarak antar tetangga, dimana peneliti menggunakan *Euclidean distance* dalam menghitung jarak antar tetangga



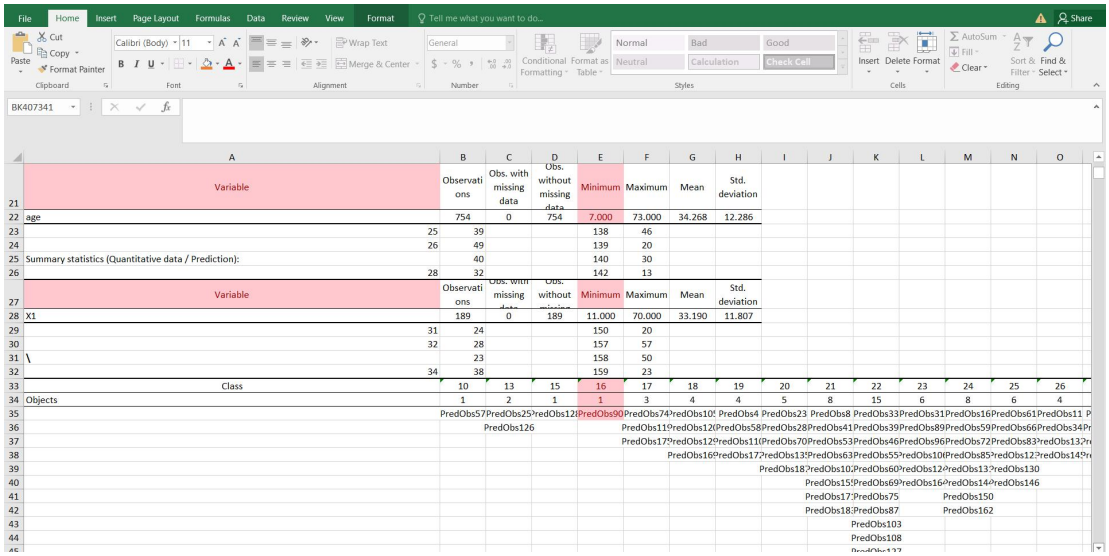
Gambar 3.11 penggunaan *Euclidean distance* pada algoritma K-NN

Dan terkahir, kita dapat memausukan *data testing* kedalam menu prediction yang ditunjukan pada gambar 3.12



Gambar 3.12 penginputan *data testing* kedalam menu *prediction*

Pada gambar 3.13 merupakan plot hasil dari perhitungan tetangga dimana nilai 0.7 merupakan nilai yang dapat menjadi nilai Eps pada algoritma DBSCAN



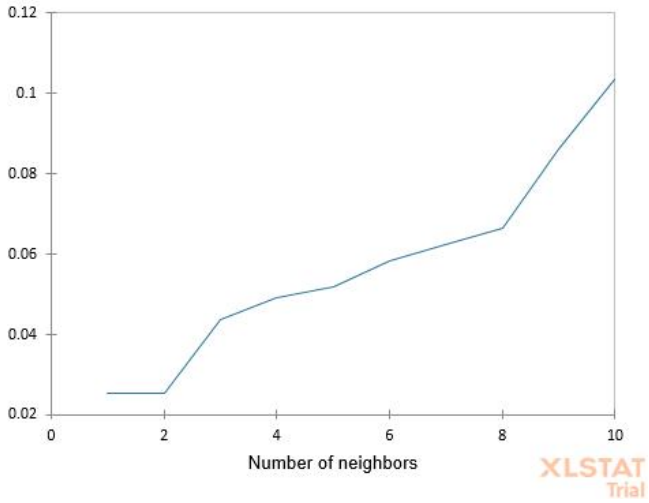
Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
age	754	0	754	7.000	73.000	34.268	12.286

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
X1	189	0	189	11.000	70.000	33.190	11.807

Gambar 3.13 proses perhitungan KNN untuk menentukan nilai epsilon

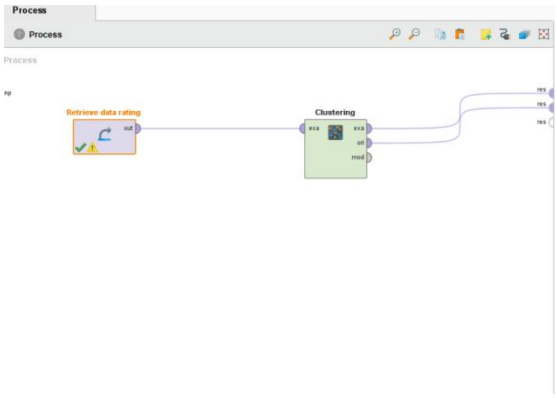
Number of neighbors	Loss estimate using cross-validation
1	**0.025199**
2	0.025199
3	0.043767
4	0.049072
5	0.051724
6	0.058355
7	0.062334
8	0.066313
9	0.086207
10	0.103448

Gambar 3.14 perhitungan cross validation untuk menentukan gambar “knee” pada KNN



Gambar 3.15 Jarak rata-rata tetangga terdekat mengguanakan algoritma KNN

Setelah ditemukan nilai Eps, selanjutnya kita dapat melanjutkan proses *clustering* menggunakan algoritma DBSCAN dengan *tools* Rapid miner seperti pada gambar 3.16

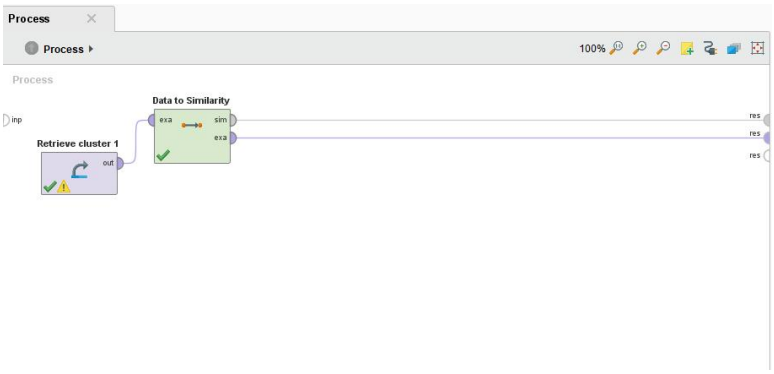


Gambar 3.16 Konfigurasi DBSCAN dengan menggunakan rapidminer

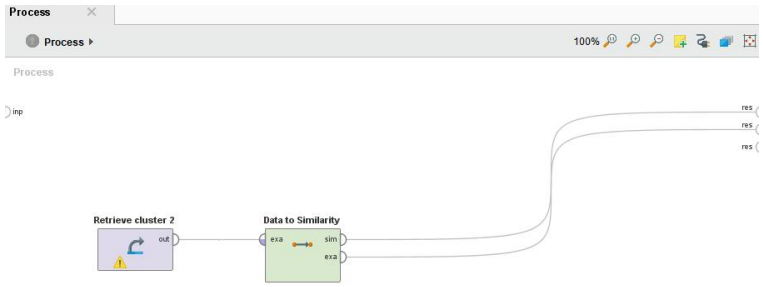
3.5 Cosine Similarity

Setelah dilakukannya penelitian, kedekatan jarak antar data yang di proses hanya pada algoritma K-Means saja. Berikut merupakan proses dari penggunaan *cosine*

similarity K-means menggunakan rapid miner yang dapat dilihat pada gambar 3.17 dan gambar 3.18



Gambar 3.17 Proses Cosine Similairty Cluster 1

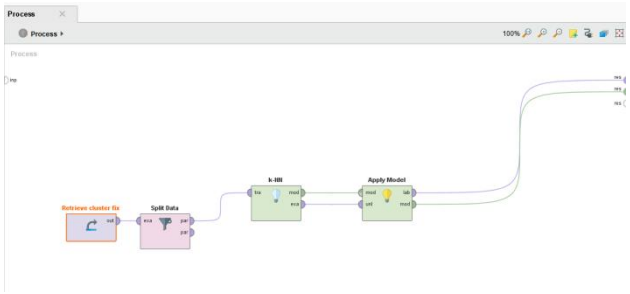


Gambar 3.18 Proses Cosine Similairty Cluster 2

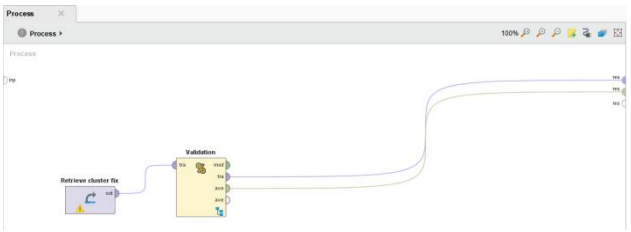
3.6 Evaluasi

Untuk memvalidasi tingkat akurasi dari algoritma yang digunakan oleh penulis, penulis melakukan uji coba akurasi data yang telah diproses menggunakan algoritma KNN yang dapat dilihat pada gambar 3.19. Algoritma KNN bekerja

dengan cara menghitung jarak setiap titik pada *dataset* dengan *data training.* Kelas yang memiliki kedekatan jarak yang paling dekat akan menjadi kelas data set tersebut. Berikut merupakan proses evaluasi hasil data menggunakan algoritma KNN dengan validasi *X-Validation* yang dapat dilihat pada gambar 3.20



Gambar 3.19 proses implementasi algoritma KNN



Gambar 3.20 proses implementasi X-Validation