BAB III
METODE PENELITIAN DAN PERANCANGAN

Pada bab ini menjelaskan langkah-langkah yang akan ditempuh dalam menyusun penelitian. Tahapan penelitian ini dirancang untuk membangun sistem rekomendasi yang dapat memberikan rekomendasi film kepada *user* dengan menggunakan *item-based clustering hybrid method*.

3.1 Studi Literatur

Dalam penelitian ini dibutuhkan studi literatur untuk merealisasikan tujuan dan penyelesaian masalah. Teori-teori berkaitan dengan sistem rekomendasi dan *item-based clustering hybrid method* digunakan sebagai dasar penelitian yang didapat dari buku, jurnal dan *browsing* dari internet. Dari data yang diperoleh akan diolah sehingga dapat digunakan untuk analisis. Setelah dilakukan analisis maka dapat diimplementasikan ke dalam aplikasi.

3.2 Data Penelitian

Untuk mengetahui bagaimana kinerja sistem rekomendasi, dibutuhkan dataset rating yang telah terisi dengan *record* rating *item* dari *user*. Dataset *item* (film) yang digunakan dalam penelitian ini diambil dari *GroupLens* (http://www.grouplens.org) yang diterbitkan pada tahun 2011. *GroupLens* merupakan proyek penelitian yang dibuat oleh Universitas Minnesota yang menyediakan dataset film.

Berikut statistik dataset film dari *GroupLens* yang digunakan pada penelitian:

Tabel 3.1 Statistik Dataset *GroupLens*

Jumlah user	943
Jumlah *item*	1.682
Jumlah rating	100.000
Jumlah *genre*	19
Nilai rating	1 - 5

Dan setiap *user* telah merating sedikitnya 20 film.

3.3 Deskripsi Umum Sistem

Secara umum sistem yang dibangun adalah perangkat lunak yang memberikan rekomendasi film kepada *user*. Dengan adanya sistem rekomendasi ini diharapkan *user* mendapatkan film yang sesuai. Metode yang digunakan dalam sistem rekomendasi film ini adalah *item-based clustering hybrid method*.

Dalam metode *item-based clustering hybrid method* terdapat tiga proses yang dilakukan. Yang pertama adalah pembentukan matriks rating, yaitu matrik *item-rating* dan matriks *group-rating*. Proses kedua adalah menghitung *similarity* dari matriks yang didapatkan dari proses pertama. Tahap ketiga adalah dilakukan perhitungan prediksi rating. Dari perhitungan prediksi rating ini akan didapatkan hasil rekomendasi.

3.4 Perancangan Proses

Tahapan proses dalam sistem rekomendasi film menggunakan metode *item-based hybrid clustering method* dibagi menjadi tiga tahapan yaitu pembentukan matriks rating, perhitungan nilai *similiarity* dan perhitungan prediksi rating.

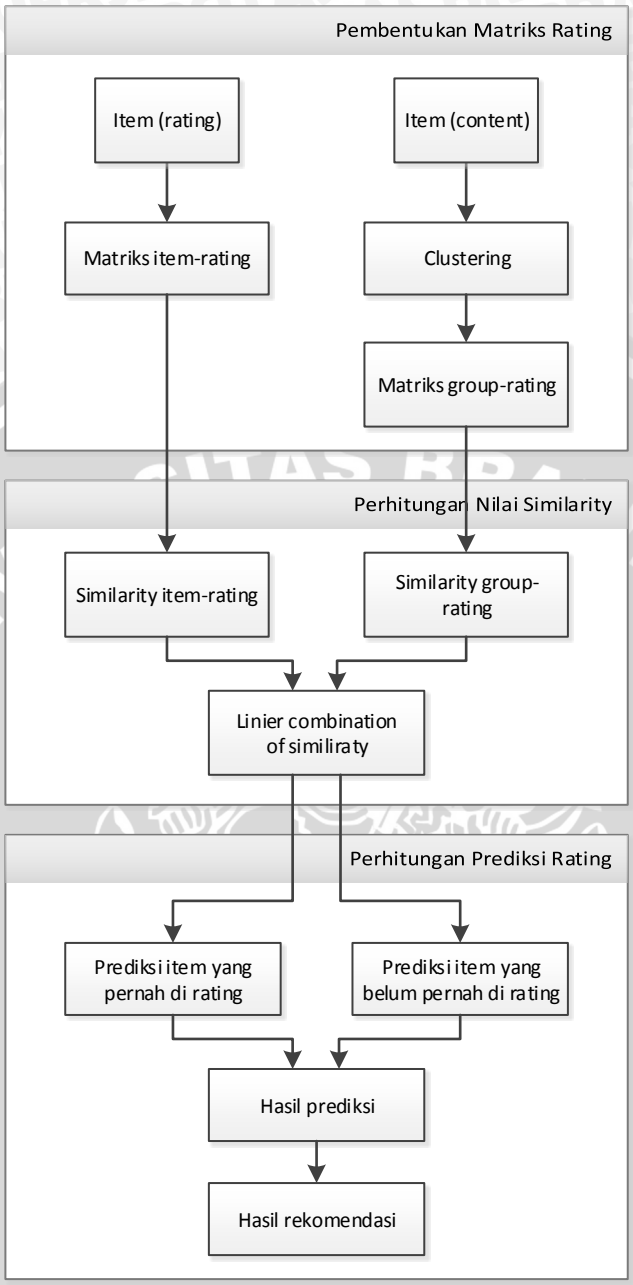
Pada tahap pertama yaitu pembentukan matriks rating, akan dibentuk dua matriks, yaitu matriks *item-rating* dan matriks *group-rating*. Kedua matriks tersebut dibangun dari *item* yang berisi atribut yaitu judul film, rating film dan *user* yang merating film. Pertama, dari atribut *item* tersebut dijadikan model matriks, dimana matriks ini disebut sebagai matriks *item-rating*. Sedangkan untuk membangun matriks *group-rating* diimplementasikan proses *clustering*. Implementasi *clustering* dilakukan dengan *input* berupa sejumlah *k cluster* dan konten *item*. Algoritma *clustering* yang digunakan adalah *K-Means clustering*. Dari hasil proses *K-Means clustering* akan didapatkan beberapa *cluster* dan hasil *cluster* tersebut dijadikan model matriks, dimana matriks ini adalah matriks *group-rating*.

Tahap kedua yaitu perhitungan *similarity* pada kedua matriks, yaitu perhitungan *similarity* matriks *item-rating* dan *group-rating*. Algoritma *pearson correlation-based similarity* digunakan untuk perhitungan *similarity* pada matriks *item-rating* (*similarity item-rating*) dan algoritma *adjusted-cosine similarity* digunakan untuk perhitungan *similarity* pada matriks *group-rating* (*similarity*

group-rating). Kemudian hasil perhitungan *similarity* dari kedua matriks tersebut digabungkan menggunakan algoritma *linier combination of similiraty*.

Tahap ketiga yaitu perhitungan prediksi rating. Algoritma *weighted average of deviation* digunakan untuk perhitungan prediksi rating pada *item* yang sudah pernah *di rating* (*non cold-start problem*), sedangkan untuk *item* yang belum pernah *di rating* sama sekali (*cold-start problem*) digunakan algoritma *weighted sum* untuk proses perhitungan prediksi rating. Dari hasil perhitungan prediksi rating tersebut dijadikan sebagai penentuan rekomendasi *item* kepada *user* (*Top-N* rekomendasi). Hasil rekomendasi berupa sejumlah *item* beserta nilainya dan telah diperingkatkan.

Diagram proses dalam sistem rekomendasi film dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Perancangan Proses

3.4.1 Pembentukan Matriks Rating

Pembentukan matriks rating dibangun dari atribut yang ada pada *item* (film), yaitu *movie title*, rating film, *genre* film dan *user* yang memberikan rating pada film. Pada data film untuk *movie title* mempunyai lebih dari satu *genre*. Proses pembentukan matriks bertujuan untuk membangun dua buah matriks, yaitu matriks *item-rating* dan matriks *group-rating*.

Misalkan diketahui data film yang diperoleh dari *user* seperti ditunjukkan dalam Tabel 3.2 di bawah ini :

Tabel 3.2 Contoh Data *Item* dan *User*

user_id	movie_title	genre	rating
X	A	action, adventure, war	2
X	B	action, comedy, drama	3
X	C	adventure, comedy, mystery	4
X	D	adventure, comedy	2
Y	A	action, adventure, war	1
Y	B	action, comedy, drama	2
Y	D	adventure, comedy	3
Y	E	action, drama, romance	2
Z	A	action, adventure, war	4
Z	B	action, comedy, drama	1
Z	D	adventure, comedy	2

Maka langkah-langkah yang dilakukan untuk membuat model matriks rating adalah sebagai berikut :

3.4.1.1 Matriks *Item-rating*

Dari data user yang ada pada Tabel 3.2, dijadikan model matriks *movie_title x user_id*, dimana matriks ini adalah matriks *item-rating*, seperti ditunjukkan pada Tabel 3.3 berikut ini.

Tabel 3.3 Matriks *Item-rating*

movie_title	user_id		
	X	Y	Z
A	2	1	4
B	3	2	1
C	4	-	-
D	2	3	2
E	-	2	-

3.4.1.2 Matriks *Group-rating*

Sedangkan untuk pembentukan matriks *group-rating* terlebih dahulu dilakukan proses *clustering*. Proses *clustering* bertujuan untuk mendapatkan sejumlah *cluster*. *Cluster* yang dihasilkan digunakan untuk membangun matriks *group-rating*.

Clustering dilakukan terhadap data *genre* dari film, yakni bobot atau nilai dari setiap *genre*. Bobot setiap *genre* didapatkan dengan menghitung rata-rata rating dari *user* terhadap film, dimana film tersebut masuk ke dalam beberapa *genre*.

Dari data pada Tabel 3.2 setiap film masuk ke dalam beberapa *genre* seperti ditunjukkan pada Tabel 3.4 berikut.

Tabel 3.4 Contoh *Genre* Film

movie_title	**genre**						
	action	adventure	comedy	drama	mystery	romance	war
A	1	1					1
B	1		1	1			
C		1	1		1		
D		1	1				
E	1			1		1	

Dari data pada Tabel 4.4 diatas dihitung rata-rata rating dari *user* terhadap film, dimana film tersebut masuk ke dalam beberapa *genre*. Setelah didapatkan rata-rata rating setiap *user* terhadap *genre*, kemudian dihitung rata-rata rating semua *user* pada setiap *genre*, seperti ditunjukkan pada Tabel 3.5 berikut ini.

Tabel 3.5 Rata-rata Rating Pada *Genre*

movie_title	**user_id**			**rata-rata rating**
	X	Y	Z	
Action	1,7	1,7	1,7	1,7
Adventure	2,7	1,3	2	2
Comedy	3	1,7	1	1,6

Drama	1,5	2	0,5	1,3
Mystery	4	0	0	1,3
Romance	0	2	0	0,7
War	2	1	1	1,3

Setelah didapatkan rata-rata rating pada setiap *genre*, didapatkan data berupa nilai setiap *genre* terhadap *movie title* seperti ditunjukkan pada Tabel 3.6 berikut.

Tabel 3.6 Nilai *Genre*

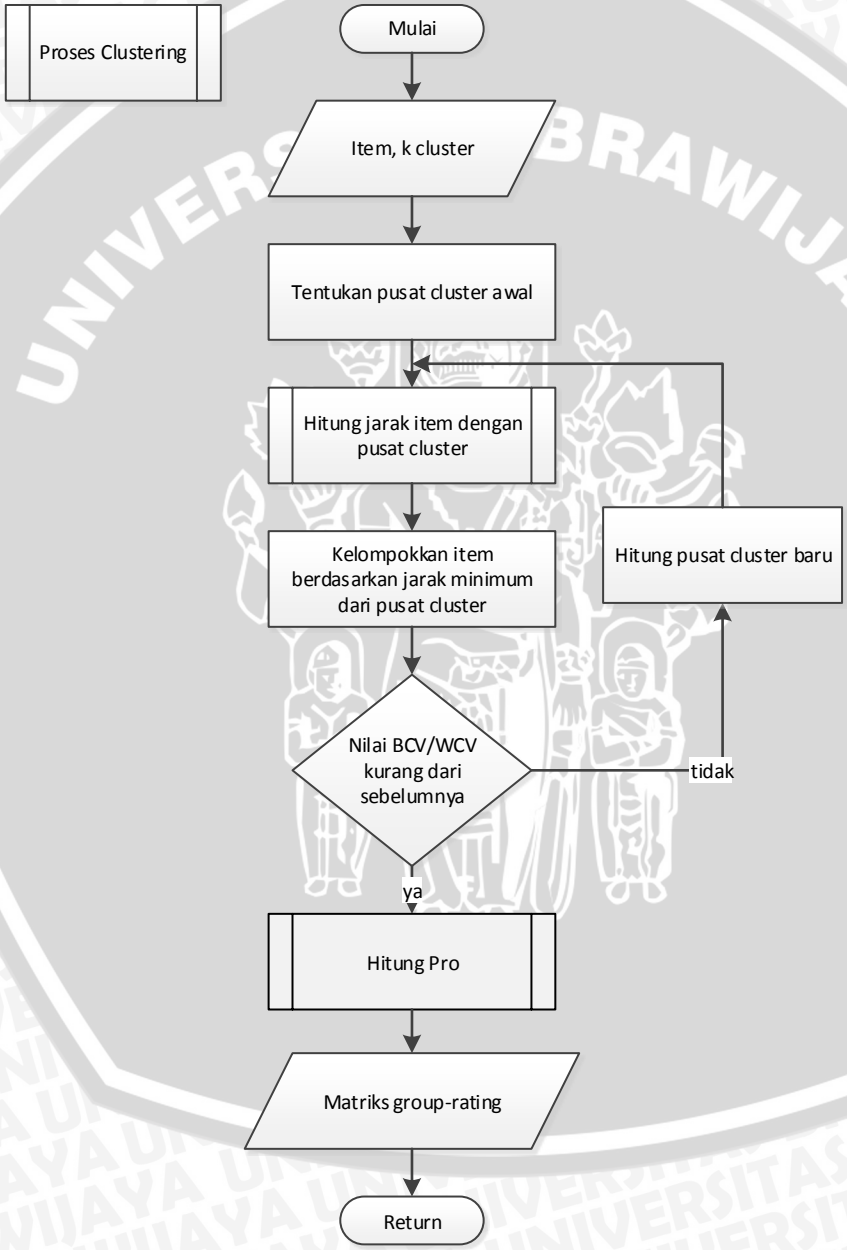
movie_title	**genre**						
	action	adventure	comedy	drama	mystery	romance	war
A	1,7	2	0	0	0	0	1,3
B	1,7	0	1,6	1,3	0	0	0
C	0	2	1,6	0	1,3	0	0
D	0	2	1,6	0	0	0	0
E	1,7	0	0	1,3	0	0,7	0

Dari data pada Tabel 3.6 dilakukan proses *clustering*. Algoritma yang digunakan dalam proses *clustering* adalah *K-Means clustering*. Langkah-langkah dalam proses *clustering* adalah sebagai berikut :

1. Masukkan (*input*) berupa nilai *genre* (*n*) dan sejumlah *k cluster*, *k* disini merupakan banyaknya *cluster* yang ingin dibentuk.
2. Tentukan pusat *cluster* awal, sebagai titik tengah *cluster*.
3. Masukkan tiap *item* ke dalam *cluster* dan hitung jarak *item* dengan pusat *cluster* menggunakan metode *Eucledian Distance*. *Eucledian Distance* dirumuskan sesuai dengan persamaan (2.1).
4. Kelompokkan tiap *item* berdasarkan jarak minimum atau terkecil dari pusat *cluster*.
5. Ulangi atau lakukan proses iterasi dari langkah 3, sampai didapatkan nilai $\frac{BCV}{WCV}$ lebih kecil dari nilai $\frac{BCV}{WCV}$ pada proses iterasi sebelumnya.

6. Setelah didapatkan *cluster* atau kelompok, dihitung keterkaitan atau peluang antar *item* terhadap *cluster* (*Pro*).
7. Keluaran (*output*) berupa *matriks group-rating*.

Adapun *flowchart* dari proses *clustering* seperti ditunjukkan pada Gambar 3.2.

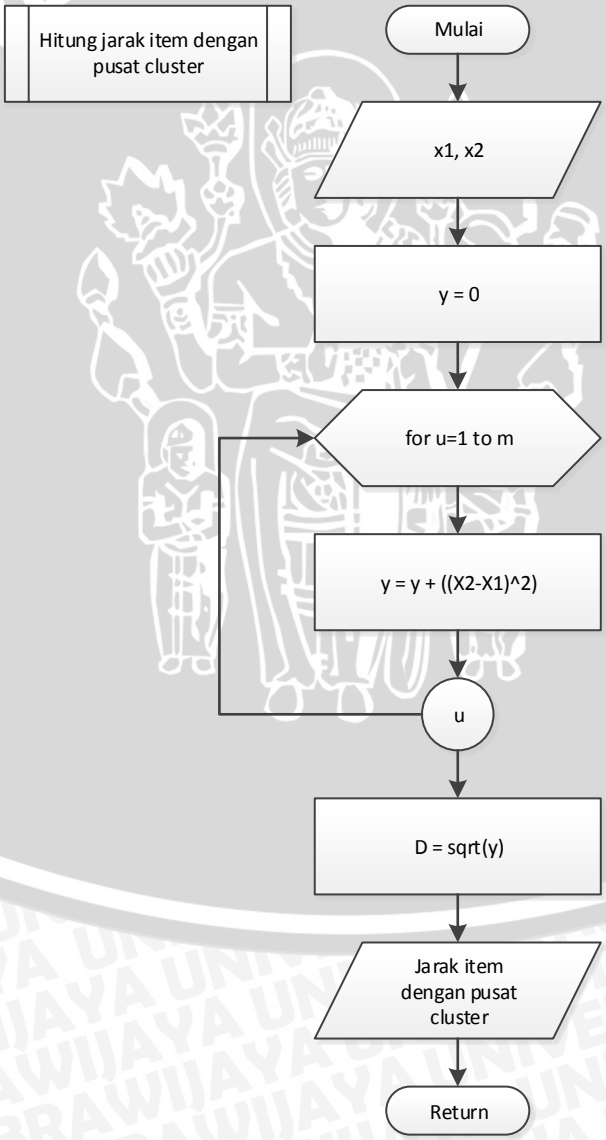


Gambar 3.2 *Flowchart* Proses *Clustering*

Sedangkan langkah-langkah dari subproses hitung jarak *item* dengan pusat *cluster* adalah sebagai berikut :

1. Masukkan (*input*) berupa pusat *cluster* (x_1) dan nilai data (x_2).
2. Lakukan pengurangan (*y)* antara nilai dari x_2 dengan x_1 kemudian hasilnya kuadratkan.
3. Lakukan proses pada langkah 2 sebanyak jumlah data (*m*).
4. Kemudian perhitungan akar (*D*) dari hasil yang didapat pada langkah 3.
5. Keluaran (*output*) berupa nilai jarak item dengan pusat *cluster*.

Adapun *flowchart* dari subproses hitung jarak item dengan pusat *cluster* seperti ditunjukkan pada Gambar 3.3.

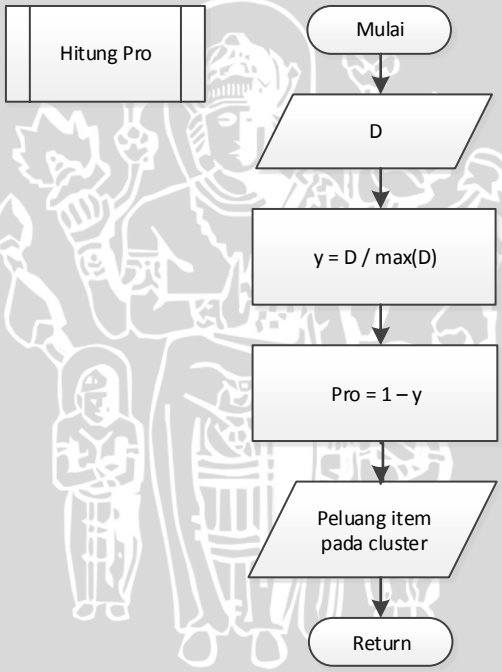


Gambar 3.3 *Flowchart* Jarak *Item* Dengan Pusat *Cluster*

Sedangkan langkah-langkah dari subproses hitung *Pro* adalah sebagai berikut:

1. Masukkan (*input*) berupa jarak item dengan pusat *cluster* (*D*).
2. Mengambil nilai *max(D)* pada *cluster*.
3. Lakukan pembagian (*y*) antara nilai *(D)* dengan nilai *max(D)*.
4. Lakukan pengurangan (*Pro*) nilai 1 dengan hasil yang didapat pada langkah 3 (y)
5. Keluaran (*output*) berupa peluang item pada *cluster*.

Adapun *flowchart* dari subproses hitung *Pro* seperti ditunjukkan pada Gambar 3.4 berikut.



Gambar 3.4 *Flowchart* Hitung *Pro*

Dari tahapan *clustering* dihasilkan sejumlah *k cluster*. Misalkan *cluster* yang dibentuk adalah 2 buah *cluster*. Dari hasil tersebut dijadikan model matriks, dimana matriks ini disebut sebagai matriks *group-rating* seperti yang ditunjukkan pada Tabel 3.7 berikut ini.

Tabel 3.7 Matriks *Group-rating*

movie_title	*Cluster* 1	*Cluster* 2
A	n	n
B	n	n
C	n	n
D	n	n
E	n	n

3.4.2 Perhitungan Nilai *Similarity*

Setelah didapatkan dua buah matriks, yaitu matriks *item*-rating dan matriks *group-rating*, maka tahapan selanjutnya adalah melakukan perhitungan nilai *similarity* pada ke dua matriks tersebut.

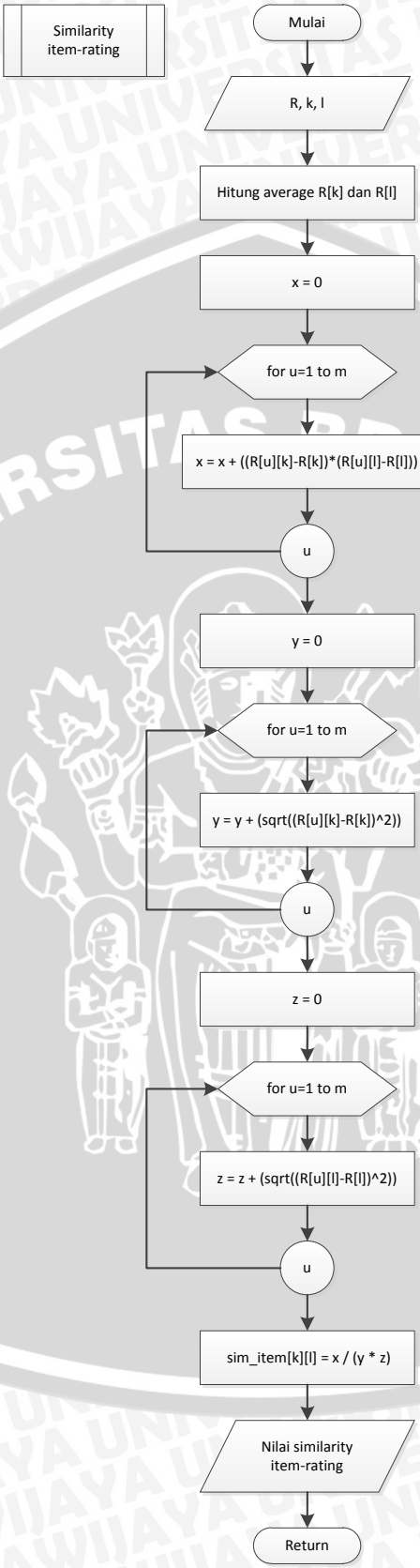
3.4.2.1 *Similarity Item-rating*

Matriks *item-rating* yang didapat pada proses sebelumnya, selanjutnya dihitung nilai *similarity*-nya. Perhitungan nilai *similarity item-rating* menggunakan algoritma *pearson correlation-based*. Langkah-langkah dalam proses perhitungan nilai *similarity item-rating* adalah sebagai berikut :

1. Masukkan (*input*) berupa matriks *item-rating* (*R*), indeks pada *item k* (*k*) dan indeks pada *item l* (*l*).
2. Mengambil nilai rating *user u* pada *item k* (*R[u][k]*) dan *l* (*R[u][l]*).
3. Menghitung rata-rata *rating* pada *item k* (*R[k]*) dan *l* (*R[l]*).
4. Lakukan pengurangan antara nilai dari *rating user u* pada *item k* (*R[u][k]*) dengan rata-rata rating pada *item k* (*R[k]*).
5. Lakukan pengurangan antara nilai dari *rating user u* pada *item l* (*R[u][l]*) dengan rata-rata rating pada *item l* (*R[l]*).
6. Lakukan perhitungan (*x*) dengan mengkalikan hasil yang didapat pada langkah 4 dengan hasil yang didapat pada langkah 5 dan dilakukan iterasi sebanyak user yang merating *item k* dan *l*.
7. Lakukan proses pada langkah 6 sebanyak jumlah user (*m*).

8. Lakukan perhitungan (*y*) dengan mengkuadratkan hasil yang didapat pada langkah 4, kemudian hasilnya dilakukan operasi akar dan dilakukan iterasi sebanyak *user* yang merating *item k*.
9. Lakukan proses pada langkah 8 sebanyak jumlah *user* (*m*).
10. Lakukan perhitungan (*z*) dengan mengkuadratkan hasil yang didapat pada langkah 5, kemudian hasilnya dilakukan operasi akar dan dilakukan iterasi sebanyak *user* yang merating *item l*.
11. Lakukan proses pada langkah 10 sebanyak jumlah *user* (*m*).
12. Lakukan perhitungan perkalian antara hasil yang didapat pada langkah 8 dengan hasil yang didapat pada langkah 10.
13. Kemudian lakukan perhitungan (*sim_item[k][l]*) dengan membagi hasil yang didapat pada langkah 6 dengan hasil yang didapat pada langkah 12.
14. Keluaran (*output*) berupa nilai *similarity item-rating*.

Adapun *flowchart* dari proses perhitungan nilai *similarity* matriks *item-rating* seperti ditunjukkan pada Gambar 3.5.



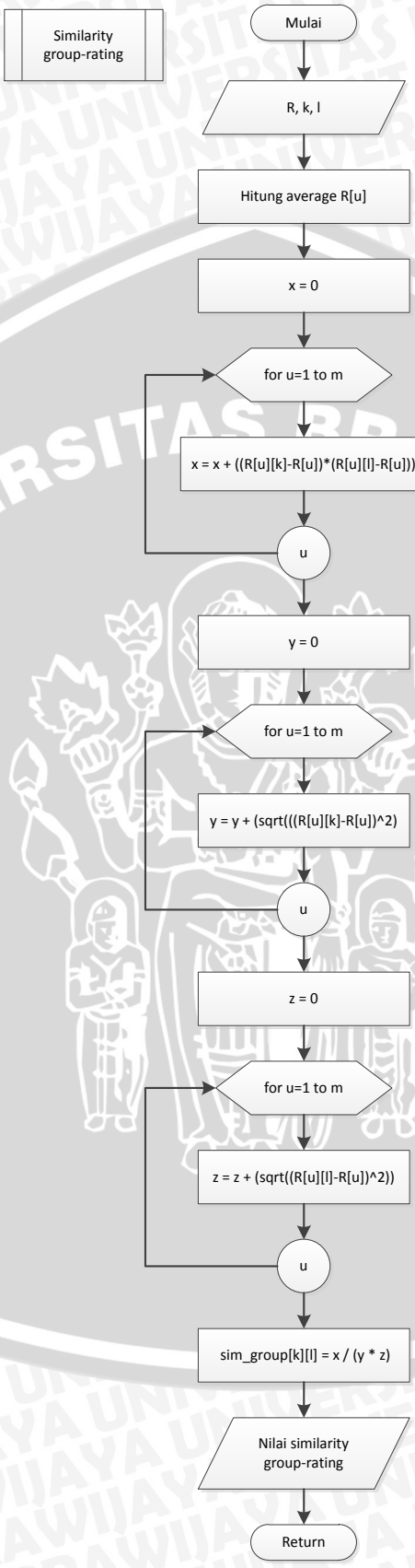
Gambar 3.5 *Flowchart Similarity Item-rating*

3.4.2.2 *Similarity Group-rating*

Selain matriks *item*-rating, matriks *group-rating* juga dilakukan perhitungan *similarity*. Perhitungan nilai *similarity* matriks *group-rating* menggunakan algoritma *adjusted-cosine*. Langkah-langkah dalam proses perhitungan nilai *similarity group-rating* adalah sebagai berikut :

1. Masukkan (*input*) berupa matriks *group-rating* (*R*), indeks pada *item k* (*k*) dan indeks pada *item l* (*l*).
2. Mengambil nilai rating *user u* pada *item k* (*R[k]*) dan *l* (*R[l]*).
3. Menghitung rata-rata *rating* dari *user u* (*R[u]*).
4. Lakukan pengurangan antara nilai dari *rating user u* pada *item k* (*R[k]*) dengan rata-rata rating dari *user u* (*R[u]*).
5. Lakukan pengurangan antara nilai dari *rating user u* pada *item l* (*R[l]*) dengan rata-rata rating dari *user u* (*R[u]*).
6. Lakukan perhitungan (*x*) dengan mengkalikan hasil yang didapat pada langkah 4 dengan hasil yang didapat pada langkah 5 dan dilakukan iterasi sebanyak *user* yang me-*rating item k* dan *l*.
7. Lakukan proses pada langkah 6 sebanyak jumlah *user* (*m*).
8. Lakukan perhitungan (*y*) dengan mengkuadratkan hasil yang didapat pada langkah 4, kemudian hasilnya dilakukan operasi akar dan dilakukan iterasi sebanyak *user* yang merating *item k*.
9. Lakukan proses pada langkah 8 sebanyak jumlah *user* (*m*).
10. Lakukan perhitungan (*z*) dengan mengkuadratkan hasil yang didapat pada langkah 5, kemudian hasilnya dilakukan operasi akar dan dilakukan iterasi sebanyak *user* yang merating *item l*.
11. Lakukan proses pada langkah 10 sebanyak jumlah *user* (*m*).
12. Lakukan perhitungan perkalian antara hasil yang didapat pada langkah 8 dengan hasil yang didapat pada langkah 10.
13. Kemudian lakukan perhitungan (*sim_group[k][l]*) dengan membagi hasil yang didapat pada langkah 6 dengan hasil yang didapat pada langkah 12.
14. Keluaran (*output*) berupa nilai *similarity group-rating*.

Adapun *flowchart* dari proses perhitungan nilai *similarity* matriks *group-rating* seperti ditunjukkan pada Gambar 3.6.



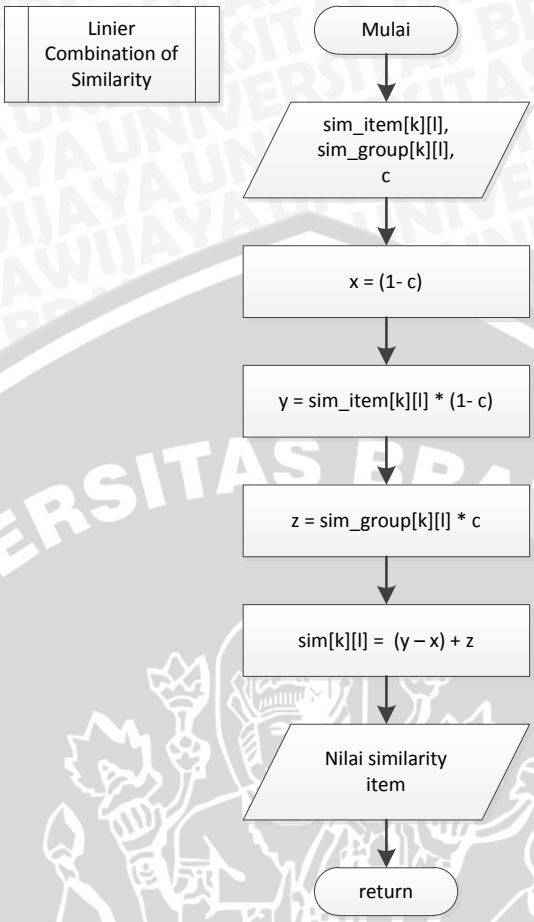
Gambar 3.6 *Flowchart Similarity Group-rating*

3.4.2.3 *Linier Combination of Similarity*

Setelah didapatkan nilai *similarity item-rating* dan *group-rating*, kemudian dilakukan perhitungan dengan menggabungkan nilai keduanya. Perhitungan dengan menggunakan algoritma *linier combination of similiraty*. Langkah-langkah dalam proses perhitungan nilai *linier combination of similiraty* adalah sebagai berikut.

1. Masukkan (*input*) nilai similarity *item*-rating (*sim_item[k][l]*) dan *group-rating* (*sim_group[k][l]*) serta koefisien *c*.
2. Hitung (*x*) nilai dari (1 – koefisien *c*).
3. Lakukan perhitungan (*y*) dengan mengkalikan nilai *similarity item-rating* (*sim_item[k][l]*) dengan hasil dari langkah 2 (*x*).
4. Lakukan perhitungan (*z*) dengan mengkalikan *nilai similarity group-rating* (*sim_group[k][l]*) dengan koefisien *c*.
5. Kemudian lakukan perhitungan (*sim[k][l]*) dengan menjumlahkan hasil yang didapat pada langkah 3 dengan langkah 4
6. Keluaran (*output*) berupa nilai *similarity item.*

Adapun *flowchart* dari proses perhitungan nilai *linier combination of similarity* seperti ditunjukkan pada Gambar 3.7.



Gambar 3.7 *Flowchart Linier Combination of Similiraty*

3.4.3 Perhitungan Prediksi Rating

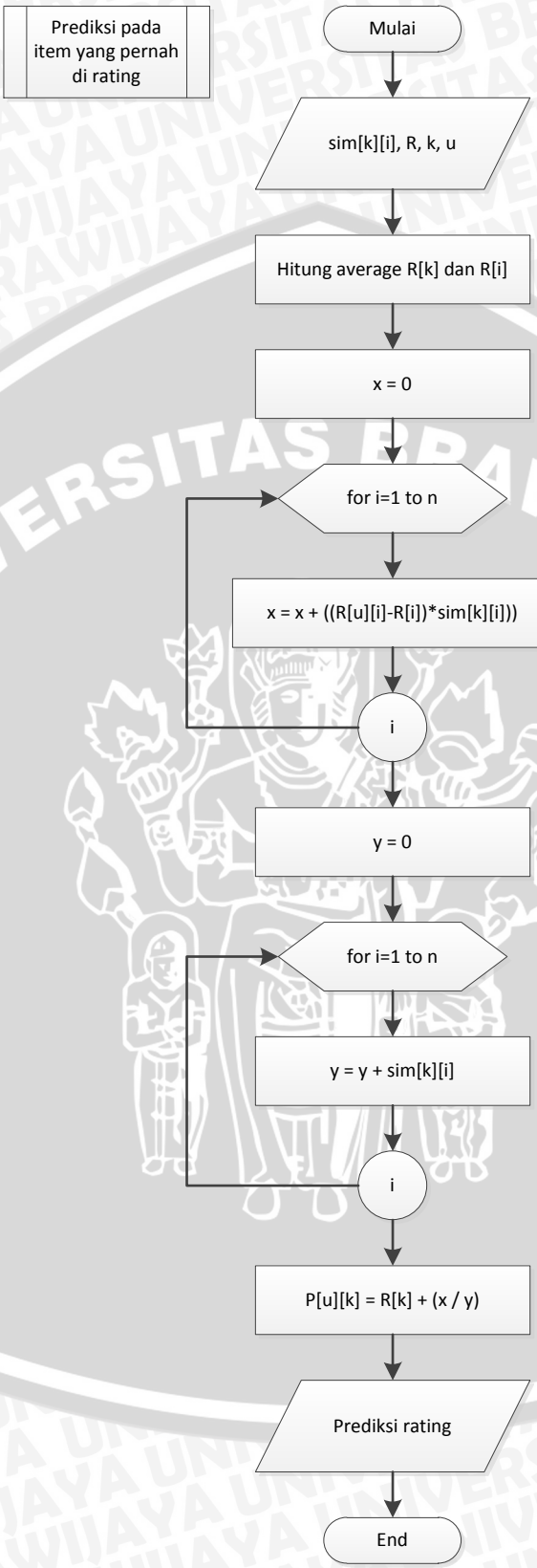
Dalam menghitung prediksi rating suatu *item* dibagi menjadi dua perhitungan, yaitu untuk *item* yang sudah pernah *di rating* oleh *user* (*non cold-start problem*) dan pada *item* yang belum pernah *di rating* oleh *user* (*cold-start problem*). Perhitungan prediksi bertujuan untuk mendapatkan *Top-N* rekomendasi, dimana *Top-N* rekomendasi ini akan menghasilkan *item-item* yang akan dijadikan rekomendasi kepada *user*.

3.4.3.1 Prediksi Pada *Item* yang Pernah di Rating

Untuk menghitung prediksi *rating* pada *item k* yang telah *di rating*, digunakan metode *weighted average of deviation* Langkah-langkah dalam proses perhitungan nilai prediksi pada *item* yang sudah pernah di rating adalah sebagai berikut.

1. Masukkan (*input*) berupa nilai *similiraty item* (*sim[k][i])*, matrik *item-rating* (*R*), indeks pada *item k* (*k*) dan *user u* (*u*).
2. Hitung rata-rata rating pada *item k* (*R[k]*).
3. Hitung rata-rata rating pada *item i* (*R[i]*).
4. Mengambil nilai rating *user u* pada *item i* (*R[u][i]*).
5. Lakukan perhitungan (*x*) pengurangan antara nilai dari *rating user u* pada *item i* (*R[u][i]*) dengan rata-rata rating pada *item i* (*R[i]*) kemudian hasilnya kalikan dengan nilai *similarity item* (*sim[k][i])* dan dilakukan iterasi sebanyak jumlah *neighboar item k*.
6. Lakukan proses pada langkah 5 sebanyak jumlah *user* (*m*).
7. Jumlahkan (*y*) nilai *similiraty item* (*sim[k][i])* sebanyak jumlah *neighbor* dari *item k.*
8. Lakukan proses pada langkah 7 sebanyak jumlah *user* (*m*).
9. Dari hasil yang didapat pada langkah 2 (*R[k]*), lakukan perhitungan (*P[u][k]*) dengan menambahkan hasil pembagian antara langkah 5 dengan langkah 7.
10. Keluaran (*output*) berupa prediksi rating.

Adapun *flowchart* dari proses perhitungan prediksi rating pada *item* yang sudah pernah *di rating* seperti ditunjukkan pada Gambar 3.8.



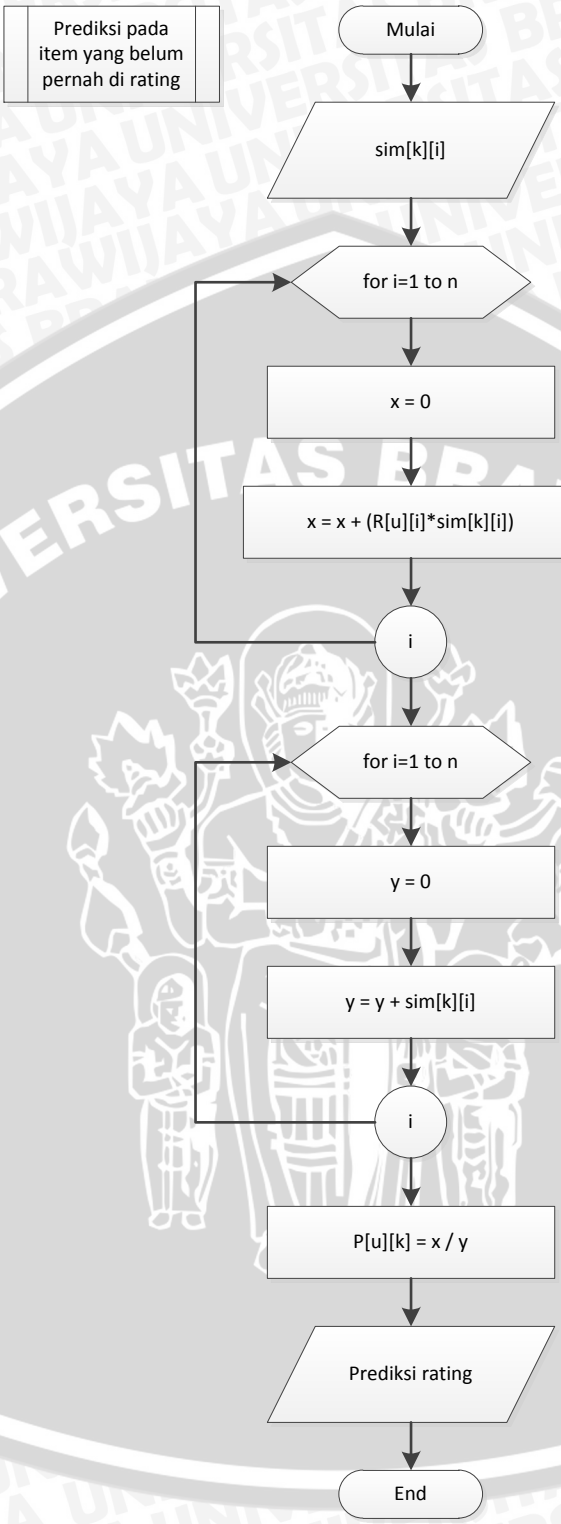
Gambar 3.8 *Flowchart* Prediksi Pada *Item* Yang Pernah di Rating

3.4.3.2 Prediksi Pada *Item* yang Belum Pernah di Rating

Untuk menghitung prediksi *rating* pada *item k* yang belum pernah *di* rating, digunakan metode *weighted sum* Langkah-langkah dalam proses perhitungan nilai prediksi pada *item* yang belum pernah di rating adalah sebagai berikut.

1. Masukkan (*input*) nilai *similiraty item* (*sim[k][i])*, matrik *item-rating* (*R*) dan *user u* (*u*).
2. Mengambil nilai rating *user u* pada *item i* (*R[i]*).
3. Lakukan perhitungan (*x*) dengan mengkalikan antara nilai rating *user u* pada *item i* (*R[i]*) dengan nilai *similarity item* (*sim[k][i])* dan dilakukan iterasi sebanyak jumlah *neighboar item k.*
4. Lakukan proses pada langkah 3 sebanyak jumlah *user* (*m*).
5. Jumlahkan (*y*) nilai *similiraty item* (*sim[k][i])* sebanyak jumlah *neighbor* dari *item k.*
6. Lakukan proses pada langkah 5 sebanyak jumlah *user* (*m*).
7. Kemudian lakukan perhitungan (*P[u][k])* dengan membagi hasil yang didapat pada langkah 3 dengan hasil yang didapat pada langkah 5.
8. Keluaran (*output*) berupa prediksi rating.

Adapun *flowchart* dari proses perhitungan prediksi rating dari *item* yang belum pernah di rating seperti ditunjukkan pada Gambar 3.9.



Gambar 3.9 *Flowchart* Prediksi Pada *Item* Yang Belum Pernah di Rating

3.4.3.3 Hasil Prediksi

Dari proses sebelumnya dihasilkan prediksi rating dari *user* pada *item*. Dimana *item* tersebut belum pernah di rating oleh *user*. Prediksi rating ini nantinya akan dijadikan atau diberikan sebagai rekomendasi kepada *user*.

3.4.3.4 Rekomendasi *Non-Personalized*

Kegunaan rekomendasi *non-personalized* adalah untuk memberikan pilihan bagi *user* yang tidak terdaftar (selanjutnya disebut sebagai *visitor*) untuk tetap bisa mendapatkan rekomendasi *item*.

Untuk dapat memberikan rekomendasi kepada *visitor* adalah dengan menyimpan hasil rekomendasi dari *user* yang sebelumnya melakukan *login* untuk kemudian diberikan kepada *user* lain yang tidak terdaftar.

3.4.3.4 Rekomendasi *Personalized*

Rekomendasi *personalized* diberikan kepada *user* yang terdaftar dalam sistem. Setelah didapatkan prediksi rating dari *user* pada *item*, dibuat daftar rangking dari terbesar berdasarkan nilai prediksi rating pada *item*, yang berisi *N item* (rekomendasi *Top-N*) yang mempunyai kemungkinan untuk dipilih atau disukai oleh *user*, dimana *item-item* ini belum pernah di rating oleh *user*.

Misal hasil *item* yang di rekomendasikan kepada *user* Z seperti ditunjukkan pada Tabel 3.8 berikut ini.

Tabel 3.8 Hasil Rekomendasi

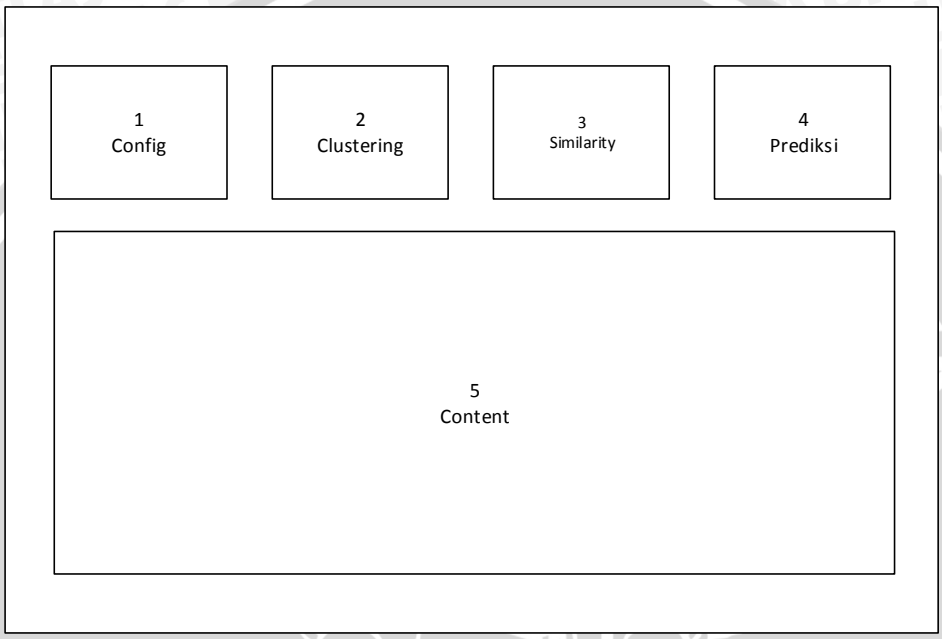
rangking	movie_title	nilai
1	C	n
2	E	n

3.5 Perancangan Antarmuka

Antarmuka yang akan dibangun dibagi menjadi tiga bagian yaitu antarmuka sistem, katalog *item*, detail *item* dan rekomendasi. Antarmuka katalog *item*, detail *item* dan rekomendasi adalah antarmuka untuk user beinteraksi dengan aplikasi. Sedangkan antarmuka sistem adalah antarmuka bagaimana proses pada sistem rekomendasi.

3.5.1 Antarmuka Sistem

Antarmuka sistem merupakan tampilan bagaimana proses sistem rekomendasi berjalan. Di dalam antarmuka ini terdapat beberapa proses yang ada pada sistem rekomendasi, yaitu *config*, *clustering*, *similarity* dan prediksi. Adapun antarmuka sistem ditunjukkan pada Gambar 3.10 berikut.



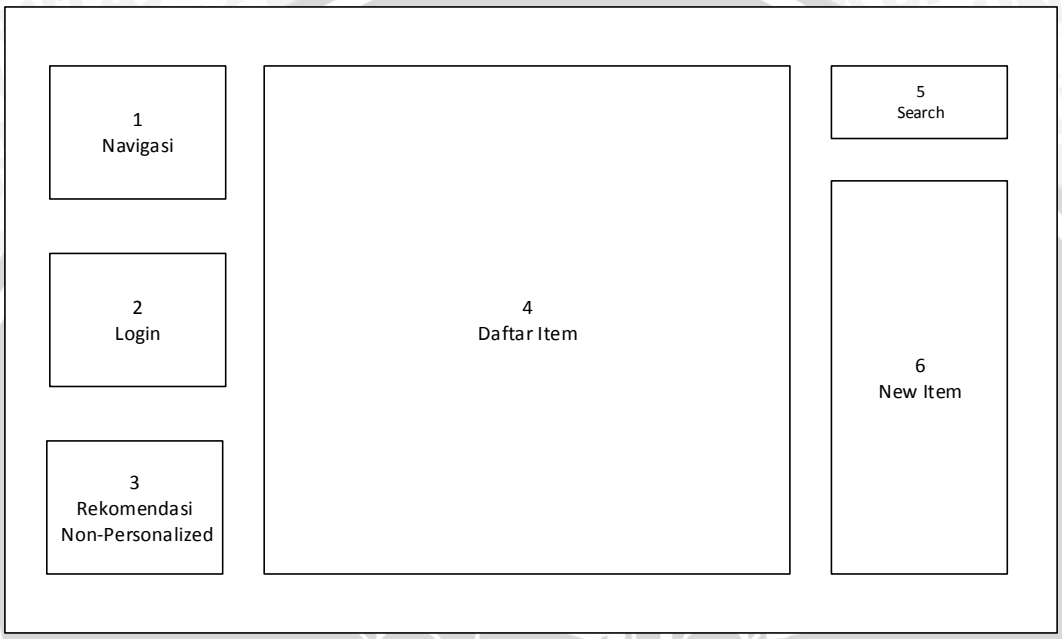
Gambar 3.10 Antarmuka Sistem

Penjelasan dari bagian-bagian antarmuka sistem di atas adalah :

1. *Config*, berfungsi untuk memberikan nilai pada jumlah *cluster* pada proses *clustering* dan nilai koefisien *c* pada proses *similarity*.
2. *Clustering*, berfungsi untuk menampilkan hasil perhitungan *clustering*.
3. *Similarity*, berfungsi untuk menampilkan hasil perhitungan *similarity*.
4. Prediksi, berfungsi untuk menampilkan hasil perhitungan prediksi.
5. *Content*, berisi tampilan atau isi dari setiap menu yang aktif

3.5.2 Antarmuka *Home*

Antarmuka *home* merupakan tampilan awal sistem. Pada antarmuka *home* terdapat sejumlah daftar *item* dimana setiap *item* yang ditampilkan disertai dengan besarnya rating. Pada antarmuka *home* juga terdapat menu untuk *login user*. Adapun antarmuka *home* ditunjukkan pada Gambar 3.11 berikut.



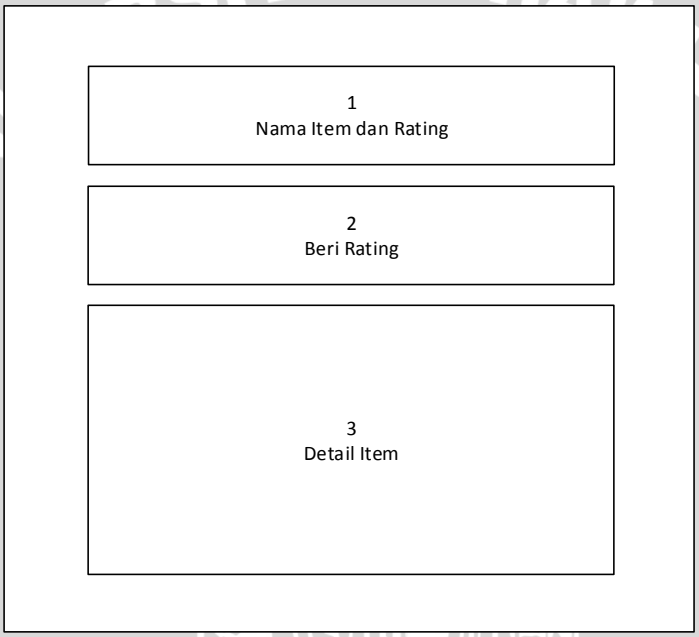
Gambar 3.11 Antarmuka *Home*

Penjelasan dari bagian-bagian antarmuka *home* di atas adalah :

1. Navigasi berisi menu home dan katalog, dimana menu katalog menampilkan katalog semua *item*.
2. Login, berfungsi untuk *login user*.
3. Rekomendasi *Non-Personalized*, berfungsi untuk menampilkan 15 item yang direkomendasikan kepada *user* yang tidak terdaftar atau tidak *login*.
4. Daftar *item*, berfungsi untuk menampilkan nama *item*, rating *item* (dengan digambarkan 5 bintang) beserta detail *item*.
5. *Search*, berfungsi untuk pencarian *item*.
6. *New item*, berfungsi untuk menampilkan *item* yang baru ditambahkan.

3.5.3 Antarmuka Detail *Item*

Pada antarmuka detail *item* menampilkan detail informasi dari sebuah *item*, termasuk juga rating dari *item* serta jumlah *user* yang telah memberi rating pada *item* tersebut. Rating dari *item* digambarkan dengan jumlah bintang yaitu sebanyak 5 bintang. Pada halaman ini juga terdapat menu untuk *user* memberikan rating pada *item*. Untuk bisa memberikan rating pada *item*, *user* (terdaftar) harus melakukan *login* terlebih dahulu. Adapun antarmuka detail *item* ditunjukkan pada Gambar 3.12 berikut.



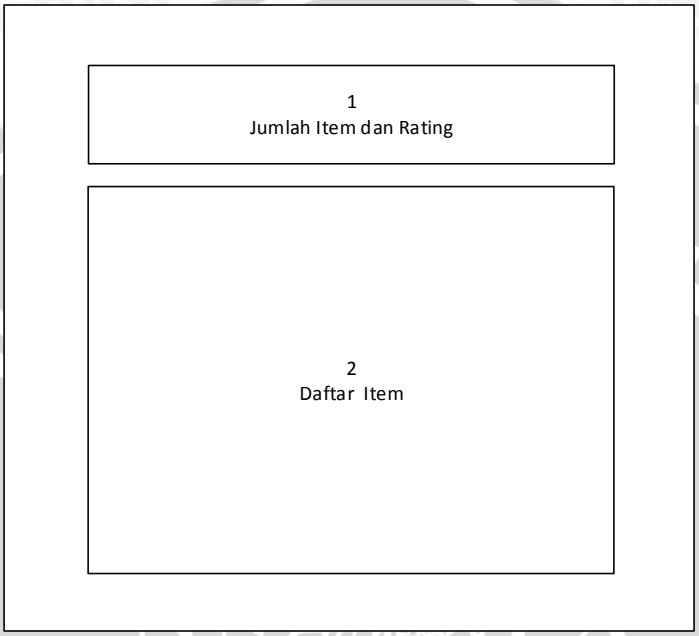
Gambar 3.12 Antarmuka Detail *Item*

Penjelasan dari bagian-bagian antarmuka detail *item* di atas adalah :

1. Nama *item* dan rating, berfungsi untuk menampilkan nama *item*, rating *item* dan jumlah *user* yang telah memberi rating pada *item* tersebut.
2. Beri rating, berfungsi untuk *user* memberikan rating pada *item*.
3. Detail *item*, berfungsi untuk menampilkan detail dari *item* yaitu *duration, description, director, writers, start, full detail* dan *image* dari *item*.

3.5.4 Antarmuka Rating *User*

Pada antarmuka rating *user* menampilkan daftar *item* yang telah di rating oleh *user* beserta rating yang diberikan oleh *user* pada *item* tersebut. Dalam antarmuka rating *user* juga terdapat jumlah *item* yang telah di rating beserta rata-rata ratingnya. Adapun antarmuka rating *user* ditunjukkan pada Gambar 3.13 berikut.



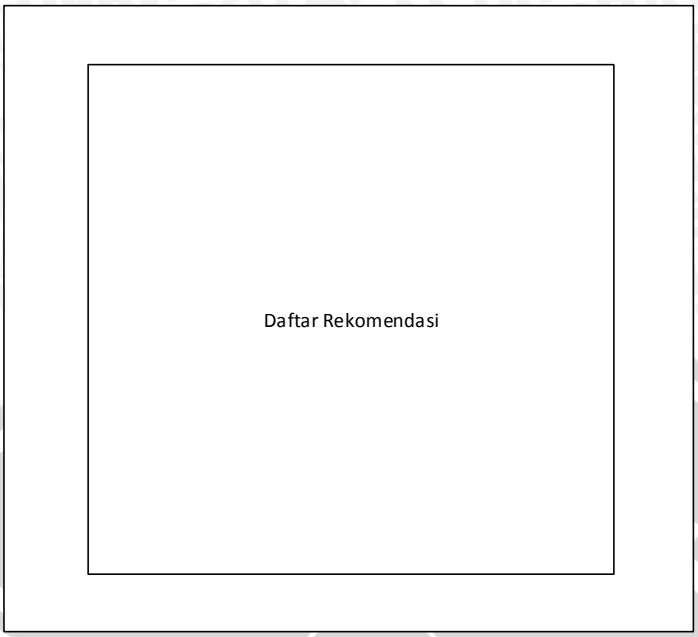
Gambar 3.13 Antarmuka Rating *User*

Penjelasan dari bagian-bagian antarmuka rating *user* di atas adalah :

1. Jumlah *item* dan rating, berfungsi untuk menampilkan jumlah *item* yang telah di rating oleh *user* beserta rata-rata ratingnya.
2. Daftar *item*, berfungsi untuk menampilkan *id item*, nama *item* dan rating dari *item* tersebut.

3.5.5 Antarmuka Rekomendasi

Pada antarmuka rekomendasi menampilkan daftar *item* yang direkomendasikan kepada *user*. Setiap *item* yang direkomendasikan kepada *user* disertai besarnya nilai prediksi dan ditampilkan dalam urutan dari nilai prediksi terbesar sampai dengan terkecil. Jumlah *item* yang direkomendasikan kepada *user* dibatasi sebanyak 15 *item*. Adapun antarmuka rekomendasi ditunjukkan pada Gambar 3.14 berikut.



Gambar 3.14 Antarmuka Rekomendasi

Penjelasan dari bagian antarmuka rekomendasi di atas adalah daftar rekomendasi, berfungsi untuk menampilkan daftar *item* yang direkomendasikan kepada *user* beserta nilai prediksi dari setiap *item*.

3.6 Perancangan Tabel

3.6.1 Tabel Data

Tabel data digunakan untuk menyimpan data rating yang diberikan oleh *user* terhadap *item* serta waktu ketika *user* tersebut memberikan rating pada *item*.

Tabel 3.9 Atribut Tabel Data

Field	Type	Keterangan
user_id	int(10)	Kode user yang memberi rating
item_id	int(10)	Kode *item* yang diberi rating
rating	float(10)	Nilai rating
timestamp	timestamp	Waktu ketika user merating *item*

3.6.2 Tabel *Item*

Tabel *item* digunakan untuk menyimpan data film dan juga *genre* dari film yang berjumlah 19 *genre*.

Tabel 3.10 Atribut Tabel *Item*

Field	Type	Keterangan
item_id	int(10)	Kode film
movie_title	int(100)	Judul film
release_date	varchar(10)	Tanggal release film
video_release_date	varchar(10)	Tanggal release video
imdb_irl	varchar(100)	Detail film
unknown	int(2)	Genre unknown
action	int(2)	Genre action
adventure	int(2)	Genre adventure
animation	int(2)	Genre animation
childrens	int(2)	Genre childrens
comedy	int(2)	Genre comedy
crime	int(2)	Genre crime
documentary	int(2)	Genre documentary
drama	int(2)	Genre drama
fantasy	int(2)	Genre fantasy
filmnoir	int(2)	Genre filmnoir
horror	int(2)	Genre horror
musical	int(2)	Genre musical
mystery	int(2)	Genre mystery
romance	int(2)	Genre romance
scifi	int(2)	Genre scifi
thriller	int(2)	Genre thriller
war	int(2)	Genre war
western	int(2)	Genre western

3.6.3 Tabel *User*

Tabel *user* digunakan untuk menyimpan data entitas setiap *user*. *User* untuk bisa merating *item* dan mendapatkan rekomendasi *item* harus *login* terlebih dahulu dengan memasukkan *username* dan *password*. Karena pada dataset film untuk *field username* dan *password* tidak disertakan, maka perlu ditambahkan *field* tersebut pada tabel *user*.

Tabel 3.11 Atribut Tabel *User*

Field	Type	Keterangan
user_id	int(10)	Kode user
username	varchar(20)	Nama user
password	varchar(20)	Password user
age	int(3)	Umur user
gender	varchar(2)	Jenis kelamin user
occupation	varchar(20)	Pekerjaan user
zipcode	int(6)	Kode pos user

3.7 Perancangan Uji Coba

Setelah sistem selesai dibuat, langkah selanjutnya adalah melakukan pengujian terhadap sistem tersebut. Pengujian dilakukan untuk mengetahui tingkat akurasi dari hasil rekomendasi film menggunakan *item-based clustering hybrid method*.

3.7.1 Bahan Pengujian

Pada penelitian ini, data yang digunakan dalam pengujian sistem adalah data film yang berasal dari *GroupLens*. Data ini terdiri dari 943 *user*, 1682 film, 100.000 rating. Nilai rating yang terdapat pada data tersebut adalah 1, 2, 3, 4 dan 5.

3.7.2 Tujuan Pengujian

Tujuan dari pelaksanaan pengujian yaitu :

1. Memeriksa kesesuaian perancangan dengan penerapan.
2. Mengukur atau mengetahui tingkat akurasi dari hasil rekomendasi yang diberikan.

3.7.3 Skenario Pengujian

Pada penelitian ini akan dilakukan tiga pengujian yaitu pertama pengujian pengaruh jumlah *cluster* terhadap tingkat akurasi hasil prediksi rating, pengaruh koefisien *c* terhadap tingkat akurasi hasil prediksi rating dan yang ketiga yaitu pengaruh jumlah data latih dan data uji terhadap tingkat akurasi hasil prediksi rating. Pengujian dilakukan terhadap nilai *MAE* (*Mean Absolute Error*).

1. Pengujian pengaruh jumlah *cluster* terhadap tingkat akurasi hasil prediksi rating. Pengujian ini dilakukan untuk mengetahui pengaruh jumlah *cluster* terhadap tingkat akurasi hasil prediksi rating. Jumlah *cluster* yang akan diuji sebanyak 10 sampai 100 *cluster* dengan *interval* 10 dan nilai koefisien *c* ditetapkan sebesar 0,5. Pada pengujian ini dataset dibagi menjadi dua bagian, yaitu 80% sebagai data latih dan 20% sebagai data uji. Pengujian yang dilakukan ada dua, yaitu yang pertama pada data latih dan data uji yang sama pada tiap jumlah *cluster* dan yang kedua yaitu pada data latih dan data uji yang berbeda pada tiap jumlah *cluster*.

Tabel 3.12 Pengujian Pengaruh Jumlah *Cluster* Terhadap *MAE*

Jumlah *Cluster*	Koefisien c	Nilai MAE
10	0,5	
20	0,5	
30	0,5	
40	0,5	
50	0,5	
60	0,5	
70	0,5	

80	0,5	
90	0,5	
100	0,5	

2. Pengujian pengaruh koefisien *c* terhadap tingkat akurasi hasil prediksi rating. Pengujian ini dilakukan untuk mengetahui pengaruh koefisien *c* terhadap tingkat akurasi hasil prediksi rating. Nilai koefisien *c* yang digunakan untuk pengujian adalah 0,1 sampai 0,9 dengan *interval* 0,1 dan ditentukan jumlah *cluster* sebanyak 10. Pada pengujian ini dataset dibagi menjadi dua bagian, yaitu 80% sebagai data latih dan 20% sebagai data uji.

Tabel 3.13 Pengujian Pengaruh Koefisien *c* Terhadap *MAE*

Koefisien c	**Jumlah *Cluster***	**Nilai MAE**
0,1	10	
0,2	10	
0,3	10	
0,4	10	
0,5	10	
0,6	10	
0,7	10	
0,8	10	
0,9	10	

3. Pengujian pengaruh jumlah data latih dan data uji terhadap tingkat akurasi hasil prediksi rating.

   Pengujian ini dilakukan untuk mengetahui pengaruh jumlah data latih dan data uji terhadap tingkat akurasi hasil prediksi rating. Nilai koefisien *c* yang digunakan adalah sebesar 0,5 dan ditentukan jumlah *cluster* sebanyak 10.

Tabel 3.14 Pengujian Pengaruh Jumlah Data Latih Dan Data Uji Terhadap *MAE*

Data Latih	Data Uji	Nilai MAE
20 %	80 %	
40 %	60 %	
60 %	40 %	
80 %	20 %	

3.8 Perhitungan Manual

Berikut ini contoh perhitungan manual yang dilakukan sistem dalam memberikan rekomendasi film kepada *user*.

Jumlah film yang akan digunakan dalam perhitungan ini adalah sebanyak 10 film, terdiri dari 9 film yang diambil dari *dataset GroupLens* dan 1 film diluar *dataset* dimana film ini belum pernah di rating. Atribut pada film yang digunakan dalam perhitungan manual adalah adalah rating, judul dan *genre* film. Serta menggunakan *id user* yang merating film, yaitu sebanyak 5 *user*. Sedangkan *genre* film yang digunakan dalam perhitungan manual ini adalah sebanyak 5 *genre*, yaitu *crime, drama, horor, romance* dan *war*. Perincian data yang digunakan pada perhitungan manual seperti ditunjukkan pada Tabel 3.15 berikut ini.

Tabel 3.15 Data Perhitungan Manual

user_id	movie_title	Genre	rating
22	Taxi Driver	drama, war	4
22	Bound	crime, drama, romance	4
22	Psycho	horror, romance	2
22	Crimson Tide	drama, war	2
22	Ran	drama, war	1
22	Forrest Gump	romance, war	2
94	Bound	crime, drama, romance	4
95	Rob roy	romance, war	2
94	True Romance	crime, romance	4
94	Crimson Tide	drama, war	1

94	Crying game	drama, romance, war	1
166	Taxi Driver	drama, war	2
166	Bound	crime, drama, romance	5
166	Ran	drama, war	3
166	Crying game	drama, romance, war	1
166	Forrest Gump	romance, war	2
186	Taxi Driver	crime, drama, romance	5
186	Rob roy	romance, war	1
186	Crimson Tide	drama, war	4
186	Ran	drama, war	1
186	Crying game	drama, romance, war	2
244	True Romance	crime, romance	2
244	Psycho	horror, romance	2
244	Casablanca	drama, romance, war	1
244	Crying game	drama, romance, war	5
244	Forrest Gump	romance, war	3
	Big fish	drama, romance	

Langkah 1

Membangun matriks *item-rating*. Dari Tabel 3.15 diatas dijadikan model matriks yakni matriks *item*-rating seperti ditunjukkan pada Tabel 3.16 berikut .

Tabel 3.16 Matriks *Item-rating*

movie_title	**user_id**				
	22	94	166	186	244
Taxi Driver	4		2	5	
Rob Roy				1	
Bound	4	4	5		
True Romance		4			2
Psycho	2				2

Crimson Tide	2	1		4	
Ran	1		3	1	
Crying Game		1	1	2	5
Forrest Gump	2		2		3
Big Fish					

Langkah 2

Membangun matriks *group-rating*. Dari data pada Tabel 3.15 diatas dihitung terlebih dahulu rata-rata rating pada setiap *genre item* untuk digunakan pada proses *clustering* menggunakan algoritma *K-means clustering*. Setelah dilakukan proses *clustering*, dihitung keterkaitan atau peluang tiap *item* terhadap *cluster*.

Tabel 3.17 *Movie Genre*

movie_title	**Genre**				
	crime	drama	horor	romance	war
Taxi Driver		1			1
Rob Roy				1	1
Bound	1	1		1	
True Romance	1			1	
Psycho			1	1	
Crimson Tide		1			1
Ran		1			1
Crying Game		1		1	1
Forrest Gump				1	1
Big Fish	1			1	

Sesuai dengan *genre* yang dimiliki setiap item diatas, dihitung rata-rata rating pada *genre* tersebut, seperti ditunjukkan pada tabel 3.18 berikut.

Tabel 3.18 Rata-rata Rating *Genre*

Genre	Userid					Rata-rata
	22	94	166	186	244	
Crime	(4+0+0) /3 =1,33	(4+4+0) /3 =2,67	(5+0+0) /3 =1,67	(0+0+0) /3 =0	(0+2+0) /3 =0,67	1,27
Drama	(4+4+2+1+0)/5 =2,2	(0+4+1+0+1) /5 =1,2	(2+5+0+3+1) /5 =2,2	(5+0+4+1+2) /5 =2,4	(0+0+0+0+5) /5 =1	1,8
Horor	(2) /1 =2	(0) /1 =0	(0) /1 =0	(0) /1 =0	(2) /1 =2	0,8
Romance	(0+4+0+2+0+2+0) /7 =1,14	(0+4+4+0+1+0+0) /7 =1,28	(0+5+0+0+1+2+0) /7 =1,14	(1+0+0+0+2+0+0) /7 =0,43	(0+0+2+2+5+3+0) /7 =1,71	1,14
War	(4+0+2+1+0+2) /6 =1,5	(0+0+1+0+1+0) /6 =0,33	(2+0+0+3+1+2) /6 =1,33	(5+1+4+1+2+0) /6 =2,17	(0+0+0+0+5+3) /6 =1,33	1,33

Setelah didapatkan rata-rata rating pada setiap *genre*, data tersebut digunakan untuk proses *cluster*ing, seperti ditunjukkan pada Tabel 3.19 berikut.

Tabel 3.19 Data *User Movie*

movie_title	user_id				
	22	94	166	186	244
Taxi Driver	0	1,8	0	0	1,33
Rob Roy	0	0	0	1,14	1,33
Bound	1.27	1,8	0	1,14	0
True Romance	1,27	0	0	1,14	0
Psycho	0	0	0,8	1,14	0
Crimson Tide	0	1,8	0	0	1,33
Ran	0	1,8	0	0	1,33
Crying Game	0	1,8	0	1,14	1,33
Forrest Gump	0	0	0	1,14	1,33
Big Fish	1,27	0	0	1,14	0

Untuk membangun *matriks group-rating*, data pada Tabel 3.19 diatas dilakukan proses *clustering*. Proses *clustering* menggunakan algoritma *K-Means clustering*. *Cluster* yang akan dibentuk adalah sebanyak 2 *cluster* (k=2), dengan pusat *cluster* C_1=(0;0;0;1,14;1,33); C_2=(0;0;0,8;1,14;0).

Mencari jarak data terhadap pusat *cluster* dengan menggunakan *Eucledian Distance*.

Contoh perhitungan jarak antara *record* ke-1 terhadap pusat *cluster* pertama dan kedua adalah :

$$C_{1,1}= \sqrt{(0-0)^2 + (1{,}8-0)^2 + (0-0)^2 + (0-1{,}14)^2 + (1{,}33-1{,}33)^2} = 2{,}13063$$

$$C_{1,2}= \sqrt{(0-0)^2 + (1{,}8-0)^2 + (0-0{,}8)^2 + (0-1{,}14)^2 + (1{,}33-0)^2} = 2{,}636001$$

Hasil perhitungan jarak data terhadap pusat *cluster* selengkapnya seperti pada Tabel 3.20 berikut :

Tabel 3.20 Jarak Data Terhadap Pusat *Cluster*

movie_title	**user_id**					***Cluster***	
	22	94	166	186	244	C_1	C_2
Taxi Driver	0	1,8	0	0	1,33	2,130634	2,636001
Rob Roy	0	0	0	1,14	1,33	0	1,552063
Bound	1.27	1,8	0	1,14	0	2,573286	2,343694
True Romance	1,27	0	0	1,14	0	1,838967	1,500966
Psycho	0	0	0,8	1,14	0	1,552063	0
Crimson Tide	0	1,8	0	0	1,33	2,130634	2,636001
Ran	0	1,8	0	0	1,33	2,130634	2,636001
Crying Game	0	1,8	0	1,14	1,33	1,8	2,376741
Forrest Gump	0	0	0	1,14	1,33	0	1,552063
Big Fish	1,27	0	0	1,14	0	1,838967	1,500966

Nilai $\frac{BCV}{WCV}$ dari iterasi ke-1 diatas adalah

$$\frac{BCV}{WCV} = \frac{\sqrt{(0-0)^2+(0-0)^2+(0{,}8-0)^2+(1{,}14-1{,}14)^2+(0-1{,}33)^2}}{(2{,}130633709)^2+(0)^2+(2{,}343693666)^2+(1{,}500966355)^2+(0)^2+(2{,}130633709)^2}$$

$$\overline{+(2{,}130633709)^2+(1{,}8)^2+(0)^2+(1{,}500966355)^2}$$

$$= 0{,}057788817$$

Setelah dilakukan proses iterasi, pada iterasi ke-3 dan ke-4, nilai $\frac{BCV}{WCV}$ pada iterasi ke-4 lebih kecil dari nilai $\frac{BCV}{WCV}$ pada iterasi ke-3 sehingga proses iterasi berhenti. Dan dihasilkan nilai *cluster* seperti pada Tabel 3.21 berikut ini.

Tabel 3.21 Hasil *Cluster*

movie_title	***cluster***	
	C_1	C_2
Taxi Driver	0,827587	2,415264
Rob Roy	1,328495	1,707015
Bound	2,016606	1,401749
True Romance	2,268634	0,587282
Psycho	2,042988	1,210372
Crimson Tide	0,827587	2,415264
Ran	0,827587	2,415264
Crying Game	0,827587	2,129296
Forrest Gump	1,328495	1,707015
Big Fish	2,268634	0,587282

Langkah 3

Dari hasil *cluster* yang didapat pada Tabel 4.22 dilakukan perhitungan peluang *item* terhadap *cluster* ($Pro_{j,k}$).

Contoh perhitungan $Pro_{j,k}$ antara *record* ke-1 terhadap *cluster* pertama dan kedua adalah :

$$Pro_{1,1} = 1 - (0{,}827587 / 2{,}268634) = 2{,}441047$$
$$Pro_{2,1} = 1 - (2{,}415264 / 2{,}415264) = 1$$

Hasil perhitungan peluang *item* terhadap *cluster* selengkapnya seperti pada tabel 3.22 berikut :

Tabel 3.22 Matriks *Group-rating*

movie_title	Pro	
	1	2
Taxi Driver	2,441047	1
Rob Roy	1,940139	1,708249
Bound	1,252028	2,013515
True Romance	1	2,827982
Psycho	1,225646	2,204892
Crimson Tide	2,441047	1
Ran	2,441047	1
Crying Game	2,441047	1,285968
Forrest Gump	1,940139	1,708249
Big Fish	1	2,827982

Langkah 4

Menghitung nilai *similarity* pada matriks *item-rating* (*similarity item-rating*). Sebelum dilakukan perhitungan *similarity item-rating*, dihitung terlebih dahulu rata-rata rating dari setiap *item*. Seperti ditunjukkan pada Tabel 3.23 berikut.

Tabel 3.23 Rata-rata Rating *Item*

movie_title	user_id					rata-rata rating *item*
	22	94	166	186	244	
Taxi Driver	4		2	5		3,67
Rob Roy		2		1		1,5
Bound	4	4	5			4,33
True Romance		4			2	3
Psycho	2				2	2
Crimson Tide	2	1		4		2,33
Ran	1		3	1		1,67
Crying Game		1	1	2	5	2,25
Forrest Gump	2		2		3	2,33
Big Fish						

Perhitungan nilai *similarity item-rating* menggunakan persamaan (2.5). Berikut adalah contoh perhitungan nilai *similarity item-rating* antara *item* Taxy Driver dengan Crimson Tide :

$$Similarity\ (Taxy,Crimson)_{item} = \frac{(4-3{,}67)(2-2{,}33)+(5-3{,}67)(4-2{,}33)}{\sqrt{(4-3{,}67)^2+(5-3{,}67)^2}\ \sqrt{(2-2{,}33)^2+(4-2{,}33)^2}} = 0{,}905474$$

Dari perhitungan diatas diperoleh nilai *similarity item-rating* antara *item* Taxy Driver dengan Crimson Tide adalah 0,905474.

Hasil perhitungan dari nilai *similarity item-rating* selengkapnya seperti pada Tabel 3.24 berikut :

Tabel 3.24 Hasil Perhitungan *Similarity Item-rating*

item 1	item 2	similarity item-rating
Taxi Driver	Rob Roy	-1
Taxi Driver	Bound	-0,96573
Taxi Driver	Psycho	0
Taxi Driver	Crimson Tide	0,905474
Taxi Driver	Ran	-0,90924
Taxi Driver	Crying Game	0,644871
Taxi Driver	Forrest Gump	0,556616
Rob Roy	Bound	-1
Rob Roy	True Romance	1
Rob Roy	Crimson Tide	-0,99364
Rob Roy	Ran	1
Rob Roy	Crying Game	-0,5547
Bound	True Romance	-1
Bound	Psycho	0
Bound	Crimson Tide	0,856581
Bound	Ran	0,99996
Bound	Crying Game	-0,3219
Bound	Forrest Gump	-0,3219
True Romance	Psycho	0
True Romance	Crimson Tide	-1
True Romance	Crying Game	-0,93633
True Romance	Forrest Gump	-1
Psycho	Crimson Tide	0
Psycho	Ran	0
Psycho	Crying Game	0
Psycho	Forrest Gump	0
Crimson Tide	Ran	-0,55662
Crimson Tide	Crying Game	0,457473
Crimson Tide	Forrest Gump	1

Ran	Crying Game	-0,78751
Ran	Forrest Gump	-1
Crying Game	Forrest Gump	0,999519

Langkah 5

Menghitung nilai *similarity* pada matriks *group-rating* (*similarity group-rating*). Sebelum dilakukan perhitungan pada matriks *group-rating*, dihitung terlebih dahulu rata-rata tiap *Pro*. Seperti ditunjukkan pada Tabel 3.25 berikut.

Tabel 3.25 Rata-rata *Group-rating*

movie_title	**Pro**	
	1	**2**
Taxi Driver	2,441047	1
Rob Roy	1,940139	1,708249
Bound	1,252028	2,013515
True Romance	1	2,827982
Psycho	1,225646	2,204892
Crimson Tide	2,441047	1
Ran	2,441047	1
Crying Game	2,441047	1,285968
Forrest Gump	1,940139	1,708249
Big Fish	1	2,827982
Rata-rata	1,812214	1,757684

Perhitungan nilai *similarity group-rating* menggunakan persamaan (2.6). Berikut adalah contoh perhitungan nilai *similarity group-rating* antara *item* Taxi Driver dengan Rob Roy :

$$Similarity\ (Taxi,Rob)_{group} = \frac{(2{,}441047-1{,}812214)(1{,}940139-1{,}812214)+(1-1{,}757684)(1{,}708249-1{,}757684)}{\sqrt{(2{,}441047-1{,}812214)^2+(1-1{,}757684)^2}\ \sqrt{(1{,}940139-1{,}812214)^2+(1{,}708249-1{,}757684)^2}}$$

$$= 0{,}873084141$$

Dari perhitungan diatas diperoleh nilai *similarity group-rating* antara *item* Taxi Driver dengan Rob Roy adalah 0,873084141.

Hasil perhitungan dari nilai *similarity group-rating* selengkapnya seperti pada Tabel 3.26 berikut :

Tabel 3.26 Hasil Perhitungan *Similarity Group-rating*

Item 1	Item 2	similarity group-rating
Taxi Driver	Rob Roy	0,873084141
Taxi Driver	Bound	-0,08206409
Taxi Driver	True Romance	-0,99905048
Taxi Driver	Psycho	-0,97442161
Taxi Driver	Crimson Tide	1
Taxi Driver	Ran	1
Taxi Driver	Crying Game	0,972637645
Taxi Driver	Forrest Gump	0,873084141
Taxi Driver	Big Fish	0,322950885
Rob Roy	Bound	-0,99822
Rob Roy	True Romance	-0,85101
Rob Roy	Psycho	-0,96032
Rob Roy	Crimson Tide	0,873084
Rob Roy	Ran	0,873084
Rob Roy	Crying Game	0,96247
Rob Roy	Forrest Gump	1
Rob Roy	Big Fish	-0,32256
Bound	True Romance	0,880803
Bound	Psycho	0,975239
Bound	Crimson Tide	-0,90059
Bound	Ran	-0,90059
Bound	Crying Game	-0,97694
Bound	Forrest Gump	-0,99822
Bound	Big Fish	0,880803

True Romance	Psycho	0,963706
True Romance	Crimson Tide	-0,99905
True Romance	Ran	-0,99905
True Romance	Crying Game	-0,96159
True Romance	Forrest Gump	-0,85101
True Romance	Big Fish	1
Psycho	Crimson Tide	-0,97442
Psycho	Ran	-0,97442
Psycho	Crying Game	-0,99997
Psycho	Forrest Gump	-0,96032
Psycho	Big Fish	0,963706
Crimson Tide	Ran	1
Crimson Tide	Crying Game	0,972638
Crimson Tide	Forrest Gump	0,873084
Crimson Tide	Big Fish	-0,99905
Ran	Crying Game	0,972638
Ran	Forrest Gump	0,873084
Ran	Big Fish	-0,99905
Crying Game	Forrest Gump	0,96247
Crying Game	Big Fish	-0,96159
Forrest Gump	Big Fish	-0,85101

Langkah 6

Menggabungkan (*linier combination of similarity*) nilai *similarity item*-rating dengan *group-rating*. Perhitungan ini untuk mendapatkan nilai *similarity* antar *item* dengan menggunakan persamaan (2.7).

Berikut adalah contoh perhitungan nilai *linier combination of similarity* antara *item* Taxi Driver dengan Rob Roy, dengan ditentukan nilai koefisien *c* adalah 0,2 :

Similarity (Taxy,Rob) = -1 * (1-0,2) + 0,87308414 * 0,2

= -0,62538

Dari perhitungan diatas diperoleh nilai *linier combination of similarity* antara *item* Taxi Driver dengan Rob Roy adalah -0,62538.

Hasil perhitungan dari nilai *linier combination of similarity* antar *item* selengkapnya seperti pada Tabel 3.27 berikut :

Tabel 3.27 Hasil Perhitungan *Linier Combination of Similarity*

Item 1	Item 2	linier combination of similarity
Taxi Driver	Rob Roy	-0,62538
Taxi Driver	Bound	-0,789
Taxi Driver	True Romance	-0,19981
Taxi Driver	Psycho	-0,19488
Taxi Driver	Crimson Tide	0,924379
Taxi Driver	Ran	-0,52739
Taxi Driver	Crying Game	0,710424
Taxi Driver	Forrest Gump	0,61991
Taxi Driver	Big Fish	0,26459
Rob Roy	Bound	-0,99964
Rob Roy	True Romance	0,629798
Rob Roy	Psycho	-0,19206
Rob Roy	Crimson Tide	-0,6203
Rob Roy	Ran	0,974617
Rob Roy	Crying Game	-0,25127
Rob Roy	Forrest Gump	0,2
Rob Roy	Big Fish	-0,26451
Bound	True Romance	-0,62384
Bound	Psycho	0,195048
Bound	Crimson Tide	0,505147
Bound	Ran	0,61985
Bound	Crying Game	-0,45291
Bound	Forrest Gump	-0,45716

Bound	Big Fish	0,176161
True Romance	Psycho	0,192741
True Romance	Crimson Tide	-0,99981
True Romance	Ran	-0,19981
True Romance	Crying Game	-0,94138
True Romance	Forrest Gump	-0,9702
True Romance	Big Fish	0,2
Psycho	Crimson Tide	-0,19488
Psycho	Ran	-0,19488
Psycho	Crying Game	-0,19999
Psycho	Forrest Gump	-0,19206
Psycho	Big Fish	0,192741
Crimson Tide	Ran	-0,2453
Crimson Tide	Crying Game	0,560506
Crimson Tide	Forrest Gump	0,974617
Crimson Tide	Big Fish	-0,19981
Ran	Crying Game	-0,43548
Ran	Forrest Gump	-0,62538
Ran	Big Fish	-0,19981
Crying Game	Forrest Gump	0,992109
Crying Game	Big Fish	-0,19232
Forrest Gump	Big Fish	-0,1702

Langkah 7

Menghitung prediksi rating *item*. Hasil dari perhitungan prediksi diurutkan dari nilai yang besar ke nilai yang kecil dan dijadikan sebagai rekomendasi kepada *user*, dimana rekomendasi ini disebut juga sebagai rekomendasi *Top-N*.

Berdasarkan pada Tabel 3.15, *item* yang akan dilakukan perhitungan prediksi dari *user* 94 adalah Taxi Driver, Psycho, Ran, Forrest Gump dan Big Fish.

Berikut adalah contoh perhitungan prediksi rating *user* 94 terhadap *item* Taxi Driver dan Big Fish dengan menggunakan persamaan (2.8).

$$P_{(94,Taxy)} = 3{,}67 + \frac{((2-1{,}5)*-0{,}62539)+((4-4{,}33)*-0{,}789)+((4-3)*-0{,}19981)+((1-2{,}33)*-0{,}924379)+((1-2{,}25)*0{,}710424)}{|-0{,}62539|+|-0{,}789|+|-0{,}19981|+|-0{,}924379|+|0{,}710424|}$$

$$=2{,}270672$$

Sedangkan untuk *item* Big Fish karena belum pernah *di rating* oleh *user*, maka perhitungannya menggunakan persamaan (2.9), seperti ditunjukkan berikut.

$$P_{(94,Big)} = \frac{(2*(-0{,}26451))+(4*0{,}176161)+(4*0{,}2)+(1*0{,}19981)+(1*(-0{,}19232))}{|-0{,}26451|+|0{,}176161|+|0{,}2|+|0{,}19981|+|-0{,}19232|}$$

$$= 0{,}564959$$

Hasil perhitungan prediksi berupa rekomendasi *Top-N* yang diberikan kepada *user* 94 berupa daftar *item* yang telah dirangking berdasarkan urutan prediksi nilai rating *item* dari nilai yang besar ke nilai yang kecil, selengkapnya seperti ditunjukkan pada Tabel 3.28 berikut.

Tabel 3.28 Rekomendasi *Item* Bagi *User* 94

rangking	movie_title	rating
1	Psycho	2,555577
2	Taxi Driver	2,270672
3	Ran	2,055261
4	Forrest Gump	1,424147
5	Big Fish	0,564959