

Penerapan Algoritma *Collaborative Filtering* Untuk Rekomendasi *Games Hardware*

Norma Yanti, Rustati Rahmi, Ruliah

Program Studi Sistem Informasi, STMIK Banjarbaru

Jl. A. Yani Km. 33,3 Loktabat Banjarbaru

email: rustati.rahmi@gmail.com dan noyajackson@yahoo.com

Abstrak

Recommender system merupakan sebuah program yang dapat digunakan untuk memprediksi sebuah item berdasarkan informasi yang diperoleh dari user. *Collaborative filtering* merupakan algoritma yang telah banyak digunakan dalam melakukan proses filtering. Penelitian ini menyajikan pengembangan sistem rekomendasi pada toko Pelita Abadi Game dengan menerapkan metode *collaborative filtering*. Metode rekomendasi yang digunakan didasarkan atas adanya kesamaan antara pemberian rating. Dari penelitian ini dapat disimpulkan bahwa sistem rekomendasi game dapat dibangun menggunakan metode *item-based collaborative filtering* dengan menerapkan algoritma *K-nearest neighbor*. Algoritma *K-nearest neighbor* ini dipakai untuk menentukan berapa jumlah *neighbor* (game) terdekat yang digunakan dalam proses prediksi. Hasil akurasi yang dicapai dalam pengujian dengan metode ini tergolong akurat, karena dengan menerapkan $k=40$ nilai MAE yang dihasilkan adalah 0,739.

Kata Kunci : *Collaborative Filtering*, Sistem Rekomendasi, Games.

Abstract

Recommender system is a program that can be used to predict an item based on information obtained from the user. *Collaborative filtering* is an algorithm that has been widely used in the process of filtering. This study presents the development of recommendation systems at Pelita Eternal Game store by applying methods of *collaborative filtering*. The method used was based upon the recommendations of the similarities between giving ratings. Hence the concept of *collaborative filtering* needs to be developed for the selection of items that have a lot of criteria that recommendations made have better quality and relevant to user needs.

Keywords: *Collaborative Filtering*, Recommendation Systems, Games

1. Pendahuluan

Pada penelitian sebelumnya mengenai *collaborative filtering* dengan judul *Item Collaborative Filtering* Untuk Rekomendasi Pembelian Buku Secara Online ('uyun, Fahrurrozi, & Mulyanto, 2011), meneliti mengenai pengembangan sistem rekomendasi pada toko buku online menggunakan *item collaborative filtering* dengan metode *waterfall*. Metode rekomendasi yang digunakan didasarkan atas adanya kesamaan antara pemberian rating terhadap suatu produk dengan produk yang dibeli. Selain itu ada penelitian tentang *collaborative filtering* yang ditulis (Sania, Maharani, & Prima K, 2010), peneliti membandingkan performansi *Pearson Correlation* dan *pearman Correlation* pada sistem rekomendasi yang menggunakan *Collaborative filtering*. Sedangkan pengukuran dan analisis terhadap tingkat error hasil prediksi rating dengan menggunakan MAE (*Mean Absolute Error*) berdasarkan *threshold co-rated items* dan ukuran *neighbourhood*. Teknik yang paling umum digunakan dalam memprediksi kecenderungan user dalam *recommender system*, adalah *Collaborative filtering*. *Collaborative filtering* memberikan informasi yang melibatkan kolaborasi antara beberapa pengguna, beberapa sudut pandang, dan kondisi dimana tersedia informasi *history* dari pengguna. Dasar dari algoritma CF adalah merekomendasikan *item* kepada user berdasarkan preferensi user dengan kemiripan *interest*. *User-based Collaborative filtering* mencari kesamaan di antara user untuk menghasilkan rekomendasi. Untuk mencari kesamaan antar user, dilakukan perhitungan korelasi, dengan semakin tinggi nilai, mengindikasikan user yang memiliki *interest* yang mirip. *Pearson* melakukan pengukuran korelasi secara langsung dan merupakan standar pengukuran korelasi karena pengukurannya memberikan hasil korelasi yang baik antara dua variabel, sedangkan *Spearman* menggunakan perhitungan ranking dan mampu menghindari masalah normalisasi rating, sehingga bisa diukur metode mana yang lebih baik diimplementasikan dalam

recommender system dengan dataset yang ada. Oleh karena itu, dalam penelitian ini diimplementasikan kedua metode tersebut pada *Recommender System* untuk melihat akurasi hasil prediksi yang dihasilkan. Hasil yang didapat digunakan untuk mengetahui performansi kedua korelasi menggunakan dataset yang tersedia.

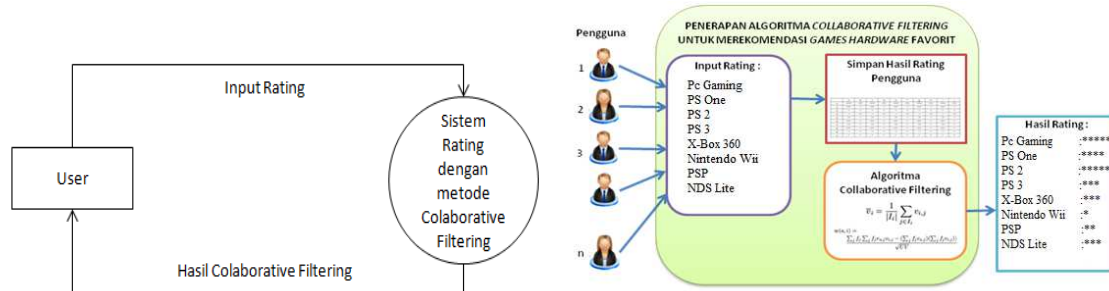
Dari tinjauan penelitian tersebut maka peneliti tertarik untuk meneliti *Games Hardware* yang ada pada Toko Abadi Pelita Game, *game hardware* terdiri dari beberapa kategori yang masuk dalam kelas terkini antara lain : *PC Gaming, Playstation One, Playstation X, Playstation 2 DVD Slim, Playstation 2 DVD, Playstation 2 Harddisk 80 GB, PSP MMC 8G, PSP Slim, Playstation 3, Playstation 3 Slim, Microsoft X-Box 360, Nintendo Wii, Playstation Portabel (PSP), Nintendo DS Lite*. Banyak sekali penggemar game yang memiliki kebiasaan memainkan salah satu atau beberapa *games hardware* tersebut. Setiap *games hardware* memiliki kelebihan dan kekurangan masing-masing dimana setiap orang ada yang mengetahui kelebihan tersebut ada juga yang tidak mengetahuinya, salah satu contoh *PC Gaming* menggunakan *DirectX 11* yang dapat menghasilkan grafis *super detail* seperti aslinya bahkan dapat di jadikan 3D dengan menggunakan *nvidia 3D Vision*, disini lain *Playstation 3* dari sony memiliki kelebihan menggunakan *Blu-Ray Disk* yang memiliki kapasitas yang besar mencapai 50 Gigabyte, sedangkan *Nintendo Wii* memiliki fitur unggulan pada *motion controller*. Melihat banyaknya *games hardware* tersebut yang memiliki banyak kelebihan dan kekurangan, tidak memungkinkan *merating games hardware* dari segi kelebihan dan kekurangan tersebut karena setiap orang memiliki persepsi dan kebiasaan yang berbeda. Oleh karena itu penulis tertarik untuk mengangkat kasus tentang penelitian ini adalah "Penerapan algoritma *collaborative filtering* untuk rekomendasi *games hardware* pada Toko Abadi Pelita Game"

2. Metodologi Penelitian

2.1. Analisa Kebutuhan

Secara umum, sistem rekomendasi bertugas membantu para user dengan cara menyediakan rekomendasi bagi para user-nya. Rekomendasi yang diberikan berupa *games-games hardware*. Pada sistem ini user dapat memberikan *rating* pada berbagai macam game yang menurutnya memiliki spesifikasi yang bagus dan layak menjadi *games hardware* terfavorit. *Rating* yang diberikan oleh masing-masing user digunakan sebagai dasar untuk menentukan rekomendasi *games hardware*.

2.2. Diagram Konteks



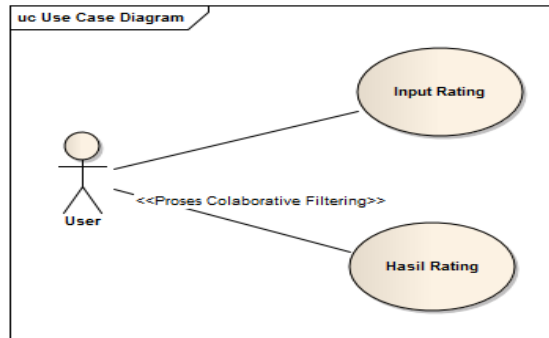
Gambar 1. Diagram Konteks dan Blok Diagram Penelitian

Diagram di atas menerangkan gambaran proses umum yang terjadi di dalam sistem. Terdapat dua komponen utama yaitu user dan sistem *rating* dengan Metode *Collaborative Filtering*. Dengan memproses hasil *collaborative filtering* pengguna menghasilkan output berupa data *games hardware* yang dirating melalui tingkat penjualan yang terjadi di toko Pelita Abadi Game. Dari gambar blok diagram penelitian di atas sudah terlihat jelas bahwa hasil dari penelitian ini nantinya berupa hasil *rating* terhadap *games hardware* yang diberikan oleh pengguna 1 s/d n yang merupakan konsumen dari toko Pelita Abadi Game.

2.3. Use Case Diagram

Pada use case diagram berikut akan dideskripsikan bagaimana system terlihat dimata pengguna. Sasaran pemodelan use case diantaranya adalah mendefinisikan kebutuhan

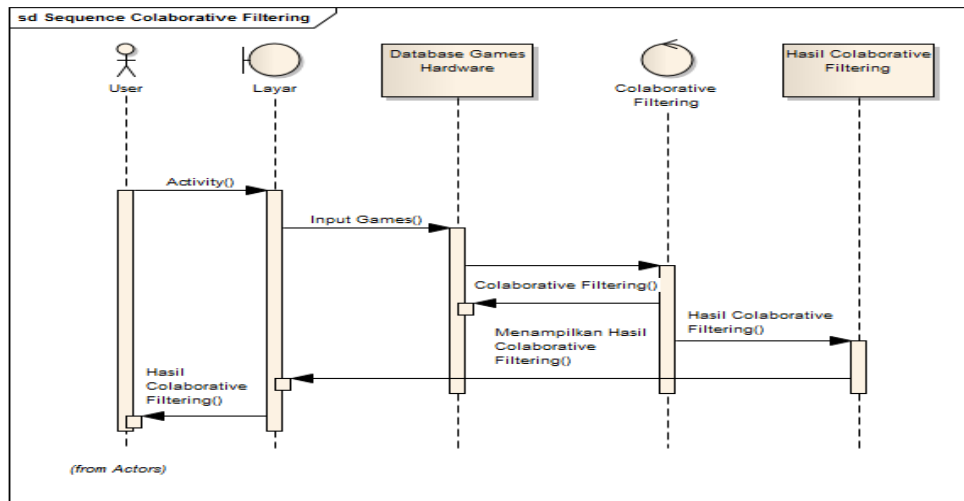
fungsiional dan operasional system dengan mendefinisikan scenario penggunaan yang disepakati antara pemakai dan pengembang.



Gambar 2. Use Case Diagram

2.4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas waktu dan obyek-obyek yang terkait.

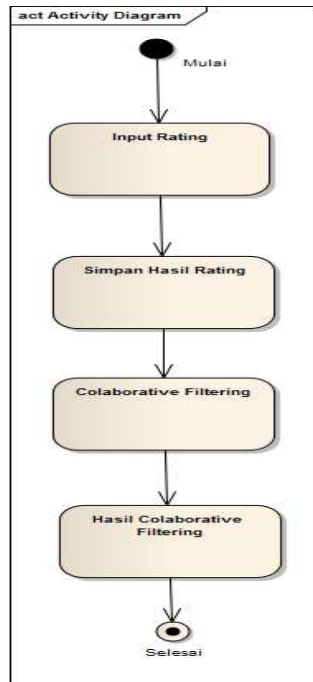


Gambar 3. Sequence Diagram

Disini user akan melihat tampilan menu pada layar yang kemudian akan dilanjutkan dengan memasukkan kata pencarian yang ingin dicari dengan menggunakan keyboard, kemudian sistem akan melakukan verifikasi kata pencarian pada database setelah itu sistem akan menampilkan hasil pencarian. Staff/admin akan melihat tampilan menu pada layar dan dengan melakukan login staff/admin dapat melakukan pemrosesan data untuk menginput, mengedit, menghapus dan menambah data, dan jika staff/admin telah selesai melakukan pemrosesan tersebut staff/admin dapat logout untuk keluar dari pemrosesan data dan tampilan pada layar akan kembali ke tampilan menu

2.5. Activity Diagram

Dalam *activity diagram* tiap aktivitas direpresentasikan dengan *rounded rectangle* yang dihubungkan dengan anak panah untuk menggambarkan transisi dari satu aktivitas ke aktivitas lain. *Activity diagram* mulai dari *initial state* dan diakhiri dengan *final state*.



Gambar 4. Activity Diagram

2.6. Desain Form Collaorative Filtering

Desain Form Collaborative Filtering merupakan *form* yang akan ditampilkan saat pertama kali membuka program. Rancangan *form Collaborative Filtering* adalah sebagai berikut:

Collaborative Filtering										
ID	<input type="text"/>									
NAMA	<input type="text"/>									
Pc Gaming	PS X	PS2	PS3	X-Box 360	N. Wii	PSP	NDS Lite			
R1 *****	R2 *****	R3 *****	R4 *****	R5 *****	R6 *****	R7 *****	R8 *****			
<input type="button" value="BARU"/> <input type="button" value="BATAL"/> <input type="button" value="SIMPAN"/> <input type="button" value="HAPUS"/> <input type="button" value="KELUAR"/>	ID	NAMA	Pc Gaming	PS X	PS 2	PS 3	X-Box	N. Wii	PSP	NDS Lite

Gambar 5. Desain Form Collaborative Filtering

2.7. Teknik Analisis Data

Teknik analisis pada penelitian ini menggunakan *collaborative filtering*. Dimana *collaborative filtering* merupakan sistem yang mampu memberi rekomendasi kepada user aktif mengenai suatu item atau produk tertentu yang mungkin menarik baginya. Dalam usaha menyusun rekomendasinya tersebut, *collaborative filtering* menggunakan kolaborasi informasi user lain yang memiliki selera hampir sama dengan user aktif. Setiap profil dari user dalam sistem, mendapatkan nilai *rating* terhadap item-item yang ada. Nilai *rating* dari user bisa merupakan angka dengan kondisi tertentu, misalnya angka 5 menunjukkan sangat suka dan akan terus turun ke angka 1 menunjukkan sangat tidak suka, jadi diantara angka-angka tersebut bisa dimasukkan.

Tabel 1. Tabel Skenario Pembangkit Rekomendasi

User	Games Hardware				
	A	B	C	D	E
0001	4	0	0	4	0
0002	5	3	3	0	0
0003	4	5	0	0	0
0004	2	5	0	3	0
0005	0	0	0	0	0

Pemberian nilai rekomendasi terdiri atas beberapa langkah, yaitu: (1) pengecekan pelanggan, jika diketahui pelanggan yang dengan user 0001 login ke sistem, maka sistem akan mengecek siapa saja pelanggan dengan riwayat pembelian atau peratingan yang sama dengan user 0001. Jika sudah diketahui pelanggan-pelanggan tersebut maka sistem akan menghitung jumlah pelanggan yang sama dengan user 0001 (C). Maka pembangkitan rekomendasi akan dihitung dengan formula:

$$R = \frac{\sum B * rating * 100}{\frac{5 * 1}{C}} \dots \dots \dots (1)$$

Keterangan:

- B = Nilai pembelian *games hardware* (1 jika membeli & 0 jika tidak membeli).
- rating = Jumlah rating yang diberikan oleh pelanggan ke *games hardware* yang disukai
- 100 = Nilai maksimum 100, nilai rekomendasi tertinggi adalah 100
- 5 = Nilai maksimum rating yang diberikan oleh pelanggan
- C = Jumlah pengguna yang mempunyai pola pembelian yang sama dengan pelanggan yang melakukan login.

Jika pelanggan dengan kode 0001 login pada sistem maka penghitungan algoritma pembangkit rekomendasi adalah sebagai berikut :

Rating User 0001=(((1*4*100)/5.1)+0+((1*4*100)/5.1)+0+0)/3=53,33
Rating User 0002=(((1*5*100)/5.1)+((1*3*100)/5.1)+((1*3*100)/5.1)+0+0)/3= 73,33
Rating User 0003=(((1*4*100)/5.1)+((1*5*100)/5.1+0+0+0))/3= 60
Rating User 0004=(((1*2*100)/5.1+0))+(1*5*100)/5.1)+0+((1*3*100)/5.1+0))/3 = 66.67

3. Hasil dan Analisis

Hasil dari penelitian ini dapat mengetahui *rating* tertinggi dari *games hardware* yang ada dalam rekomendasi sistem yang dibuat berdasarkan *rating* pengguna, dan dapat juga dijadikan rekomendasi kepada calon pembeli apabila ingin membeli *games hardware* terfavorit berdasarkan *rating* dari pengguna sebelumnya.

Seluruh data game yang digunakan dalam penelitian ini disimpan pada tabel game, yang terdiri dari id, nama dan *games hardware*. Tabel ini berisi nilai *rating* dari setiap *user* untuk setiap game yang telah dirating.

Pada tabel 2 berisi data *rating* dari *user* i terhadap game j yang berjumlah sebanyak 23 *instance* dengan rentang nominal *rating* antara 1-5 dengan iterasi 1, nilai *rating* 1 berarti game dinilai sangat buruk dan *rating* 5 game dinilai sangat baik. Dalam tabel itu juga berisi nilai rata-rata *rating* dari setiap *user*.

Tabel 2. Tabel Rating User

ID	Games Hardware					Rata- rata rating user
	A	B	C	D	E	
0019	5	3	2	5	-	3,75
0032	3	4	5	5	4	4,2
0036	-	4	3	-	-	3,5
0037	3	4	-	-	-	3,5
0044	4	-	3	-	-	3,5
0046	5	-	5	5	5	5
0047	4	2	4	3	-	3,25

Dalam implementasi Tabel Rating User disimpan dalam *array* seperti terlihat pada tabel 3 berikut ini, dan hasil dari *query* tersebut yaitu berupa *view table* yang berisi nilai rata-rata *rating* dari setiap *user*.

Tabel 3. Tabel Data Rating

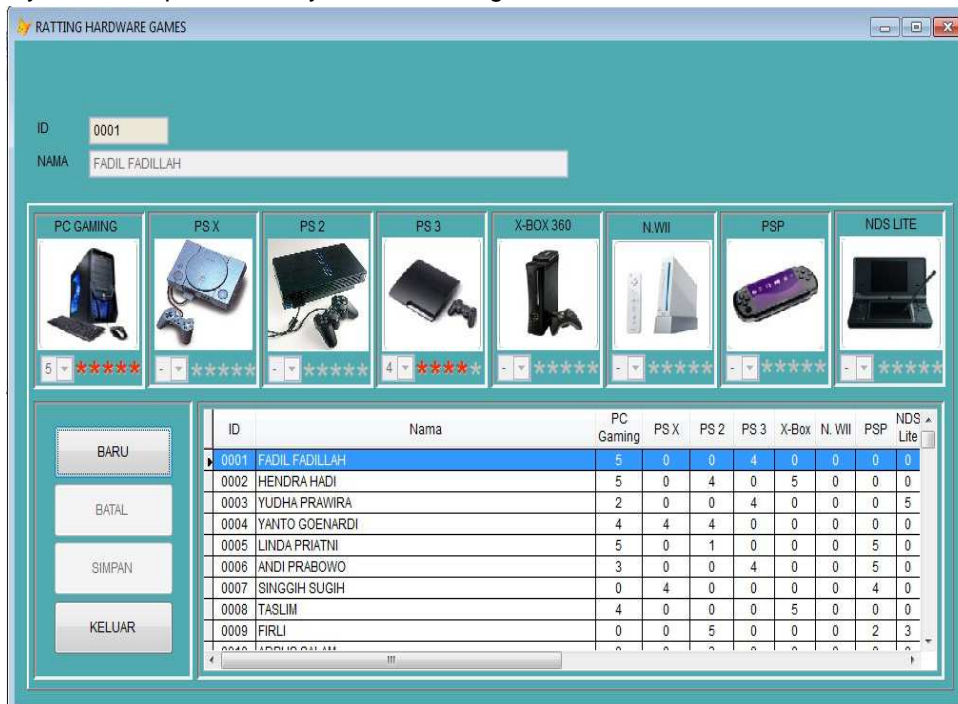
ID	Games Hardware	Rating
0019	A	5
0019	B	3
0019	C	2
0019	D	5
0032	A	3
0032	B	4
0032	C	5
0032	D	5
0032	E	4
0036	B	4
0036	C	3
0037	A	3
0037	B	4
0044	A	4
0044	C	3
0046	A	5
0046	C	5
0046	D	5
0046	E	5
0047	A	4
0047	B	2
0047	C	4
0047	D	3

Tabel 4. Tabel Rata-rata Rating User

ID	Rating
0019	3,75
0032	4,2
0036	3,5
0037	3,5
0044	3,5
0046	5
0047	3,25

3.1. Tampilan aplikasi

Tampilan aplikasi dibuat berdasarkan desain yang sudah dirancang sebelumnya, kemudian di implimentasikan dalam *form – form* dalam proses pembuatan *source code* dimasukkan metode algoritma *collaborative filtering*. *Form Collaborative Filtering* merupakan *form* yang pertama kali tampil. Pada *form* ini terdapat semua proses yang dilakukan selanjutnya. Hasil implementasinya adalah sebagai berikut :



Gambar 7. Form Collaborative Filter

3.2. Pembuatan Nilai Kemiripan Antar Item

Tahap ini adalah tahapan untuk membuat nilai kemiripan diantara game yang telah *dirating* oleh *user*. Untuk membuat nilai kemiripan, digunakan persamaan *adjusted-cosine* (2) yaitu :

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}} \dots (2)$$

$sim(i, j)$ = Nilai kemiripan antara item i dan item j

$u \in U$ = Himpunan user u yang merating item i dan item j

$R_{u,i}$ = Rating user u pada item i

$R_{u,j}$ = Rating user u pada item j

\bar{R}_u = Nilai rata-rata rating user u

Berdasarkan data rating yang ada pada tabel 1, maka contoh implementasi dari persamaan 2. adalah sebagai berikut:

Menghitung kemiripan antara game a dengan game b

Himpunan user yang merating game a dan game b : $u = \{19,32,37,47\}$

Kemudian hitung persamaan 2 :

$$sim(a, b) = \frac{(5 - 3,75)(3 - 3,75) + \dots + (4 - 3,25)(2 - 3,25)}{\sqrt{(5 - 3,75)^2 + \dots + (4 - 3,75)^2} \sqrt{(3 - 3,75)^2 + \dots + (3 - 3,25)^2}}$$

$$= -0,62101989841827$$

Dari perhitungan di atas diperoleh bahwa nilai kemiripan antara game a dengan game b adalah -0,621.

Selanjutnya, seluruh data kemiripan antar game ini disajikan dalam format tabel seperti dibawah ini:

Tabel 5. Tabel nilai kemiripan antar game

Game 1	Game 2	Nilai Kemiripan
A	B	-0,62101989841827
A	C	-0,68308840694963
A	D	0,14604417053005
A	E	1
B	C	-0,010599383345815
B	D	-0,35449213430496
B	E	1
C	D	-0,55822059870772
C	E	-1
D	E	-1

Dalam sistem yang dikembangkan, nilai kemiripan ini tidak disimpan dalam database, tetapi dikalkulasi ketika akan melakukan rekomendasi dan hasilnya disimpan dalam suatu array. Hal tersebut dilakukan karena nilai kemiripan cenderung terus berubah ketika ada rating baru yang masuk.

Efek yang diakibatkan oleh hal tersebut adalah cost komputasi dan waktu akan meningkat dalam setiap melakukan proses prediksi, tetapi dilain sisi prediksi sistem akan lebih akurat karena data kemiripan yang digunakan dalam dalam proses prediksi dihasilkan secara langsung saat itu juga.

3.3. Penghitungan Prediksi & Eksperimen Nearest Neighbor

Sesudah nilai kemiripan antar game didapatkan, maka tahap selanjutnya adalah membuat prediksi *rating* terhadap game yang belum *dirating* oleh *user*. Untuk menghitung prediksi *rating*, digunakan persamaan 3 yaitu persamaan *weighted sum* dengan menerapkan *nearest neighbor* dengan menentukan jumlah *neighbor* yang digunakan dalam proses penghitung prediksi.

Berikut persamaan *weighted sum* :

$$P(u, j) = \frac{\sum_{i \in I_u^K(j)} (R_{u,i} * S_{i,j})}{\sum_{i \in I_u^K(j)} |S_{i,j}|} \dots \dots \dots (3)$$

Keterangan :

- $P(u, j)$ = Prediksi untuk user u pada item j
- $i \in I_u^K(j)$ = Himpunan K item yang mirip dengan item j
- $R_{u,i}$ = Rating user u pada item i
- $S_{i,j}$ = Nilai kemiripan antara item i dan item j

Dalam proses memprediksi suatu *rating* game terhadap seorang *user*, sebelumnya ditentukan dahulu berapa jumlah K yang akan digunakan. K dalam hal ini merupakan sejumlah game yang telah dirating oleh *user* dan mempunyai nilai kemiripan paling tinggi dengan game yang akan diprediksi ratingnya.

Contoh implementasi dari persamaan 3 dalam menghitung nilai prediksi *rating* untuk setiap item yang belum dirating oleh *user* berdasarkan tabel data *rating* 3 adalah sebagai berikut :

- Prediksi *rating* untuk *user* 0046 terhadap game b
 Pada perhitungan ini penulis berasumsi *nearest neighbor* yang digunakan adalah 3 game.
 Berdasarkan tabel nilai kemiripan 5, himpunan 3 game yang paling mirip dengan game b (diurutkan berdasarkan nilai kemiripannya) :
 $i = \{E, C, D\}$
 Kemudian hitung menurut persamaan 2.3

$$P(0046, b) = \frac{(5 * 1) + (5 * -0,010) + (5 * -0,354)}{|1| + |-0,010| + |-0,354|}$$

$$= \frac{3,18}{1,368} = 2,331$$

Maka, prediksi *rating* untuk *user* 0046 terhadap game b adalah 2,331
 Berikut daftar tabel prediksi *rating* untuk seluruh game yang belum dirating oleh *user*.

Tabel 6. Hasil Prediksi game yang belum dirating menggunakan $K=3$

ID	Game	Prediksi
0019	E	2,795
0036	A	-4
0036	D	-3,388
0036	E	-3
0037	C	-3,014
0037	D	-2,584
0037	E	-2,584
0044	B	2,331
0044	D	-1,552
0044	E	1,492
0046	B	5
0047	E	0,666

Pada penelitian ini, eksperimen dilakukan dalam menentukan jumlah *nearest neighbor* yang digunakan sehingga mendapat prediksi yang nantinya menghasilkan nilai MAE yang paling rendah.

3.4. Menghitung Nilai MAE (Mean Absolute Error)

Proses ini dilakukan untuk menghitung keakuratan sistem dalam melakukan prediksi *rating*. Persamaan yang digunakan adalah *mean absolute error* (persamaan 4) yaitu:

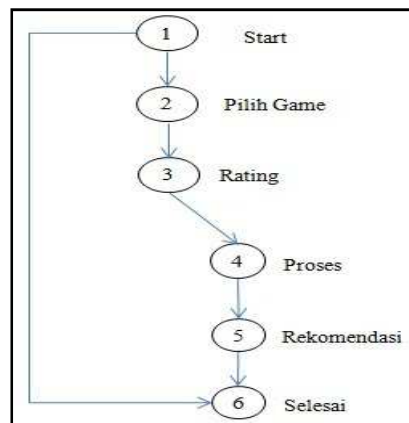
$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \dots \dots \dots (4)$$

Keterangan :
 MAE = Nilai rata-rata kesalahan hitung
 N = Jumlah item yang dihitung
 p_i = Nilai prediksi item ke-i
 q_i = Nilai rating sebenarnya item ke-i

Menghitung nilai absolut selisih *rating* yang ada pada tabel hasil prediksi (tabel 6) dengan *rating* sebenarnya dibagi dengan jumlah data *rating*. Sebagai tambahan, penelitian ini mengasumsikan bahwa hanya game yang mempunyai prediksi *rating* di atas 0 yang akan dihitung dalam proses ini. Hal tersebut dikarenakan suatu game yang mempunyai nilai prediksi *rating* 0 terhadap seorang *user*, tidak memiliki nilai kemiripan dengan game yang telah dirating oleh *user* itu sebelumnya. Sehingga sistem menganggap game tersebut sebagai *unpredicted objek* dan keluar dari proses MAE.

3.5. Pengujian White box dan Black box

Pada pengujian *white box testing* digunakan untuk mengetahui cara kerja suatu perangkat lunak secara internal. Pengujian dilakukan untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang telah dirancang. Pengujian ini pun tidak dilakukan terhadap keseluruhan program tetapi hanya dilakukan pada saat sampel pengujian diproses mulai, Alur system keseluruhan dapat dilihat pada gambar berikut ini :



Gambar 8. Flowgraph Collaborative Filtering

Kemudian dilakukan Pengujian *black box* digunakan untuk menguji fungsi-fungsi khusus dari aplikasi yang dikembangkan. Test input dan output untuk fungsi yang ada tanpa memperhatikan prosesnya. Pada pengujian ini kebenaran aplikasi yang diuji dilihat berdasarkan keluaran tersebut. Pengujian disini dilakukan hanya pada menu yang ada pada aplikasi *collaborative filtering* sebagai berikut, dapat dilihat pada tabel 7 dan 8.

Tabel 7. Black Box Menu Pelatihan

No.	Tes Case	Hasil Harapan	Hasil Keluaran	Hasil Uji
1.	User mengklik Tombol Proses	Muncul form input data User	Tampilan form input data User	Sesuai

Tabel 8. Black Box Menu Pengujian

No.	Tes Case	Hasil Harapan	Hasil Keluaran	Hasil Uji
1.	User mengklik Tombol Baru	Muncul form input data User	Tampilan form input data User	Sesuai
2.	User mengklik Tombol Simpan	Menampilkan hasil rekomendasi <i>collaborative filtering</i>	Hasil identifikasi dari rating yang telah diisi.	Cukup sesuai

4. Kesimpulan

Dari penelitian di atas, dapat disimpulkan bahwa sistem rekomendasi game dapat dibangun menggunakan metode *item-based collaborative filtering* dengan menerapkan algoritma *K-nearest neighbor*. Algoritma *K-nearest neighbor* ini dipakai untuk menentukan berapa jumlah *neighbor* (game) terdekat yang digunakan dalam proses prediksi. Hasil akurasi yang dicapai dalam pengujian dengan metode ini tergolong akurat, karena dengan menerapkan $k=40$ nilai MAE yang dihasilkan adalah 0,739.

Referensi

- [1] Islahudin, H. *Sistem Rekomendasi Bahan Ajar untuk Elearning*. Seminar Nasional Aplikasi Teknologi Informasi 2006 (SNATI 2006) - A125-A130. Yogyakarta. 2006
- [2] Sania, R., Maharani, W., & Prima K, A. Analisis Perbandingan Metode Pearson dan Spearman Correlation pada Recommender System. Konferensi Nasional Sistem dan Informatika 2010, 99-105. 2010
- [3] Santoso, L. W., Noertjahyana, A., & Tandarto, R. *Aplikasi E-Commerce dengan Intelligent Agen Software pada Optik Indo*. SNASTI 2010. Surabaya. 2010: 1-8
- [4] Uyun, S., Fahrurrozi, I., & Mulyanto, A. Item Collaborative Filtering untuk Rekomendasi Pembelian Buku secara Online. *JUSI*. 2011; 1: 63-70.
- [5] Winarko, E., Wiranto. *Konsep Multicriteria Collaborative Filtering untuk Perbaikan Rekomendasi*. Prosiding Seminar Nasional Informatika (semnasIF 2010) UPN Yogyakarta. Yogyakarta. 2010: 95-101