

Personalisasi Web *E-Commerce* Menggunakan *Recommender System* dengan Metode *Item-Based Collaborative Filtering*

Farid Masruri

Wayan Firdaus Mahmudy, (wayanfm@brawijaya.ac.id)

ABSTRAK

Personalisasi web merupakan suatu proses untuk menyesuaikan halaman web menurut selera, atau kebiasaan masing-masing *user*. Personalisasi dibutuhkan karena saat ini sedang terjadi luapan informasi di internet yang membuat *user* mengalami kesulitan dalam mencari informasi yang berguna dan sedang dibutuhkan. Sebuah cara untuk melakukan personalisasi dalam web adalah melalui *recommender system* (RSs). Tujuan utama RSs adalah memprediksi item atau suatu informasi yang kemungkinan menarik minat atau berguna bagi *user*. Dalam web *e-commerce*, RSs berbasis *collaborative filtering* (CF) dapat digunakan untuk membantu *user* dalam memilih item yang sesuai berdasarkan rating yang diberikan *user* lain. Makalah ini menjelaskan pembuatan web *e-commerce* yang mampu memberi rekomendasi secara otomatis kepada *user*. Metode yang digunakan adalah metode *item-based CF* dengan algoritma Slope One yang membuat model korelasi antar item dengan membandingkan rating antara item satu dengan yang lain dan menghitung rata-rata selisih ratingnya. Semakin kecil rata-rata selisih rating antara 2 item, maka semakin dekat korelasinya. Berdasarkan model korelasi antar item, diprediksi berapa rating yang diberikan *user* terhadap sebuah item. Rekomendasi ditampilkan kepada *user* dalam bentuk daftar item hasil prediksi yang telah diurutkan dari rating hasil prediksi yang terbesar. Dalam aplikasi yang dibuat, rekomendasi yang diberikan kepada *user* ada 2 macam, yaitu rekomendasi *personalized* bagi *user* yang terdaftar dan rekomendasi *non-personalized* bagi *user* yang tidak terdaftar atau pengunjung.

Kata kunci: personalisasi web, *recommender system*, *collaborative filtering*, *item-based*

1. PENDAHULUAN

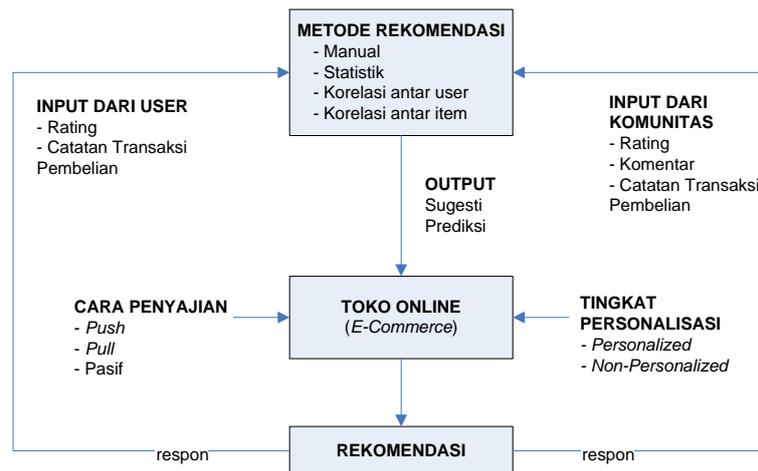
Beragam informasi tersedia di internet dan dapat diakses secara luas oleh penggunanya. Banyaknya data baik berupa teks, gambar, maupun video yang ada menyebabkan terjadinya luapan informasi yang membuat pengguna/*user* internet seringkali mengalami kesulitan dalam mencari informasi yang benar-benar sedang dicari atau dibutuhkan. Dalam perdagangan toko online atau *e-commerce*, seperti penjualan CD atau DVD film misalnya, kesulitan yang dihadapi *user* sebagai calon konsumen adalah ketika mencari sebuah produk film yang tepat dan sesuai dengan selera atau kesukaannya. Kebanyakan web *e-commerce* yang ada saat ini mempunyai struktur yang besar dan kompleks sehingga membuat informasi yang dibutuhkan oleh calon konsumen justru tidak tersampaikan. Oleh karena itu, personalisasi informasi di internet menjadi hal yang penting saat ini. Bagi sebuah web *e-commerce*, dibutuhkan sebuah sistem yang mampu beradaptasi dengan profil masing-masing konsumen dan dapat memberi saran atau sugesti kepada konsumen tentang sebuah produk yang cocok bagi konsumen. Dalam kajian web mining, sistem personalisasi web semacam ini disebut dengan *recommender system* (RSs).

RSs bekerja untuk membantu *user* dalam memilih item yang akan dibeli dengan cara memberi informasi atau sugesti berdasarkan tingkat ketertarikan atau selera masing-masing *user* [2]. Ketertarikan atau selera *user* dapat diketahui dari beberapa hal, misalnya dari cara *user* mengakses halaman web, dari rating yang diberikan *user* terhadap suatu item produk, atau dari data transaksi *user* pada waktu lampau. Sedangkan informasi yang bisa diberikan kepada *user* dapat berupa ulasan mendetail tentang item, angka penjualan, atau dengan menunjukkan nilai rating dari *user-user* yang lain (komunitas lain). Contoh web *e-commerce* populer yang telah cukup sukses mengimplementasikan RSs adalah Amazon [1] dan CDNow [3].

2. RECOMMENDER SYSTEM

Recommender system (RSs) adalah salah satu bentuk *personalized information system* yang digunakan dalam web *e-commerce* untuk menawarkan item kepada *user* dan memberi informasi yang dapat membantu

user dalam memilih atau membeli item [8]. Gambar 1 berikut ini menunjukkan taksonomi *recommender system* [5].



Gambar 1. Taksonomi *recommender system*

Pada Gambar 1 terdapat tiga 3 komponen utama dalam sebuah *recommender system* [8], yaitu:

1. Input/ Output

Input yang dianalisa RSs didapat baik secara eksplisit maupun implisit dari *user* kemudian dikombinasikan dengan input dari *user-user* lain/ komunitas. Input yang didapat secara eksplisit, misalnya dengan cara meminta *user* untuk memberi rating terhadap suatu item. Sedangkan secara implisit, misalnya dari data transaksi pembelian item oleh *user* pada waktu lampau, atau bisa juga dengan memonitor item-item mana saja yang telah dilihat oleh *user*. Output yang dihasilkan oleh RSs dapat berupa sugesti (merekomendasikan sebuah item secara khusus) atau prediksi (bisa berupa prediksi per item atau beberapa item sekaligus dalam bentuk *list/* daftar).

2. Metode Rekomendasi

Metode yang digunakan dalam memberikan rekomendasi dapat dilakukan dengan beberapa cara, antara lain secara manual, melalui pendekatan statistik, dengan berdasarkan korelasi antar *user* (*user-to-user*), atau juga dengan berdasarkan korelasi antar item (*item-to-item*).

3. Desain Rekomendasi

Desain rekomendasi terkait pada 2 hal, yakni bagaimana rekomendasi disajikan dan bagaimana sifat rekomendasi atau tingkat personalisasinya. Ada 3 cara untuk menyajikan rekomendasi kepada *user*, yaitu:

- *Push* : bentuk aktif pemberian rekomendasi, seperti mengirimkan kepada *user* melalui email.
- *Pull* : rekomendasi tidak ditampilkan jika *user* tidak meminta
- Pasif : menampilkan item lain yang berhubungan item yang sedang dilihat/ diakses *user* pada saat itu.

Tingkat personalisasi rekomendasi yang diberikan kepada *user* ada 2 macam, yaitu:

- *Personalized* : rekomendasi yang diberikan kepada *user* tidak sama antara satu dengan yang lain, bergantung pada masing-masing profil *user*.
- *Non-personalized* : bentuk rekomendasi ini tidak melihat profil masing-masing *user*, dengan kata lain rekomendasi bersifat umum sehingga dapat diberikan bagi semua pengunjung/ visitor.

Berbagai teknik data mining seperti *association rule* dan *market basket analysis* [9], *nearest neighbor* [12], hingga *clustering* telah digunakan dalam membangun sebuah RSs. Namun, secara garis besar teknik yang digunakan dalam RSs ada 2 macam, yakni *content-based filtering* dan *collaborative filtering* [8]. *Content-based filtering* bekerja dengan mencari item yang mempunyai korelasi dengan item yang disukai *user* berdasarkan *content/* informasi tekstual tiap item. Sedangkan *collaborative filtering* (CF) merekomendasikan item kepada seorang *user* berdasarkan rating yang diberikannya terhadap item. CF lebih

banyak digunakan karena dalam beberapa domain, seperti musik dan film, sulit dilakukan ekstraksi *content* tiap item yang merupakan langkah utama dari *content-based filtering* [9].

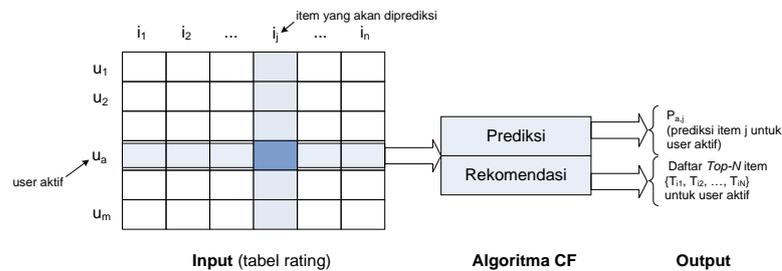
Penelitian tentang RSs berbasis CF telah banyak dilakukan. RSs berbasis CF yang ada pertama kali menggunakan metode *nearest-neighbor* [12]. Metode ini dikenal pula dengan nama *memory-based* atau *user-based* karena mencari *user* yang paling mirip dengan *user* target dalam hal merating item, lalu merekomendasikan item-item yang disukai *user* terdekat kepada *user* target. Kelemahan metode *memory-based* adalah masalah skalabilitas seiring dengan makin banyaknya jumlah *user* dan jumlah item. Metode *model-based* atau *item-based* muncul sebagai solusi metode *user-based* CF. Berbeda dengan metode *user-based* yang menghitung kemiripan antar *user* secara *online*, metode *item-based* melakukan pembuatan model korelasi antar item terlebih dahulu secara *offline* untuk kemudian digunakan dalam membuat rekomendasi secara *online* sehingga rekomendasi dapat diberikan secara *real-time* [7].

Pada makalah ini metode *item-based* yang digunakan adalah algoritma *Slope One* yang mencari korelasi antar item berdasarkan “beda popularitas” [4]. Dengan algoritma ini, akan dibuat 2 macam bentuk rekomendasi kepada *user*, rekomendasi *personalized* dan *non-personalized*.

2.1. Collaborative Filtering

Collaborative filtering (CF) adalah suatu metode dalam membuat prediksi otomatis untuk memperkirakan ketertarikan atau selera seorang *user* terhadap suatu item dengan cara mengumpulkan informasi dari *user-user* yang lain yang direpresentasikan dalam bentuk nilai rating [7]. Secara umum, ada 2 proses yang dilakukan dalam CF, yaitu:

1. Mencari *user* lain yang mempunyai kemiripan pola rating dengan *user* target (*user* yang akan diberikan prediksi).
2. Menggunakan nilai rating dari *user* lain yang didapat dari langkah 1 di atas untuk menghitung prediksi bagi *user* aktif.



Gambar 2. Proses *collaborative filtering* [7]

Sebagaimana ditunjukkan pada Gambar 2, algoritma CF menghasilkan 2 bentuk output, yaitu [7]:

- **Prediksi ($P_{a,j}$)** adalah suatu nilai yang menyatakan prediksi besarnya rating item j yang mungkin didapat dari *user* aktif, dimana item j belum pernah dirating oleh *user* aktif.
- **Rekomendasi** adalah sebuah daftar yang berisi N item yang mempunyai kemungkinan terbesar untuk disukai oleh *user* aktif, dimana item-item tersebut belum pernah dirating oleh *user* aktif. Bentuk ini disebut juga sebagai rekomendasi Top- N .

Ada perbedaan antara rekomendasi (dengan seorang *user*, disarankan item-item yang mungkin menarik bagi *user* tersebut) dan prediksi (dengan sebuah item, dilakukan prediksi berapa rating yang mungkin diberikan *user* pada item tersebut). Namun, pada dasarnya untuk memberi rekomendasi kepada *user* diperlukan perhitungan prediksi tingkat ketertarikan *user* terhadap item-item.

Secara umum, berbagai algoritma CF yang ada saat ini dapat dikelompokkan menjadi 2 kategori [7], yaitu:

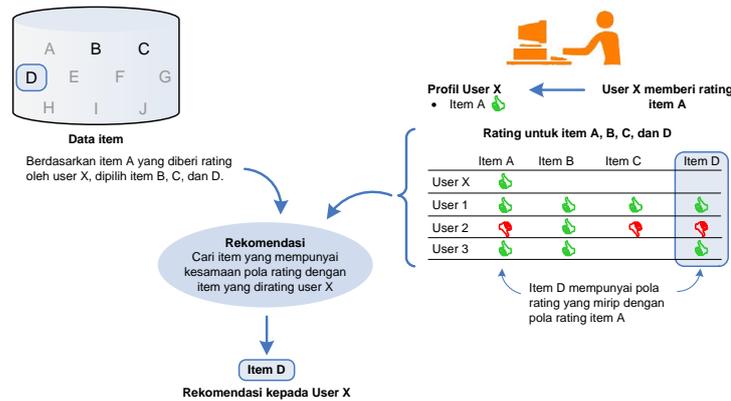
1. *Memory-based* CF

Algoritma *memory-based* mempergunakan seluruh data rating dalam membuat prediksi. Metode ini memakai teknik statistik dalam mencari sekumpulan *user*, disebut dengan *nearest-neighbor*, yang

mempunyai kemiripan dengan *user* target. Berdasarkan kemiripan antar *user* tersebut, dibuat prediksi dan rekomendasi *Top-N* untuk *user* target. Metode ini sering disebut juga dengan *nearest-neighbor* atau *user-based collaborative filtering*.

2. *Model-based CF*

Berbeda dengan *memory-based* yang mencari hubungan antar *user*, *model-based* mencari hubungan antar item berdasarkan tabel rating untuk membuat rekomendasi. Sehingga metode ini disebut juga dengan *item-based*. Sebelum menghasilkan rekomendasi, algoritma *model-based* membuat sebuah model korelasi antar item terlebih dahulu untuk mengetahui hubungan antar item berdasarkan nilai rating yang didapat. Proses pembuatan model dapat dilakukan dengan menggunakan berbagai teknik, seperti *association rule*, *classification*, atau *clustering*. Pembuatan model korelasi antar item dilakukan secara *offline*. Berdasarkan model yang sudah dibentuk, maka perhitungan prediksi atau pemberian rekomendasi dapat dilakukan secara *online*. Skema *model-based* atau *item-based* dapat dilihat pada Gambar 4.

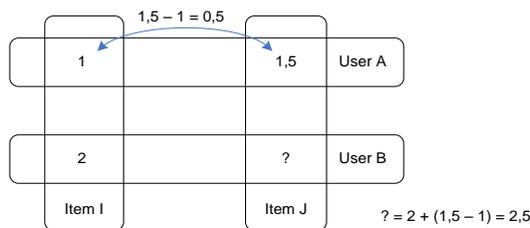


Gambar 3. Skema *model-based/ item-based CF*

2.2. Algoritma Slope One

Prinsip utama *algoritma Slope One* adalah bekerja berdasarkan atas “perbedaan popularitas” antara item yang satu dengan yang lainnya dengan cara menghitung selisih rating antara 2 item [4]. Dari selisih rating yang didapat kemudian digunakan untuk memprediksi berapa besarnya nilai rating yang diberikan terhadap suatu item. Untuk dapat melakukan prediksi, algoritma Slope One membutuhkan 2 input, yaitu rating dari *user* target dan item mana yang akan diprediksi. Oleh karena itu, jika *user* tidak memberi satu rating pun terhadap item, maka prediksi tidak dapat dilakukan.

Dimisalkan terdapat 2 *user*, *user A* dan *user B*, dan 2 item, item I dan J. Sebagaimana diperlihatkan pada Gambar 5, *user A* memberi nilai rating 1 untuk item I dan 1,5 untuk item J, sedangkan *user B* memberi nilai rating 2 untuk item I saja. Dari sini dapat diketahui bahwa selisih rating antara item I dan J sebesar $1,5 - 1 = 0,5$. Dengan selisih ini, maka dapat diprediksi bahwa *user B* akan memberi rating terhadap item J sebesar $2 + 0,5 = 2,5$.



Gambar 4. Skema algoritma *Slope One* [4]

Secara matematis, pendekatan algoritma Slope One dapat dijelaskan sebagai berikut: Diberikan rating dari seorang *user* yang disebut sebagai *evaluasi*, dinotasikan dengan u , dimana u_i adalah rating yang

diberikan *user* u terhadap item i . Set atau kumpulan item yang termasuk dalam u adalah $S(u)$, sedangkan \mathcal{X} adalah kumpulan evaluasi dalam training set. Banyaknya elemen dalam S dinotasikan dengan $card(S)$. Rata-rata rating dalam sebuah evaluasi u dinotasikan dengan u , sedangkan $S_i(\mathcal{X})$ adalah set/ kumpulan semua evaluasi $u \in \mathcal{X}$ yang mencakup item i di dalamnya ($i \in S(u)$). Maka, perhitungan korelasi antar item untuk mengetahui seberapa dekat hubungan antara 2 item berdasarkan selisih rating dan banyaknya rating dirumuskan dengan:

$$dev_{j,i} = \sum_{u \in S_{j,i}(\mathcal{X})} \frac{u_j - u_i}{card(S_{j,i}(\mathcal{X}))} \dots\dots\dots(1)$$

$dev_{j,i}$ = rata-rata selisih rating item i terhadap item j

u_j = rating item j

u_i = rating item i

\mathcal{X} = training set

$S_{j,i}(\mathcal{X})$ = kumpulan semua evaluasi $u \in \mathcal{X}$ yang mencakup item i dan j di dalamnya ($i, j \in S(u)$)

$card(S_{j,i}(\mathcal{X}))$ = banyaknya elemen dalam $S_{j,i}(\mathcal{X})$

Apabila telah diketahui rata-rata selisih rating item i terhadap item j , maka dapat dilakukan perhitungan prediksi rating untuk item j yang dirumuskan dengan:

$$P^{S1}(u)_j = \frac{\sum_{i \in S(u) - \{j\}} (dev_{j,i} + u_i) c_{j,i}}{\sum_{i \in S(u) - \{j\}} c_{j,i}} \dots\dots\dots(2)$$

$P^{S1}(u)_j$ = prediksi Slope One untuk item j

$c_{j,i}$ = $card(S_{j,i}(\mathcal{X}))$ = banyaknya elemen dalam $S_{j,i}(\mathcal{X})$

3. METODE PENELITIAN

3.1. Persiapan Data

Untuk mengetahui bagaimana kinerja *recommender system*, digunakan *dataset* rating yang telah terisi dengan *record* rating item dari *user*. Dataset yang digunakan dalam aplikasi web ini adalah dataset “MovieLens” [6], sebuah dataset rating film hingga tahun 2001. Dataset ini berisi 6.040 data *user*, 3.883 data item film, dan lebih dari 1 juta data rating dengan skala rating antara 1 sampai 5.

Tabel 1. Statistik dataset “MovieLens”

Jumlah <i>user</i>	6.040
Jumlah item	3.883
Banyaknya rating	1.085.433
Rata-rata rating per item	3,58
Rata-rata banyaknya rating per <i>user</i>	179,71
Rata-rata banyaknya rating per item	279,53

Pembersihan data dilakukan dengan menghapus rating yang tidak relevan dengan data *user* dan data item. Kategori item disederhanakan menjadi satu kategori saja. Sebelumnya, sebuah item film, seperti misalnya Toy Story, masuk dalam 3 kategori sekaligus yakni “animation”, “children”, dan “comedy”. Untuk mempermudah dalam implementasi, kategori diambil berdasarkan kategori yang tercantum paling awal sehingga untuk contoh film Toy Story dimasukkan dalam kategori “animation”.

3.2. Perancangan Tabel

Tabel-tabel database yang dibentuk sebagai berikut

1. Tabel *user*. Menyimpan data identitas setiap *user*. Data *user* yang didapat dari dataset “MovieLens” tidak mempunyai *field* nama *user* sehingga perlu dibuat sebuah *field* *username* agar sistem yang dibuat lebih interaktif.
2. Tabel *item*. Menyimpan data film yang terbagi dalam 17 kategori yang berbeda. Sebagaimana tabel *user*, tabel *item* juga dilengkapi dengan beberapa *field* seperti harga dan deskripsi item untuk memperjelas item.

Tabel 2. Atribut tabel *user*

Field	Tipe
<u><i>user_id</i></u>	int(11)
<i>username</i>	varchar(255)
<i>gender</i>	varchar(1)
<i>Age</i>	varchar(2)
<i>zip_code</i>	varchar(255)

Tabel 3. Atribut tabel *item*

Field	Tipe
<u><i>item_id</i></u>	int(11)
<u><i>category</i></u>	int(11)
<i>name</i>	int(11)
<i>description</i>	Text
<i>price</i>	Float

3. Tabel *rss_rating*. Merupakan komponen dasar dalam RSs berbasis *collaborative filtering* yang menyimpan data rating yang diberikan *user* terhadap item-item. Tabel *rss_rating* sangat diperlukan dalam pembuatan model item sebagai dasar perhitungan prediksi.
4. Tabel *rss_links*. Menyimpan data model item yang telah dibuat secara *offline*. Terdapat *field* *cnt_slope* dan *diff_slope* untuk dipergunakan dalam perhitungan prediksi. Tabel ini dapat di-*update* secara langsung, artinya tidak perlu lagi dilakukan pembuatan model dari awal setiap kali ada sebuah rating baru karena nilai rata-rata korelasi antara 2 item tidak disimpan dalam 1 *field*, akan tetapi dipecah menjadi 2 *field* yakni *cnt_slope* dan *diff_slope* sehingga perhitungan rata-rata dapat dilakukan dalam program.

Tabel 4. Atribut tabel *rss_rating*

Field	Tipe
<u><i>user_id</i></u>	int(11)
<u><i>item_id</i></u>	int(11)
<u><i>category</i></u>	int(11)
<i>rating</i>	Float
<i>ts</i>	timestamp(14)

Tabel 5. Atribut tabel *rss_links*

Field	Tipe	Keterangan
<u><i>item_id1</i></u>	int(11)	item yang diperbandingkan/ dipasangkan
<u><i>item_id2</i></u>	int(11)	item yang sedang dilihat
<i>category</i>	int(11)	kategori kedua item
<i>cnt_slope</i>	int(11)	<i>counter/</i> penghitung pasangan <i>item_id1</i> dan <i>item_id2</i>
<i>diff_slope</i>	Float	Selisih rating antara <i>item_id1</i> dan <i>item_id2</i>

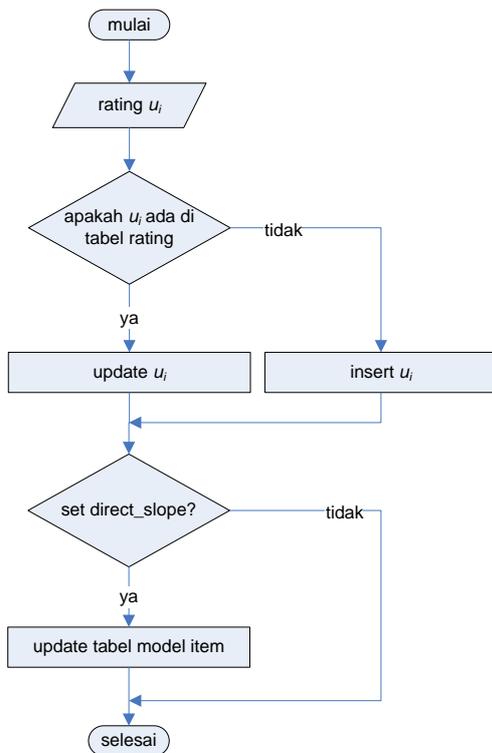
3.3. Alur Kerja Recommender System

1. Pemberian Rating Terhadap Item

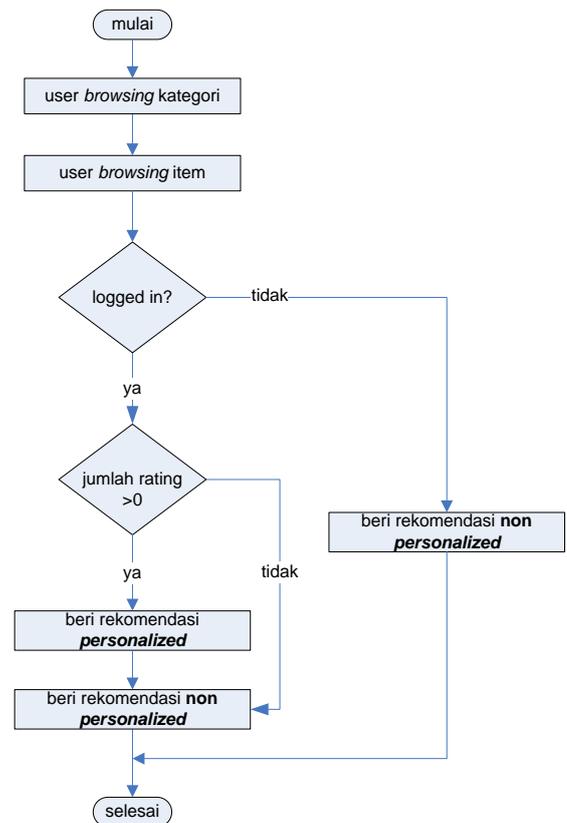
Proses pemberian rating oleh *user* terhadap sebuah item dapat ditunjukkan melalui *flowchart* pada Gambar 6. Rating u_i yang diberikan *user* u terhadap item i akan diperiksa apakah u_i sudah ada pada tabel rating atau belum. Jika sudah, u_i yang lama akan di-*update* dengan u_i yang baru, sebaliknya jika belum maka u_i akan dimasukkan ke dalam rating sebagai data rating baru. *Direct_slope* adalah sebuah fungsi yang melakukan *update* tabel *links* secara langsung pada saat rating diberikan. Jika fungsi *direct_slope* diaktifkan, maka setiap kali seorang *user* memberi rating terhadap sebuah item, maka data model item ter-*update* secara otomatis.

2. Pemberian Rekomendasi

Flowchart proses pemberian rekomendasi kepada user ditunjukkan pada Gambar 7. Pada saat user memilih item dalam sebuah kategori, diperiksa apakah user telah terdaftar atau belum. Jika user telah terdaftar, maka perlu diperiksa lagi apakah user telah memberi paling tidak sebuah rating pada sebuah item lain pada kategori yang sama. Jika kondisi benar, user akan diberi rekomendasi *personalized*. Namun jika belum, user akan diberi rekomendasi *non-personalized*. Bagi user yang tidak terdaftar atau *visitor*, akan selalu diberikan rekomendasi *non-personalized*.



Gambar 5. Alur pemberian rating terhadap item



Gambar 6. Alur pemberian rekomendasi kepada user

3.4. Pengukuran Tingkat Akurasi

Tingkat keakurasian rekomendasi yang dihasilkan dapat diukur dengan menggunakan salah satu metode standar dalam statistika yang disebut dengan *Mean Average Error* atau MAE [7]. Secara mendasar, MAE menghitung kesalahan/ *error* absolut antara rating yang sebenarnya (p) dan rating hasil prediksi (q), semakin kecil nilai MAE yang didapat maka prediksi yang dihasilkan semakin akurat. Jika pengukuran dilakukan terhadap N data, maka MAE dapat dirumuskan dengan:

$$MAE = \sum_{i=1}^N \frac{|p_i - q_i|}{N} \dots\dots\dots(3)$$

Berdasarkan beberapa penelitian tentang metode-metode dalam *collaborative filtering* yang telah dilakukan, nilai MAE yang akurat tidak lebih dari 1,000. Metode *user-based* misalnya, pada dataset MovieLens [6] mempunyai nilai MAE sebesar 0,755 [7]. Sedangkan salah satu metode *item-based* lainnya yakni *adjusted-cosine* mendapatkan nilai MAE sebesar 0,749 [7] pada dataset yang sama.

4. HASIL DAN PEMBAHASAN

4.1. Pembuatan Model Item

Model item dibuat berdasarkan data rating item dalam tabel rating. Oleh karena perhitungan prediksi rating dilakukan untuk masing-masing kategori, maka item-item yang diperbandingkan dipasangkan juga dalam satu kategori yang sama. Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Untuk setiap *user*, dicari item-item dalam satu kategori yang sama.
2. Selanjutnya, untuk masing-masing item, dipasangkan satu sama lainnya lalu dihitung selisih ratingnya. Setiap pasangan item yang terbentuk juga menambahkan 1 nilai *cnt_slope* untuk menunjukkan jumlah rating pasangan item.
3. Setelah diketahui pasangan item yang terbentuk dan selisih ratingnya, dilakukan penyimpanan ke dalam tabel *rss_links*.

Pembuatan model item dilakukan secara *offline* untuk mencari semua pasangan item sehingga dapat digunakan untuk perhitungan prediksi dan pembuatan rekomendasi secara *online*. Pembuatan awal model item dilakukan dengan menjalankan sebuah file *batch* yang menghitung selisih rating dan jumlah rating masing-masing pasangan item dari tabel *rss_rating*, kemudian menyimpannya dalam tabel *rss_links*.

Berikut ini adalah tabel yang menunjukkan waktu yang diperlukan untuk memproses komputasi pada setiap kategori. Banyaknya item dapat diketahui dari tabel *rss_rating*, yakni dengan melihat banyaknya item yang dipasang.

Tabel 6. Waktu komputasi pembuatan model item

Kategori	Banyaknya Item yang Dipasangkan	Waktu Komputasi (jam:menit:detik)
Action	486	06:43:10
Adventure	154	01:52:56
Animation	90	00:50:22
Children	88	00:30:05
Comedy	665	08:55:21
Crime	96	00:46:29
Documentary	95	00:35:05
Drama	325	05:33:14
Film-Noir	27	00:01:52
Horror	187	01:13:24
Musical	24	00:01:27
Mystery	35	00:01:59
Romance	44	00:00:19
Sci-Fi	44	00:02:20
Thriller	91	00:06:51
War	11	00:00:12
Western	31	00:01:15
Total	2493	27:16:21

4.2. Pemberian Rating

Pemberian rating adalah salah satu proses utama dalam sistem sebagai bentuk interaksi antara *user* dengan sistem. Pada proses ini, selain menyimpan data rating *user* ke dalam tabel *rss_ratings*, juga dapat dilakukan proses *update* secara langsung (*on-the-fly*) terhadap tabel *rss_links*. Dengan demikian, tabel model item selalu *ter-update* mengikuti pola rating dari para *user* sehingga tidak perlu lagi dilakukan pembuatan model item dari awal.

4.3. Fasilitas Recommender System

Fasilitas *Recommender System* diperuntukkan bagi *user* yang telah terdaftar dalam sistem. Kegunaan dari fasilitas ini adalah untuk mendapatkan input rating dari *user* sekaligus memudahkan *user* merating item-item dalam rangka meningkatkan kualitas rekomendasi bagi *user* yang bersangkutan.

4.3.1. Halaman Rating User

Halaman rating *user* menampilkan banyaknya item yang dirating, rata-rata rating yang diberikan pada setiap itemnya, serta daftar semua item yang telah diberi rating oleh *user* yang bersangkutan.

4.3.2. Halaman Beri Rating

Halaman ini dikhususkan bagi *user* untuk melakukan rating terhadap beberapa item sekaligus dalam 1 kategori yang diketahuinya tanpa harus menuju halaman detail item terlebih dahulu. Item dapat ditampilkan dengan urutan kode atau nama item.

Rating Anda
Anda telah merating 33 item, dengan rata-rata 3.79 / 5.

ID	Item	Kategori	Rating / 5
2019	Seven Samurai (The Magnificent Seven) (Shichinin no samurai) (1954)	Action	5.00
3408	Erin Brockovich (2000)	Drama	5.00
2353	Enemy of the State (1998)	Action	5.00
1350	Omen, The (1976)	Horror	5.00
3952	Contender, The (2000)	Drama	5.00
3785	Scary Movie (2000)	Comedy	5.00
733	Rock, The (1996)	Action	5.00
500	Mrs. Doubtfire (1993)	Comedy	5.00
6	Heat (1995)	Action	5.00
3578	Gladiator (2000)	Action	5.00
2581	Never Been Kissed (1999)	Comedy	5.00
3452	Romeo Must Die (2000)	Action	5.00
3793	X-Men (2000)	Action	4.00
1438	Dante's Peak (1997)	Action	4.00
2355	Bug's Life, A (1998)	Animation	4.00
2706	American Pie (1999)	Comedy	4.00

Gambar 7. Tampilan halaman rating oleh *user*

Beri Rating pada Kategori Top > Adventure

Kategori: None

Item:

ID	Nama Item	Rating Anda					
		?	1	2	3	4	5
2	Jumanji (1995)	<input checked="" type="radio"/>	<input type="radio"/>				
8	Tom and Huck (1995)	<input checked="" type="radio"/>	<input type="radio"/>				
29	City of Lost Children, The (1995)	<input checked="" type="radio"/>	<input type="radio"/>				
33	Wings of Courage (1995)	<input checked="" type="radio"/>	<input type="radio"/>				
56	Kids of the Round Table (1995)	<input checked="" type="radio"/>	<input type="radio"/>				
60	Indian in the Cupboard, The (1995)	<input checked="" type="radio"/>	<input type="radio"/>				
86	White Squall (1996)	<input checked="" type="radio"/>	<input type="radio"/>				
107	Muppet Treasure Island (1996)	<input checked="" type="radio"/>	<input type="radio"/>				
126	NeverEnding Story III, The (1994)	<input checked="" type="radio"/>	<input type="radio"/>				
146	Amazing Panda Adventure, The (1995)	<input checked="" type="radio"/>	<input type="radio"/>				

[Simpan Profil Rating](#)

Gambar 8. Tampilan halaman beri rating

4.3.3. Halaman Rekomendasi *User*

Halaman ini menampilkan hasil perhitungan prediksi pada setiap kategori berupa sebuah daftar item sehingga memudahkan *user* mendapatkan rekomendasi dengan segera tanpa harus melakukan *browsing* pada halaman katalog item.

Daftar item yang pernah Anda beli.
Anda dapat memberi rating pada item-item ini untuk menunjukkan tingkat kepuasan Anda pada item yang sudah Anda beli tersebut.
Silakan klik pada nama item untuk memberi rating pada item tersebut.

ID	Item	Kategori
3578	Gladiator (2000)	Action
3578	Gladiator (2000)	Action
3408	Erin Brockovich (2000)	Drama
1	Toy Story (1995)	Animation
2	Jumanji (1995)	Adventure
2	Jumanji (1995)	Adventure
6	Heat (1995)	Action
10	GoldenEye (1995)	Action
652	301, 302 (1995)	Mystery
904	Rear Window (1954)	Mystery
77	Nico Icon (1995)	Documentary
192	Show, The (1995)	Documentary

Gambar 9. Tampilan halaman item yang sudah dibeli

Rekomendasi Anda pada Kategori Top > Animation

Kategori: None

Item:

Rekomendasi untuk Anda		
1148	Wrong Trousers, The (1993)	4.00
1223	Grand Day Out, A (1992)	3.73
741	Ghost in the Shell (Kokaku kidota) (1995)	3.60
2761	Iron Giant, The (1999)	3.49
1151	Faust (1994)	3.36
3751	Chicken Run (2000)	3.36
1023	Winnie the Pooh and the Blustery Day (1968)	3.31
2138	Watership Down (1978)	3.26
2857	Yellow Submarine (1968)	3.22
2700	South Park: Bigger, Longer and Uncut (1999)	3.20
1282	Fantasia (1940)	3.20
595	Beauty and the Beast (1991)	3.19
2495	Fantastic Planet, The (La Planète sauvage) (1973)	3.17
364	Lion King, The (1994)	3.15
3159	Fantasia 2000 (1999)	3.14
3213	Batman: Mask of the Phantasm (1993)	3.11
2139	Secret of NIMH, The (1982)	3.11

Gambar 10. Tampilan halaman rekomendasi *user*

4.3.4. Rekomendasi Non-Personalized

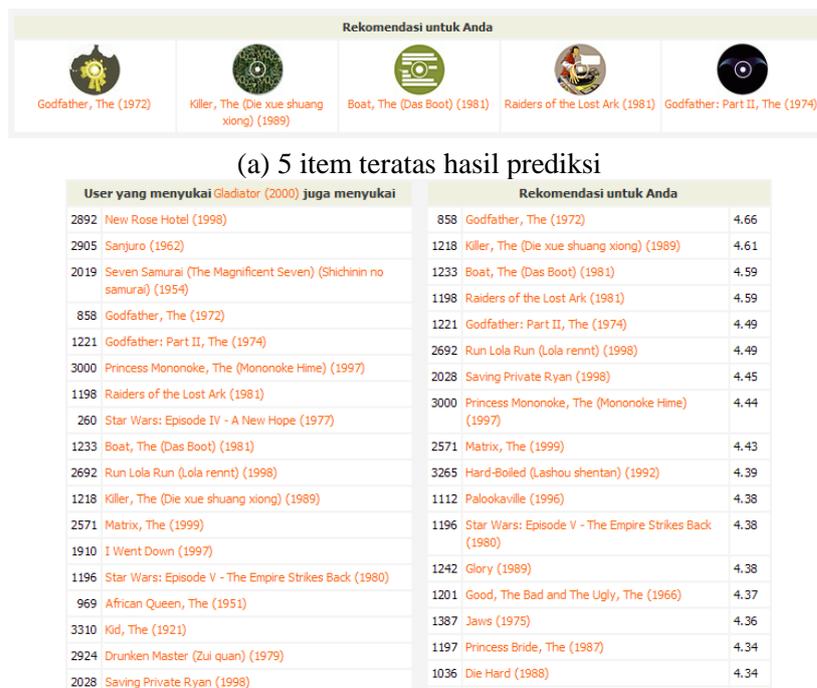
Rekomendasi yang bersifat *non-personalized* ditampilkan berupa 5 item teratas yang paling berkorelasi dengan item yang sedang dilihat/ ditampilkan. Untuk mengetahui semua item yang berkorelasi, disediakan tampilan lengkap seperti pada Gambar 11 (b).



Gambar 11. Tampilan rekomendasi *non-personalized*

4.3.6. Rekomendasi *Personalized*

Rekomendasi *personalized* ditampilkan dalam bentuk 2 output, yakni 5 item teratas dari hasil perhitungan prediksi dan 5 item teratas yang paling berkorelasi dengan item yang sedang dilihat/ditampilkan sebagaimana pada rekomendasi *non-personalized*. Namun, apabila *user* menginginkan tampilan keseluruhan rekomendasi, *user* dapat memilih menu yang akan secara lengkap perhitungan prediksi dan korelasi antar item, seperti diperlihatkan pada Gambar 12(b).



Gambar 12. Tampilan rekomendasi *personalized*

4.4. Analisa Hasil

Jika seorang *user* belum merating satu pun item dalam sebuah kategori, dia tidak memperoleh rekomendasi *personalized*. Hal ini disebabkan karena sistem tidak dapat mengetahui preferensi rating dari *user* yang bersangkutan sehingga tidak ada input \$itemID1 untuk diikutsertakan dalam perhitungan prediksi pada kategori tersebut. Sebagai solusinya, *user* masih dapat menerima rekomendasi berupa daftar item-item yang berkorelasi dengan item yang sedang dilihat, seperti halnya pada rekomendasi *non-personalized*.

Pengujian dilakukan untuk mengetahui nilai keakuratan prediksi dan waktu komputasi *online* dari metode *item-based* yang digunakan. Pada pengujian ini, dataset yang digunakan adalah dataset "MovieLens" sebagaimana telah digunakan dalam implementasi. Dari hasil pengujian, besarnya *MAE* yang didapatkan adalah sebesar 0,641, sehingga dapat disimpulkan bahwa rating setiap item yang diprediksi mengalami rata-rata penyimpangan sebesar 0,641 dari rating *user* yang sebenarnya dalam skala rating 1-5.

Untuk mengetahui waktu komputasi, dilakukan 5 kali percobaan dengan menampilkan prediksi rating item bagi 5 *user* yang berbeda, secara *online* pada halaman web. Kelima *user* diambil secara random dan dicatat berapa waktu yang diperlukan untuk menampilkan keseluruhan prediksi berdasarkan kategori dan item yang sedang dilihat/ dipilih.

Tabel 7. Hasil uji coba waktu komputasi prediksi secara *online*

Uji Coba	UserID	ItemID yang Dilihat	Jumlah item yang diprediksi	Waktu komputasi
1	6041	1198	460	1,25 detik
2	3600	586	58	0,69 detik
3	0048	1175	191	0,87 detik
4	0014	3578	481	1,71 detik
5	6064	1221	475	1,60 detik
			Rata-rata	1,22 detik

5. KESIMPULAN

- Model korelasi antar item dapat di-*update* secara langsung dan tidak hanya bergantung pada proses pembuatan model item dari awal, artinya sebuah rating baru dapat mempengaruhi hasil prediksi rating sebuah item.
- Keakuratan hasil perhitungan prediksi dicapai dengan nilai *mean average error* yang didapatkan sebesar 0,641.
- Hasil perhitungan prediksi secara *online* dalam sebuah halaman web dapat ditampilkan dalam kisaran waktu kurang dari 2 detik sehingga mampu memberikan rekomendasi *real-time* kepada *user*.

6. SARAN

Untuk mempertajam hasil rekomendasi, dapat dilakukan pengklasifikasian *user* berdasarkan data demografinya, sehingga akan dapat diketahui pola seperti misalnya: *user* yang berusia antara 18–25 menyukai film-film aksi.

DAFTAR PUSTAKA

- [1] Amazon, <http://www.amazon.com>. Tanggal akses: 21-04-2006.
- [2] Barneveld, J.J.F. *User Interfaces for Personalized Information Systems*. Telematica Instituut. 2003.
- [3] CDNOW, <http://www.cdnw.com>. Tanggal akses: 21-04-2006.
- [4] Lemire, Daniel, Anna Maclachlan. *Slope one predictors for online rating-based collaborative filtering*. SIAM Data Mining (SDM'05). 2005.
- [5] Mladenic, Dunja. *Text-Learning and Related Intelligent Agents: A Survey*. IEEE. 1999.
- [6] MovieLens, <http://www.cs.umn.edu/GroupLens>. Tanggal akses: 20-04-2006.

ORIGINAL ARTICLE

- [7] Sarwar, B. M., G. Karypis, J. A. Konstan, dan J. Riedl. *Item-Based Collaborative Filtering Recommender Algorithms*. WWW10. 2001.
- [8] Schafer, J. B., J. A. Konstan, dan J. Riedl, *E-Commerce Recommendation Application*. Data Mining and Knowledge Discovery. 2001.
- [9] Weiyang, Lin. *Association Rule Mining for Collaborative Recommender Systems*. Thesis. Worcester Polytechnic Institute. 2000.