

**IMPLEMENTASI *RECOMMENDER SYSTEM* DENGAN
MENGUNAKAN ALGORITMA *SLOPE-ONE* PADA
WEB *E-COMMERCE* PENJUALAN DVD GAMES**

SKRIPSI



Disusun Oleh :

**Aldi K. Dipasanta
NIM. 0512720**

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011**

LEMBAR PERSETUJUAN

**IMPLEMENTASI RECOMMENDER SYSTEM DENGAN
MENGUNAKAN ALGORITMA SLOPE-ONE PADA
WEB E-COMMERCE PENJUALAN DVD GAMES**

SKRIPSI

*Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai
gelar Sarjana Teknik*

Disusun Oleh:

**Aldi K. Dipasanta
NIM. 0512720**

Mengetahui
Ketua Program Studi Teknik Elektro S-1

Diperiksa dan Disetujui
Dosen Pembimbing

Ir. Yusuf Ismail Nakhoda, MT.
NIP.V. 1018890198

I Komang Somawirata, ST, MT.
NIP.P. 1030100361

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011**

ABSTRAK

IMPLEMENTASI *RECOMMENDER SYSTEM* DENGAN MENGUNAKAN ALGORITMA *SLOPE-ONE* PADA WEB *E-COMMERCE* PENJUALAN DVD GAMES

Aldi K. Dipasanta, NIM : 0512720

Dosen Pembimbing : I Komang Somawirata, ST, MT.

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 Systems* (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* dapat digunakan untuk membantu user dalam memilih item yang sesuai dengan selera user terhadap item berdasarkan rating yang diberikannya.

Dalam Skripsi ini, dibuat sebuah web *e-commerce* sederhana 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 (*visitor*). Masukan dari pengguna (*user*) berupa rating yang diberikan kepada barang (*item*) dapat langsung mempengaruhi hasil prediksi tidak hanya bergantung pada pembuatan model item dari awal. Dengan menggunakan menggunakan metode *Mean Average Error* (MAE) didapatkan nilai keakurasian sebesar 1,74, sehingga dapat disimpulkan bahwa rating setiap item yang diprediksi mengalami rata-rata penyimpangan sebesar 1,742 dari rating pengguna (*user*) yang sebenarnya dalam skala rating 1- 5. Hasil uji coba komputasi prediksi secara *online* didapatkan nilai dibawah 1 detik, sehingga sehingga mampu memberikan rekomendasi *real-time* kepada pengguna. Dengan adanya sistem rekomendasi ini diharapkan dapat membantu pengguna untuk menemukan barang yang diinginkan dan dapat membentuk sebuah komunitas yang saling memberi masukan.

Kata Kunci: *Rekomendasi, Slope One, e-commerce*

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa karena hanya dengan limpahan Rahmat, Karunia dan Hidayah-Nya sehingga Penulis dapat menyelesaikan Skripsi dengan judul **“IMPLEMENTASI *RECOMMENDER SYSTEM* DENGAN MENGGUNAKAN ALGORITMA *SLOPE-ONE* PADA WEB *E-COMMERCE* PENJUALAN DVD GAMES”**.

Skripsi ini diajukan sebagai syarat untuk memperoleh gelar Sarjana di Fakultas Industri, Jurusan Teknik Elektro, Program Studi Komputer dan Informatika, Institut Teknologi Nasional Malang. Atas terselesaikannya Skripsi ini, Penulis mengucapkan terima kasih kepada:

1. Bapak Prof Dr. Ir. Abraham Lomi, MSEE , selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noertjahjono, MT , selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT , selaku Ketua Jurusan Teknik Elektro Institut Teknologi Nasional Malang.
4. Bapak I Komang Somawirata, ST, MT , selaku Dosen Pembimbing.
5. Segenap Bapak dan Ibu Dosen yang telah mendidik dan mengajarkan ilmu selama Penulis menempuh pendidikan di Jurusan Teknik Elektro Institut Teknologi Nasional Malang.
6. Segenap Staff dan Karyawan di Jurusan Teknik Elektro yang telah membantu dalam pelaksanaan Skripsi ini.
7. Orang Tua atas dukungan materi dan doa restunya.
8. Semua pihak yang telah membantu terselesaikannya Skripsi ini.

Penulis menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penulis mengharapkan saran dan kritik yang membangun demi kesempurnaan laporan ini.

Malang, Februari 2011

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR TABEL	v
DAFTAR GAMBAR.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah.....	3
1.5 Metode Penelitian.....	3
1.5.1 Studi Literatur.....	3
1.5.2 Metode Perancangan Sistem.....	4
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	5
2.1 Data Mining	5
2.2 Web Mining	7
2.3 Recommender Systems	8
2.3.2 Content-Based Filtering.....	10
2.3.3 Collaborative Filtering.....	10
2.4 Algoritma Slope One.....	13
2.4.1 Deskripsi Algoritma Slope One.....	13
2.4.2 Pendekatan Algoritma Slope One	13
2.5 Pengukuran Tingkat Akurasi.....	15
2.6 MySQL.....	15
2.7 PHP	16
BAB III METODE DAN PERANCANGAN SISTEM.....	18
3.1 Perancangan Collaborative-Filtering	18
3.1.1 Pembuatan Model Korelasi Antar-Item.....	18
3.1.2 Rekomendasi <i>Non-Personalized</i>	22
3.1.3 Rekomendasi <i>personalized</i>	23

3.1.3.1	Prediksi per Item	23
3.1.3.2	Rekomendasi Top-N	25
3.2	Perancangan Tabel	26
3.2.1	Tabel user	26
3.2.2	Tabel item.....	26
3.2.3	Tabel rating.....	28
3.2.4	Tabel Slope One	28
3.3	Alur Kerja Recommender System	30
3.3.2	Proses Login	31
3.3.3	Proses Pemberian Rating.....	32
BAB IV	IMPLEMENTASI DAN PEMBAHASAN	33
4.1	Implementasi	33
4.1.1	Persiapan Data	33
4.1.2	Implementasi Basis Data	33
4.1.3	Deskripsi Program.....	34
4.1.3.1	Pembuatan Model Item	34
4.1.3.2	Pemberian Rating.....	35
4.1.3.3	<i>Update</i> Model Item	36
4.1.3.4	Pemberian Rekomendasi <i>non-personalized</i>	37
4.1.3.5	Pemberian Rekomendasi <i>personalized</i>	37
4.2	Penerapan Aplikasi.....	38
4.2.1	Pembuatan Model Item.....	38
4.2.2	Web <i>e-commerce</i> penjualan dvd games	39
4.2.3	Katalog item	39
4.2.4	Detil item	41
4.2.5	<i>Recommender Systems</i>	43
4.2.5.1	Rekomendasi <i>non-personalized</i>	43
4.2.5.2	Rekomendasi <i>personalized</i>	44
4.2.6	Shopcart.....	45
4.3	Analisa Hasil	46
BAB V	PENUTUP	50

5.1	Kesimpulan	50
5.2	Saran.....	51
DAFTAR PUSTAKA.....		52
LAMPIRAN		

DAFTAR TABEL

Tabel 3.1	Contoh data rating dari User.....	18
Tabel 3.2	Data Rating dari User 1	19
Tabel 3.3	Data Korelasi Sementara.....	19
Tabel 3.4	Data rating dari user 2	20
Tabel 3.5	Data korelasi antar item setelah di- <i>update</i>	20
Tabel 3.6	Data model item dengan jumlah rating > 1	22
Tabel 3.7	Rekomendasi <i>non-personalized</i> bagi visitor	23
Tabel 3.8	Data korelasi antar item yang memuat item_ID2 - C.....	24
Tabel 3.9	Rekomendasi <i>personalized</i> bagi user 3	26
Tabel 3.10	Atribut tabel rss_user.....	26
Tabel 3.11	Atribut tabel rss_catalog_items	27
Tabel 3.12	Atribut tabel rss_user_ratings.....	28
Tabel 3.13	Atribut tabel rss_slope_one	29
Tabel 4.1	Statistik dataset.....	33
Tabel 4.2	Waktu komputasi pembuatan model item	38
Tabel 4.3	Contoh 10 rating hasil prediksi.....	48
Tabel 4.4	Hasil uji coba waktu komputasi prediksi secara <i>online</i>	49

DAFTAR GAMBAR

Gambar 2.1	Skema model-based / item-based	6
Gambar 2.2	Web Mining	7
Gambar 2.3	Recommender Systems	9
Gambar 2.4	Proses Collaborative filtering	11
Gambar 2.5	Skema memory-based / user-based	12
Gambar 2.6	Skema model-based / item-based	13
Gambar 2.7	Skema Algoritma <i>Slope One</i> ⁽⁹⁾	14
Gambar 3.1	Alur Kerja Recommender Systems	30
Gambar 3.2	Alur kerja proses log in	31
Gambar 3.3	Alur kerja proses pemberian rating	32
Gambar 4.1	Tampilan katalog berdasarkan kategori	39
Gambar 4.2	Halaman Pendaftaran dan Login	40
Gambar 4.3	Tampilan katalog pada user yang telah terdaftar	41
Gambar 4.4	Tampilan katalog per-item	42
Gambar 4.6	Rekomendasi non-personalized	43
Gambar 4.7	Tampilan rekomendasi <i>personalized</i>	45
Gambar 4.8	Keranjang Belanja	46
Gambar 4.9	Tampilan permintaan pada Microsoft [®] Outlook	46

BAB I

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun terakhir ini tidak dapat dipungkiri bahwa internet telah mengalami perkembangan yang sangat pesat. Beragam informasi tersedia di internet dan dapat diakses secara luas oleh penggunanya. Akan tetapi, dengan banyaknya data baik berupa teks, gambar, maupun video yang ada di internet saat ini menyebabkan terjadinya luapan informasi yang membuat pengguna atau *user* internet seringkali mengalami kesulitan dalam mencari informasi yang benar-benar sedang dicari atau dibutuhkan. Dalam perdagangan toko *online* atau *e-commerce*, seperti penjualan DVD Games misalnya, kesulitan yang dihadapi pengguna internet sebagai calon konsumen adalah ketika mencari sebuah produk yang sesuai dengan keinginan dan kurangnya informasi akan produk tersebut dari konsumen lain. 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 Systems*.

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

Secara umum, ada 2 macam metode yang digunakan dalam RSs, yaitu *content-based filtering* dan *collaborative filtering*⁽³⁾. *Content-based filtering* bekerja dengan mencari item yang mempunyai korelasi dengan item yang disukai pengguna berdasarkan *content* / informasi tekstual tiap item. Sedangkan *collaborative filtering* (CF) merekomendasikan item kepada seorang pengguna berdasarkan rating yang diberikannya terhadap item. CF lebih banyak digunakan karena dalam beberapa domain, seperti games dan musik, sulit dilakukan ekstraksi *content* tiap item yang merupakan langkah utama dari *content-based filtering*⁽⁴⁾.

Berbagai penelitian tentang RSs berbasis CF telah banyak dilakukan dengan berbagai teknik. RSs berbasis CF yang ada pertama kali menggunakan metode *nearest-neighbor*. Metode ini dikenal pula dengan nama *memory-based* atau *user-based* karena mencari pengguna yang paling mirip dengan pengguna target dalam hal merating item, lalu merekomendasikan item-item yang disukai pengguna terdekat kepada pengguna target. Kelemahan metode *memory-based* adalah masalah skalabilitas seiring dengan makin banyaknya jumlah pengguna dan jumlah item⁽⁵⁾. Metode *model-based* atau *item-based* muncul sebagai solusi metode *user-based* CF. Berbeda dengan metode *user-based* yang menghitung kemiripan antar pengguna 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*⁽⁵⁾.

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

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka rumusan masalah yang diambil adalah:

1. Diperlukan sebuah personalisasi di internet yang mampu beradaptasi dengan masing-masing pengguna.
 2. Diperlukan Sistem yang akurat dengan menggunakan metode yang memiliki skalabilitas luas dan performa komputasi yang baik .
-

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penulisan Skripsi ini antara lain:

1. Membuat sebuah Sistem Rekomendasi (*Recommender Systems*) yang bekerja dengan cara memberi sugesti berdasarkan tingkat ketertarikan pengguna.
2. Menghasilkan rekomendasi *online* berdasarkan metode *item-based collaborative filtering* dengan Algoritma Slope-One.
3. Implementasi *Recommender Systems (RSs)* pada sebuah web *e-commerce*.

1.4 Batasan Masalah

Batasan masalah dalam pembuatan Skripsi ini adalah sebagai berikut:

1. Metode *item-based collaborative filtering* yang digunakan hanya melakukan analisis terhadap data rating item.
2. Algoritma *slope-one* yang digunakan untuk mencari korelasi antar item.
3. Input sistem secara eksplisit dengan rating yang digunakan adalah 1 – 5 dengan kenaikan 1 angka.
4. Web *e-commerce* yang digunakan hanya berperan sebagai alat simulasi.
5. Sebuah item produk hanya masuk dalam 1 kategori saja, pengelompokan kategori berdasarkan *genre* yang didapat dari Gamefaqs^(b).
6. Bahasa pemrograman web yang digunakan adalah PHP, sedangkan web server dan database server menggunakan Apache dan MySQL (*offline*).
7. Analisa hasil yang dilakukan menggunakan metode *Mean Absolut Error (MAE)* sebagai pengukuran tingkat akurasi dan pengujian komputasi dilakukan secara online.

1.5 Metode Penelitian

1.5.1 Studi Literatur

Kajian tentang Data Mining dan *Knowledge Discovery in Database (KDD)* yang berhubungan dengan *Recommender System (RSs)* sebagai salah satu bentuk tujuan Data Mining yaitu *Personalized Information System*.

1.5.2 Metode Perancangan Sistem

- a) *Analisa Sistem*. Menentukan struktur dan representasi data set / item. Memilih Algoritma yang tepat untuk pemberian rating. Bertujuan untuk manajemen data yang dibutuhkan Sistem Rekomendasi secara efisien.
- b) *Perancangan Sistem*. Proses perancangan *Collaborative Filtering*. Pembuatan model korelasi antar item mencakup rating dan prediksi. Perancangan Tabel dan *diagram flow Recommender Systems (RSs)*.
- c) *Implementasi dan Pembuatan Sistem*. Persiapan Data dan Implementasi Basis Data. Pembuatan program Model Item mencakup, korelasi item, pemberian rating, prediksi item. Pembuatan sebuah web *e-commerce* sederhana dan Penerapan Aplikasi.
- d) *Uji Analisa*. Bertujuan untuk mengetahui nilai keakuratan prediksi dan waktu komputasi.

1.6 Sistematika Penulisan

Disusun Sistematika penulisan Skripsi yang terbagi atas 5 bab, yakni:

BAB I. PENDAHULUAN

Berisikan tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penulisan, Manfaat Skripsi dan Sistematika Penulisan.

BAB II. LANDASAN TEORI

Berisikan tinjauan umum teori – teori yang mendasari *Recommender Systems(RSs)*.

BAB III. METODE DAN PERANCANGAN

Berisikan penjelasan metode dan algoritma *Collaborative Filtering* yang dipakai untuk *Recommender Systems(RSs)*.

BAB IV. IMPLEMENTASI DAN UJI ANALISA

Berisikan pembahasan tentang penerapan *Recommender Systems (RSs)* dan uji analisa yang dilakukan.

BAB V. PENUTUP

Merupakan Bab terakhir yang berisi kesimpulan dan saran.



BAB II

LANDASAN TEORI

2.1 Data Mining

Data mining adalah suatu proses dalam mencari korelasi, pola, atau tren yang bermanfaat dari sebuah data yang berukuran besar dengan menggunakan teknik statistika ataupun matematika ⁽⁷⁾. Istilah data mining dan *knowledge discovery in databases* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, akan tetapi berkaitan satu sama lain, dimana salah satu tahapan dalam keseluruhan proses KDD adalah data mining.

Secara garis besar proses KDD dapat dijelaskan sebagai berikut ⁽⁸⁾:

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing / Cleaning*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak.

3. *Transformation*

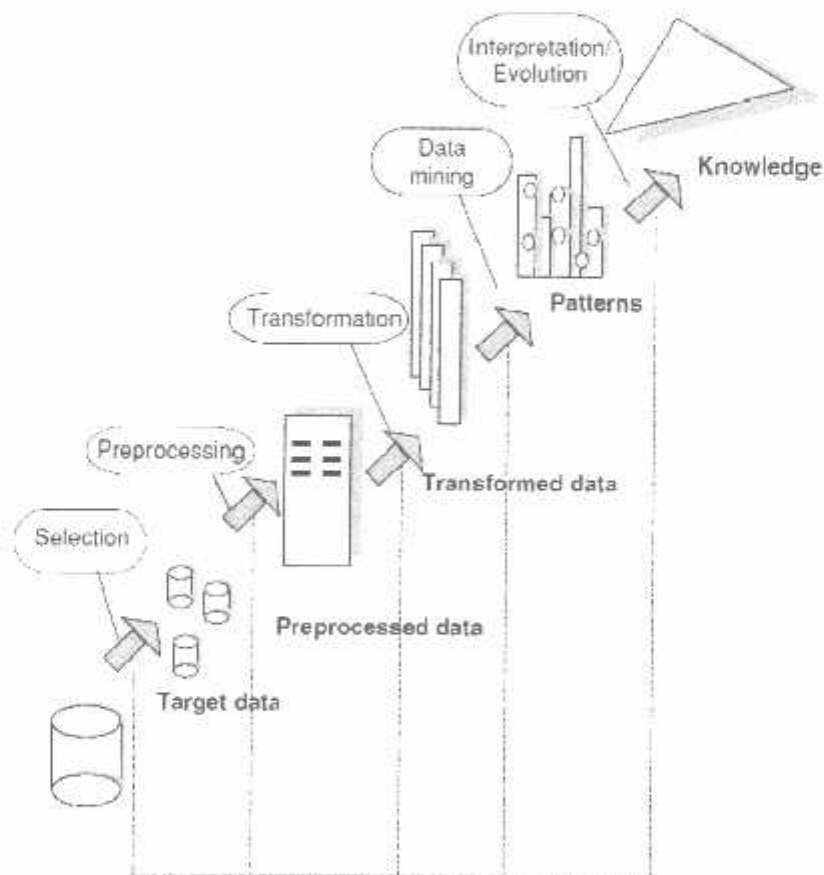
Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data Mining*

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation / Evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

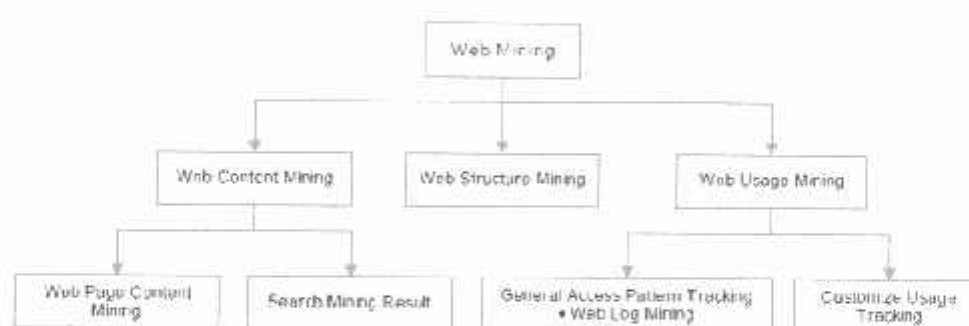


Gambar 2.1 Skema model-based / item-based

2.2 Web Mining

Pendekatan teknik-teknik dalam data mining dapat juga dilakukan pada web, yang disebut dengan web mining. Dengan kata lain web mining adalah data mining yang diaplikasikan pada data web. Data web itu sendiri adalah data yang berkaitan dengan web, mulai dari teks, halaman HTML, data multimedia, data transaksi, hingga *web server log* ⁽⁹⁾.

Tujuan utama web mining adalah bagaimana membuat sistem yang bersifat personal bagi setiap pengguna yang disebut dengan *personalized information system*. *Personalized information system* adalah sebuah sistem informasi yang menyediakan konten dan servis, beradaptasi terhadap ketertarikan, preferensi, atau kebiasaan pengguna, dan mampu membantu pengguna dalam menemukan tujuannya.



Gambar 2.2 Web Mining

Berbagai data diolah dan dianalisa untuk menghasilkan sistem yang bersifat personal bagi pengguna. Berdasarkan jenis data web yang dianalisa atau yang digali, web mining dapat dibagi menjadi 3 kategori ⁽¹⁰⁾, yaitu:

1. *Web Content Mining*

Ada 2 macam *web content mining* yaitu *web page content* yang mengekstrak informasi penting dari *content* (isi) dokumen web yang bisa berupa teks biasa atau halaman HTML/XML, dan *search result mining* yang melakukan analisa terhadap hasil pencarian.

2. *Web Structure Mining*

Web structure mining mengambil informasi dari struktur web seperti *hyperlink*. Seperti misalnya, banyaknya *link* yang mengarah ke sebuah halaman web menandakan popularitas dan penting tidaknya halaman web yang dituju,

sedangkan banyaknya *link* yang ke luar dari suatu halaman web menandakan halaman tersebut mempunyai banyak topik atau kaya akan isi.

3. *Web Usage Mining*

Web usage mining mengekstrak informasi dari log akses web (*web server log*) yang berisi catatan segala interaksi antara pengguna dan web. Dua macam *web usage mining* adalah *General Access Pattern Tracking* yang mengamati aktivitas pengguna secara umum terhadap semua pengguna lalu melakukan penyesuaian secara menyeluruh, sedangkan *Customized Usage Tracking* menganalisa aktivitas tiap pengguna dan melakukan penyesuaian terhadap pengguna yang bersangkutan.

2.3 Recommender Systems

Recommender system (RSs) adalah salah satu bentuk *personalized information system* yang digunakan dalam web *ecommerce* untuk menawarkan item kepada pengguna dan memberi informasi yang dapat membantu pengguna dalam memilih atau membeli item⁽⁹⁾.

Sebagaimana dapat dilihat pada gambar 2.3, ada 3 komponen utama dalam sebuah *recommender system*⁽⁹⁾, yaitu:

1. Input / Output

Input yang dianalisa RSs didapat baik secara eksplisit maupun implisit dari pengguna lalu dikombinasikan dengan input dari para pengguna lain atau komunitas. Input yang didapat secara eksplisit, misalnya dengan cara meminta pengguna untuk memberi rating terhadap suatu item. Sedangkan secara implisit, misalnya dari data transaksi pembelian item oleh pengguna pada waktu lampau, atau bisa juga dengan memonitor item-item mana saja yang telah dilihat oleh pengguna.

Sedangkan 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 pengguna (*user-to-user*), atau juga dengan berdasarkan korelasi antar item (*item-to-item*).

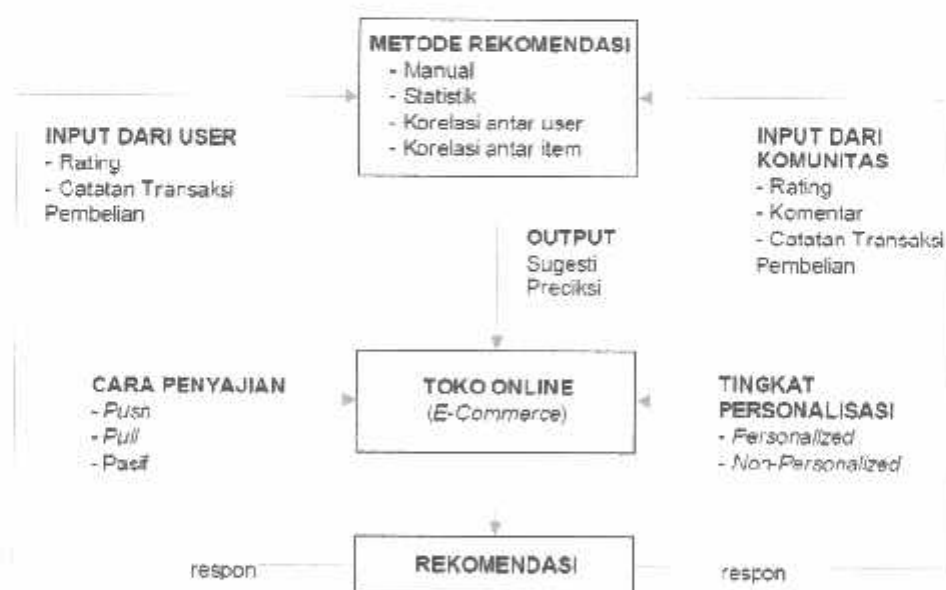
3. Desain Rekomendasi

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

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

Sedangkan tingkat personalisasi rekomendasi yang diberikan kepada pengguna ada 2 macam, yaitu:

- *Personalized* : Diberikan kepada pengguna berdasarkan masing-masing profil pengguna.
- *Non-personalized* : Bersifat umum dapat diberikan kepada pengunjung (*non-registered user*).



Gambar 2.3 Recommender Systems

Berbagai teknik data mining seperti *association rule* dan *market basket analysis*, *nearest neighbor*, hingga *clustering* telah digunakan dalam membangun sebuah RSs.

Namun, secara garis besar teknik yang digunakan dalam RSs ada 2 macam, yakni *contentbased filtering* dan *collaborative filtering* ⁽⁹⁾.

2.3.2 Content-Based Filtering

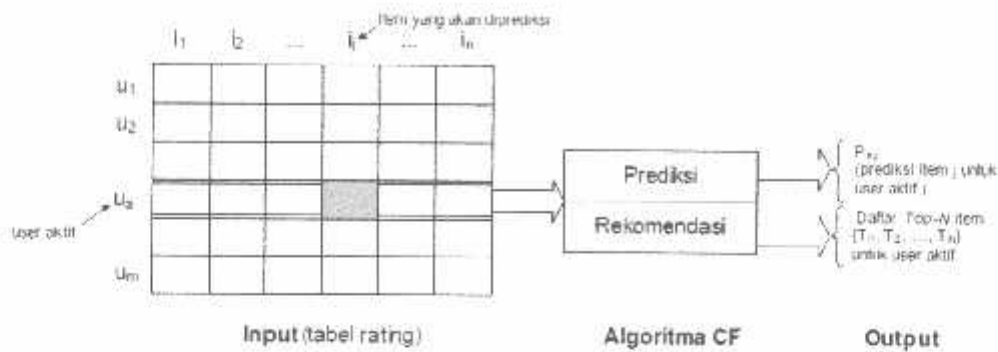
Content-based filtering (CBF) bekerja dengan cara mencari item lain yang mirip dengan item yang disukai atau yang pernah dibeli oleh pengguna berdasarkan informasi *content* / tekstual dari setiap item. CBF banyak digunakan untuk penyaringan informasi pada sistem berbasis teks, seperti fasilitas "*related pages*" dalam *search engine*. Pada sistem tersebut, sebuah algoritma untuk query pencarian diaplikasikan terhadap sebuah matriks berisi pasangan dokumen web dan *keyword*. Oleh karena itu, metode CBF disebut juga dengan *search-based filtering*. Untuk pengguna yang hanya memiliki sedikit record dalam catatan transaksi atau data rating, metode CBF berjalan dengan baik. Namun, ketika pengguna telah memiliki banyak record item baik yang telah dibeli maupun yang dirating, maka pencarian terhadap semua item yang mirip menjadi tidak praktis lagi.

Secara umum, kualitas rekomendasi yang dihasilkan CBF relatif rendah. Pada kasus tertentu, rekomendasi bersifat terlalu umum (misalnya hanya judul film terlaris dalam kategori drama) atau terlalu khusus (hanya merekomendasikan buku dengan penulis yang sama). Kelemahan lain, pada domain seperti musik, film, atau gambar, CBF sulit diimplementasikan karena penyaringan informasi tekstual sulit dilakukan.

2.3.3 Collaborative Filtering

Collaborative filtering (CF) adalah suatu metode dalam membuat sebuah prediksi otomatis untuk memperkirakan ketertarikan atau selera seorang pengguna terhadap suatu item dengan cara mengumpulkan informasi dari para pengguna yang lain yang direpresentasikan dalam bentuk nilai rating ⁽⁵⁾. Secara umum, ada 2 proses yang dilakukan dalam CF, yaitu:

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



Gambar 2.4 Proses Collaborative filtering

Pengguna (*User*) sebanyak m , $U = \{u_1, u_2, \dots, u_m\}$.

Item sebanyak n , $I = \{i_1, i_2, \dots, i_n\}$.

User u_i merating item ke dalam daftar I_{u_i} , baik secara eksplisit maupun implisit.

$I_{u_i} \subseteq I$, dan I_{u_i} bisa juga himpunan kosong.

Pengguna aktif dinotasikan dengan u_a dimana, $u_a \in U$, merupakan tujuan output *collaborative filtering*.

- **Prediksi** : $P_{a,j}$, dimana $i_j \notin I_{u_a}$ untuk pengguna aktif u_a . Nilai prediksi besarnya rating item j yang mungkin didapat dari pengguna aktif, dimana item j belum pernah dirating oleh pengguna aktif.
- **Rekomendasi** : satu atau beberapa dalam bentuk daftar N item, $I_r \subset I$ dan $I_r \cap I_{u_a} = \emptyset$, berisi kemungkinan terbesar untuk disukai oleh pengguna aktif. Disebut juga *Top-N Recommendation*.

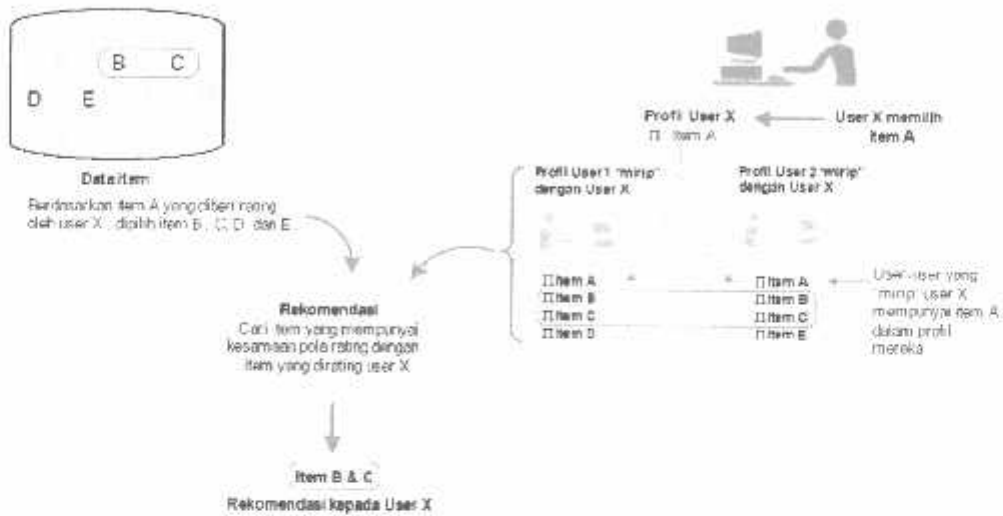
Ada perbedaan antara rekomendasi (dengan seorang pengguna, disarankan item-item yang mungkin menarik bagi pengguna tersebut) dan prediksi (dengan sebuah item, dilakukan prediksi berapa rating yang mungkin diberikan pengguna pada item tersebut). Namun, pada dasarnya untuk memberi rekomendasi kepada pengguna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

Secara umum, berbagai algoritma CF yang ada saat ini dapat dikelompokkan menjadi 2 kategori⁽⁵⁾, yaitu:

1. *Memory-based CF*

Algoritma *memory-based* menggunakan seluruh data rating dalam membuat prediksi. Metode ini memakai teknik statistik dalam mencari

sekumpulan pengguna, disebut dengan *nearest-neighbor*, yang mempunyai kemiripan dengan pengguna target. Berdasarkan kemiripan antar pengguna tersebut, dibuat prediksi dan rekomendasi *Top-N* untuk pengguna target. Metode ini sering disebut juga dengan *nearest-neighbor* atau *user-based collaborative filtering*. Secara lebih jelas, dapat dilihat pada Gambar 2.5 di bawah ini.

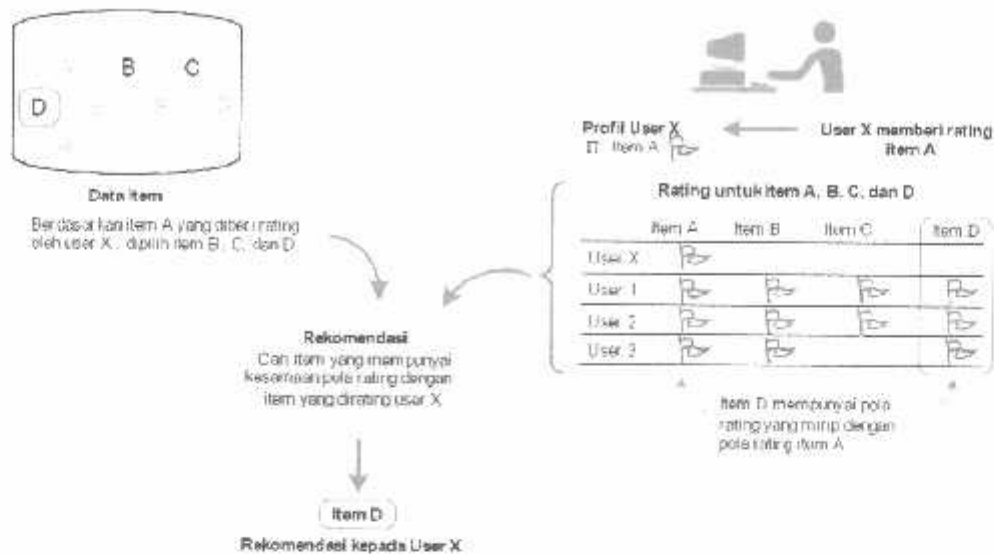


Gambar 2.5 Skema memory-based / user-based

2. Model-based CF

Berbeda dengan memory-based yang mencari hubungan antar pengguna, 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 2.6 di bawah ini.



Gambar 2.6 Skema model-based / item-based

2.4 Algoritma Slope One

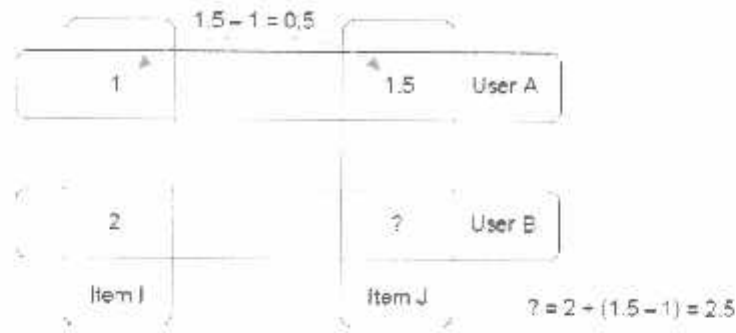
2.4.1 Deskripsi Algoritma Slope One

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⁽¹¹⁾. 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 pengguna target dan item mana yang akan diprediksi. Oleh karena itu, jika pengguna tidak memberi satu rating pun terhadap item, maka prediksi tidak dapat dilakukan.

2.4.2 Pendekatan Algoritma Slope One

Dimisalkan terdapat 2 pengguna, *user A* dan *user B*, dan 2 item, item I dan J. Sebagaimana diperlihatkan pada Gambar 2.6, *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 2.7 Skema Algoritma Slope One⁽¹¹⁾

Secara matematis, pendekatan algoritma Slope One dapat dijelaskan sebagai berikut:

Diberikan rating dari seorang pengguna 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 x adalah kumpulan evaluasi dalam training set. Banyaknya elemen dalam S dinotasikan dengan $card(S)$. Rata-rata rating dalam sebuah evaluasi u dinotasikan dengan \bar{u} , sedangkan $S_i(x)$ adalah set/ kumpulan semua evaluasi $u \in x$ 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:

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

dimana,

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

u_j = rating item j

u_i = rating item i

(x) = training set

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

$card(S_{j,i}(x))$ = banyaknya elemen dalam $S_{j,i}(x)$

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

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

dimana,

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

$c_{j,i} = card(S_{j,i}(x))$ = banyaknya elemen dalam $S_{j,i}(x)$

2.5 Pengukuran Tingkat Akurasi

Tingkat keakurasian rekomendasi yang dihasilkan dapat diukur dengan menggunakan salah satu metode standar dalam statistika yang disebut dengan *Mean Absolut Error* atau MAE⁽⁵⁾. Secara mendasar, MAE menghitung kesalahan / *error absolut* antara rating yang sebenarnya (p) dan rating hasil prediksi (q), dimana semakin kecil nilai MAE yang didapat, maka prediksi yang dihasilkan semakin akurat. Jika pengukuran dilakukan terhadap N data, maka MAE dapat dirumuskan dengan:

$$MAE = \sum_{t=1}^N \frac{|p_t - q_t|}{N} \dots\dots\dots (3)$$

2.6 MySQL

MySQL merupakan sebuah Sistem Manajemen Basisdata. Basisdata ialah sebuah sekumpulan data yang terstruktur. Dapat berupa apa saja, baik daftar belanja hingga sebuah galeri foto atau informasi jaringan perusahaan yang sangat banyak jumlahnya. Untuk menambah, mengakses dan memproses data yang disimpan di dalam sebuah database, diperlukan sebuah Sistem Manajemen Basisdata layaknya MySQL server⁽¹²⁾.

MySQL merupakan sebuah Sistem Manajemen Basisdata Relasional (*Relational DataBase Management System*) atau RDBMS yang memiliki performa tinggi, beroperasi secara *multi-thread*, *multi-user* yang dibangun berdasarkan arsitektur klien-server⁽¹³⁾.

Popularitas MySQL sangat tinggi dikarenakan MySQL memiliki kelebihan antara lain: kecepatan, kehandalan, keleluasaan, dan kode *open-source*.

Dalam sebuah RDBMS, kecepatan – waktu yang diperlukan oleh sebuah query untuk mengeksekusi dan menampilkan hasil – adalah segalanya. *Benchmarks* dari MySQL yang tersedia di situs MySQL menunjukkan bahwa MySQL melampaui hampir semua pesaing sistem basisdata yang sekarang ini ada⁽¹⁴⁾.

Pada umumnya, sebuah basisdata berperforma tinggi kurang handal, namun tidaklah demikian untuk MySQL. MySQL didesain untuk memberikan kehandalan maksimal dan dapat beroperasi dalam jangka waktu yang lama.

MySQL mendukung hampir seluruh fitur penting dari standar ANSI-SQL, dan juga menambahkan *custom-extension*, *functions*, dan *data-types* yang didesain untuk meningkatkan portabilitas dan menyediakan pengguna dengan fungsionalitas yang lebih baik.

Kode sumber MySQL bebas tersedia dibawah lisensi GNU General Public License. Lisensi ini yang membuat MySQL berkembang pesat karena peran aktif komunitas MySQL yang menjadikan MySQL mampu bersaing dengan para competitor.

Selain itu MySQL juga memiliki fitur seperti :

- Skalabilitas
- Mudah penggunaannya
- *Multiuser*
- *Internationalization*
- Dukungan aplikasi yang luas

2.7 PHP

PHP merupakan akronim rekursif dari *PHP: Hypertext Preprocessor*, yang digunakan luas sebagai *general-purpose scripting language* yang berkode sumber terbuka (*open-source*) yang pada khususnya digunakan untuk pengembangan web dan dapat dilekatkan pada HTML⁽¹⁵⁾.

Dalam beberapa tahun terakhir ini PHP, menjadi standar *de facto* bagi pengembangan aplikasi web yang berbasis data, karena dilaporkan memiliki

skalabilitas, mudah penggunaannya dan dukungan luas akan basisdata dan format data (16).

Performa PHP sebagai *server-side language*, menjadi semakin cepat semenjak PHP 5.0 didesain ulang dengan mengoptimalkan alokasi memori untuk meningkatkan performa dan peningkatan kecepatan yang signifikan dari versi sebelumnya. Sebagai tambahan, *accelerator* dari pihak ketiga tersedia guna meningkatkan waktu respon dan performa.

Portabilitas PHP sebagai bahasa pemrograman tersedia di berbagai sistem operasi, baik UNIX, Mac OS, OS/2, dan Microsoft Windows selain itu PHP juga portabel antar *platform*.

Kemudahan penggunaan PHP dikarenakan *syntax* yang jelas dan konsisten, dengan 5000 lebih dokumentasi yang menyertainya. Hal ini dapat sedikit mempermudah para pengguna baik pemula maupun tingkat lanjut untuk mempelajarinya.

PHP merupakan sebuah proyek *open-source*, bahasa pemrograman ini dikembangkan secara luas oleh tim sukarelawan yang menjadikan kode sumbernya dapat diakses secara gratis di web.

Salah satu hal menarik dari sebuah bahasa pemrograman yang didukung oleh komunitas yang luas adalah terdapat banyak kreatifitas dan imajinasi dari para pengembang di seluruh dunia. Luasnya PHP dapat dibuktikan di PEAR, *PHP Extension and Application Repository* (<http://pear.php.net/>) dan PECL, *PHP Extension Community Library* (<http://pecl.php.net/>).

Saat ini PHP 5.3 didukung oleh lebih dari 15 Sistem Manajemen Basisdata termasuk MySQL, PostgreSQL, Oracle dan Microsoft SQL Server.

BAB III

METODE DAN PERANCANGAN SISTEM

3.1 Perancangan Collaborative-Filtering

3.1.1 Pembuatan Model Korelasi Antar-Item

Pembuatan model item dilakukan secara *offline* dengan memasangkan masing-masing item dari data rating dan menghitung rata-rata selisih rating setiap pasangan item tersebut.

Algoritma pembuatan model korelasi antar item adalah:

1. Dari data rating, dicari para pengguna (*user*) u yang melakukan rating.
2. Untuk setiap pengguna (*user*), didapatkan item-item apa saja yang dirating.
3. Untuk setiap item, dipasangkan satu sama lain dan dilakukan perhitungan:

$$\text{JUMLAH RATING} = \text{Jumlah Rating} + 1$$

$$\text{SELISIH RATING} = \text{Selisih Rating}_{(n-1)} + (\text{Selisih Rating}_{(n)})$$

Misalkan diketahui data rating yang diperoleh dari pengguna, dengan menggunakan skala rating 1 – 5 ditunjukkan dalam tabel.

Tabel 3.1 Contoh data rating dari User

User_ID	Item_ID	Rating
1	A	5
1	B	3
1	C	4
1	D	2
2	A	5
2	B	2
2	D	3
2	E	4
3	A	4
3	B	3

Maka, langkah-langkah yang dilakukan untuk membuat model item adalah sebagai berikut:

1. Untuk setiap *user*, dicari item apa saja yang telah dirating.

Tabel 3.2 Data Rating dari User 1

User_ID	Item_ID	Rating
1	A	5
1	B	3
1	C	4
1	D	2

2. Masing-masing item yang dirating dipasangkan satu sama lainnya membentuk sebuah matriks item ke item. Matriks ini dapat direpresentasikan ke dalam bentuk sebuah tabel korelasi item ke item yang berisi selisih rating dan jumlah rating yang didapat setiap pasangan item.

Untuk pasangan $A \rightarrow B$ (item_ID1 \rightarrow item_ID2) akan didapat:

JUMLAH RATING = 1 (sementara dari user 1)

SELISIH RATING A ke B = 3 - 5 = -2

Sehingga data korelasi yang dihasilkan adalah:

Tabel 3.3 Data Korelasi Sementara

item_ID1	Item_ID2	Jumlah Rating	Selisih Rating
A	B	1	-2
A	C	1	-1
A	D	1	-3
B	A	1	+2
B	C	1	+1
B	D	1	-1
C	A	1	+1
C	B	1	-1
C	D	1	-2
D	A	1	+3
D	B	1	+1
D	C	1	+2

3. Dengan cara yang sama, untuk *user 2* dan *user 3* juga dilakukan proses update pada tabel model item di atas. Berdasarkan Tabel 3.1, diketahui rating yang diberikan oleh *user 2* seperti ditunjukkan pada Tabel 3.4.

Tabel 3.4 Data rating dari user 2

User_ID	Item_ID	Rating
2	A	5
2	B	2
2	D	3
2	E	4

Maka korelasi untuk $A \rightarrow B$ akan menjadi:

$$\text{JUMLAH RATING} = 1 + 1 = 2$$

$$\text{SELISIH RATING A ke B} = -2 + (2-5) = -5$$

Sedangkan untuk *user 3* yang hanya memberi 2 buah rating masing-masing bernilai 4 untuk item A dan 3 untuk item B, maka korelasi $A \rightarrow B$ menjadi:

$$\text{JUMLAH RATING} = 2 + 1 = 3$$

$$\text{SELISIH RATING A ke B} = -5 + (3 - 4) = -6$$

Sehingga setelah korelasi setiap pasangan dihitung, secara keseluruhan tabel model item akan mengalami perubahan menjadi:

Tabel 3.5 Data korelasi antar item setelah di-update

item_ID1	Item_ID2	Jumlah Rating	Selisih Rating
A	B	3	-6
A	C	1	-1
A	D	2	-5
B	A	3	+6
B	C	2	+1
B	D	2	-1
C	A	1	+1
C	B	2	-1
C	D	1	-2
D	A	2	+5
D	B	2	+1
D	C	1	+2
A	E	2	-1
E	A	2	+1
B	E	1	+2
E	B	1	-2
D	E	1	+1
E	D	1	-1

3.1.2 Rekomendasi *Non-Personalized*

Rekomendasi *non-personalized* adalah rekomendasi yang dibuat tidak berdasarkan atas profil atau preferensi *user* dalam hal merating item pada masa lampau. Kegunaan rekomendasi *non-personalized* adalah untuk memberi alternatif bagi *user* yang tidak terdaftar (selanjutnya disebut sebagai *visitor*) untuk tetap bisa mendapatkan rekomendasi yang akurat. Pendekatan seperti ini dapat diilustrasikan dengan: “pengguna yang menyukai item I juga menyukai item J dan K”.

Untuk dapat memberikan rekomendasi kepada *visitor*, metode yang digunakan adalah dengan membagi selisih rating setiap pasangan-pasangan item dengan jumlah ratingnya. Sebuah nilai *threshold* jumlah rating dapat diberlakukan untuk membatasi jumlah rating minimum yang harus dipenuhi setiap pasangan item untuk dapat ditampilkan sebagai rekomendasi.

Misalkan seorang *visitor* sedang menampilkan item A, maka dicari item-item lain (item_ID2) yang berkorelasi dengan item A. Berdasarkan Tabel 3.3, jika ditentukan minimal jumlah rating > 1 , maka pasangan item yang memenuhi syarat adalah:

Tabel 3.6 Data model item dengan jumlah rating > 1

item_ID1	Item_ID2	Jumlah Rating	Selisih Rating
A	B	3	-6
A	D	2	-5

Rangking item yang ditampilkan diurutkan berdasarkan nilai absolut selisih rating dibagi jumlah ratingnya, dinotasikan dengan $dev_{i,j}$. Dengan kata lain, semakin kecil selisih rating antara 2 item, maka semakin erat korelasi kedua item tersebut. Perhitungan korelasi pasangan $A \rightarrow B$ dan $A \rightarrow D$ adalah sebagai berikut:

Korelasi $A \rightarrow B$

$$dev_{B,A} = \sum \frac{u_B - u_A}{card(S_{B,A}(x))} = \frac{|-6|}{3} = 2$$

Korelasi $A \rightarrow D$

$$dev_{D,A} = \sum \frac{u_D - u_A}{card(S_{D,A}(x))} = \frac{|-5|}{2} = 2,5$$

Sehingga jika dirangking dengan nilai paling rapat atau rendah, rekomendasi *non-personalized* yang diberikan kepada *visitor* ditunjukkan pada tabel.

Tabel 3.7 Rekomendasi *non-personalized* bagi *visitor*

Ranking	Item_ID	$dev_{i,j}$
1	B	2
2	D	2,5

3.1.3 Rekomendasi *personalized*

Rekomendasi *personalized* diberikan kepada pengguna (*user*) yang terdaftar dalam sistem. Berbeda dengan rekomendasi *non-personalized*, rekomendasi *personalized* mengikutsertakan rating-rating yang diberikan pengguna terhadap item pada masa lampau untuk melakukan prediksi bagaimana seorang pengguna menyukai sebuah item yang diberikan. Setelah diketahui korelasi antar item dari model korelasi yang telah terbentuk, dapat dilakukan perhitungan prediksi untuk selanjutnya dari hasil prediksi tersebut dibuat sebagai daftar rekomendasi bagi pengguna target.

3.1.3.1 Prediksi per Item

Prediksi per item dilakukan terhadap sebuah item berdasarkan rating dari item lain yang berkorelasi dengan item yang akan diprediksi. Sesuai dengan persamaan (2), langkah-langkah yang dilakukan untuk menghitung prediksi rating item_ID2 bagi seorang pengguna adalah sebagai berikut:

1. Mencari item-item yang telah dirating oleh pengguna target (item_ID1) selain item yang sedang dilihat atau yang sedang ditampilkan (item_ID2).
2. Untuk setiap item item_ID1 yang mempunyai korelasi dengan item_ID2, dihitung rata-rata selisih ratingnya dengan item_ID2 dengan cara membagi selisih rating dengan jumlah ratingnya.
3. Melakukan prediksi rating item_ID2 berdasarkan persamaan (2).

Melanjutkan penjelasan sebelumnya, misalnya akan dilakukan prediksi berapa nilai rating item C bagi pengguna 2. Sebelum dilakukan prediksi, dibutuhkan perhitungan selisih item C dengan item-item yang telah dirating oleh pengguna (*user*) 2, yakni item A, B, D, dan E. Oleh karena item E tidak mempunyai korelasi dengan item C, maka item E tidak diikutsertakan dalam perhitungan. Berdasarkan dari data korelasi

antar item pada Tabel 3.5, dapat diketahui korelasi item C dengan item yang telah dirating seperti pada tabel di bawah ini.

Tabel 3.8 Data korelasi antar item yang memuat item_ID2 = C

item_ID1	Item_ID2	Jumlah Rating	Selisih Rating
A	C	1	-1
B	C	2	+1
D	C	1	+2

Maka, berdasarkan persamaan (1) besarnya rata-rata selisih rating item C dengan item A, B dan D adalah:

$$dev_{C,A} = \sum \frac{u_C - u_A}{card(S_{C,A}(x))} = \frac{-1}{1} = -1$$

$$dev_{C,B} = \sum \frac{u_C - u_B}{card(S_{C,B}(x))} = \frac{1}{2} = 0,5$$

$$dev_{C,D} = \sum \frac{u_C - u_D}{card(S_{C,D}(x))} = \frac{2}{1} = 2$$

Setelah diketahui rata-rata selisih ratingnya, maka prediksi untuk item C dapat diketahui berdasarkan persamaan (2):

$$P^{sl}(u)_C = \frac{\sum_{i \in S(u)-\{C\}} (dev_{C,i} + u_i) c_{C,i}}{\sum_{i \in S(u)-\{C\}} c_{C,i}}$$

Dengan I adalah item A, B dan D, maka diperoleh

$$P^{sl}(u)_C = \frac{(-1+5)1 + (0,5+2)2 + (2+3)1}{1+2+1} = \frac{14}{4} = 3,5$$

Prediksi rating item C kepada *user* 2 adalah 4 (pembulatan).

3.1.3.2 Rekomendasi Top-N

Prediksi terhadap item-item dapat pula diberikan sebagai bentuk rekomendasi. Langkah yang dilakukan hampir sama dengan langkah-langkah dalam memprediksi rating sebuah item. Bedanya, prediksi dilakukan terhadap banyak item sekaligus dan hasilnya diurutkan dari prediksi rating yang terbesar hingga yang terkecil. Prediksi semacam ini disebut juga dengan rekomendasi Top-N.

Masih terkait dengan penjelasan sebelumnya, maka rekomendasi Top-N untuk *user* 3 dapat dijabarkan dalam langkah-langkah di bawah ini:

1. Berdasarkan rating *user* 3 dan data model item pada Tabel 3.4, diketahui item-item yang berkorelasi dengan item A dan B adalah item C, D, dan E, maka rata-rata selisih rating setiap pasangan diperoleh:

$$dev_{C,A} = \sum \frac{u_C - u_A}{card(S_{C,A}(x))} = \frac{-1}{1} = -1$$

$$dev_{D,A} = \sum \frac{u_D - u_A}{card(S_{D,A}(x))} = \frac{-5}{1} = -2,5$$

$$dev_{E,A} = \sum \frac{u_E - u_A}{card(S_{E,A}(x))} = \frac{-1}{1} = -1$$

$$dev_{C,B} = \sum \frac{u_C - u_B}{card(S_{C,B}(x))} = \frac{1}{1} = 1$$

$$dev_{D,B} = \sum \frac{u_D - u_B}{card(S_{D,B}(x))} = \frac{-1}{1} = -1$$

$$dev_{E,B} = \sum \frac{u_E - u_B}{card(S_{E,B}(x))} = \frac{2}{1} = 2$$

2. Perhitungan prediksi rating untuk masing-masing item C, D, dan E bagi *user* 3 adalah sebagai berikut.

$$P^{SI}(u)_C = \frac{(-1+4)1+(1+3)1}{1+1} = \frac{7}{2} = 3,5$$

$$P^{SI}(u)_D = \frac{(-2,5+4)2+(-1+3)1}{2+1} = \frac{5}{3} = 1,66$$

$$P^{SI}(u)_E = \frac{(-1+4)1+(2+3)1}{1+1} = \frac{8}{2} = 4$$

3. Rekomendasi Top-N diberikan kepada *user* 3 berupa daftar item yang telah dirangking berdasarkan urutan prediksi nilai rating item C, D, dan E dari yang terbesar.

Tabel 3.9 Rekomendasi *personalized* bagi user 3

Ranking	Item_ID	Prediksi Rating
1	E	4
2	C	3,5
3	D	1,66

3.2 Perancangan Tabel

3.2.1 Tabel user

Tabel `rss_user` menyimpan data identitas setiap pengguna (*user*). Dalam tabel ini disimpan data kode *user*, password dan email *user*.

Tabel 3.10 Atribut tabel `rss_user`

Field	Type	Null
<u>user_id</u>	int(11)	No
user_name	varchar(20)	No
password	blob	No
email	varchar(255)	No

Keterangan:

user_id : Kode user

user_name : Nama user

password : Password

email : Email user

3.2.2 Tabel item

Tabel `item` menyimpan data dvd games yang terbagi ke dalam kategori yang berbeda. Tabel `item` dilengkapi beberapa field yang diperlukan sebagai parameter untuk tampilan melalui `php`.

Tabel 3.11 Atribut tabel rss_catalog_items

Field	Type	Null	Default
item_id	int(11)	No	
rank	int(11)	Yes	<i>NULL</i>
is_enabled	tinyint(4)	Yes	<i>NULL</i>
title	varchar(255)	Yes	<i>NULL</i>
comment	text	Yes	<i>NULL</i>
description	text	Yes	<i>NULL</i>
updated_at	varchar(255)	Yes	<i>NULL</i>
categ_id	int(11)	Yes	<i>NULL</i>
file	varchar(255)	Yes	<i>NULL</i>
file_thumb	varchar(255)	Yes	<i>NULL</i>
price	int(11)	Yes	<i>NULL</i>

Keterangan:

- id : Kode item
- rank : Ranking urutan item (belum dipergunakan)
- is_enabled : meng(non)aktifkan suatu item.
- title : Judul item.
- comment : Komentar singkat tentang item
- description : Deskripsi tentang item
- updated_at : Waktu item masuk
- categ_id : id kategori untuk item
- file : nama file gambar (jpg)
- file_thumb : nama file *thumbnail*
- price : harga item

3.2.3 Tabel rating

Tabel *rss_user_ratings* merupakan komponen dasar dalam sistem *Recommender System* yang berbasis *collaborative filtering* yang menyimpan data rating yang diberikan user terhadap item-item. Tabel *rss_user_ratings* sangat diperlukan dalam pembuatan model item sebagai dasar perhitungan prediksi.

Tabel 3.12 Atribut tabel *rss_user_ratings*

Field	Type	Null	Default
<i>user_id</i>	int(11)	Yes	<i>NULL</i>
<i>item_id</i>	int(11)	Yes	<i>NULL</i>
<i>rating</i>	int(5)	Yes	<i>NULL</i>

Keterangan:

user_id : kode user yang memberi rating

item_id : kode item yang diberi rating

rating : nilai rating

Untuk menambah kecepatan *query* maka kolom *user_id* dan *item_id* diberi index.

3.2.4 Tabel Slope One

Tabel *rss_slope_one* digunakan untuk menyimpan data model item yang telah dibuat secara offline, dan kemudian nantinya akan *di-update*, tiap kali pengguna (*user*) merating. Pada tabel *rss_slope_one* terdapat kolom *times* dan *rating* untuk dipergunakan dalam perhitungan prediksi. Sebagaimana dijelaskan pada bab sebelumnya, tabel *rss_slope_one* dapat *di-update* secara langsung, melalui pemanggilan sebuah *stored procedure*.

Tabel 3.13 Atribut tabel *rss_slope_one*

Field	Type	Null
item_id1	int(11)	No
item_id2	int(11)	No
times	int(11)	No
rating	decimal(14,4)	No

Keterangan:

item_id1 : kode item yang sedang dilihat.

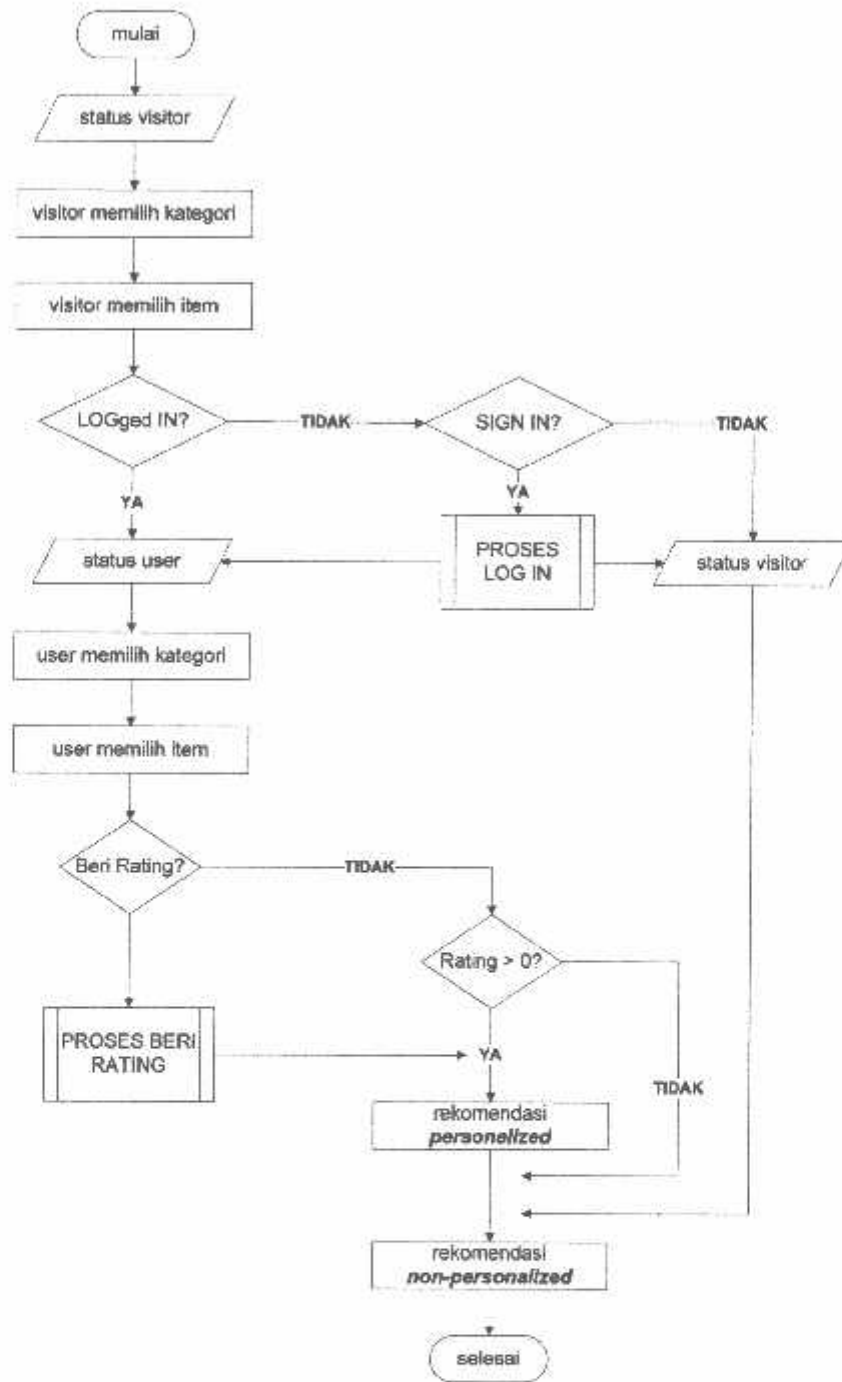
item_id2 : kode item yang dipasangkan / diperbandingkan.

times : counter / penghitung jumlah pemasangan antar item

rating : selisih rating antara item_id1 dan item_id2.

3.3 Alur Kerja Recommender System

Berikut ini merupakan alur kerja utama dari sistem dalam bentuk diagram alir:



Gambar 3.1 Alur Kerja Recommender Systems

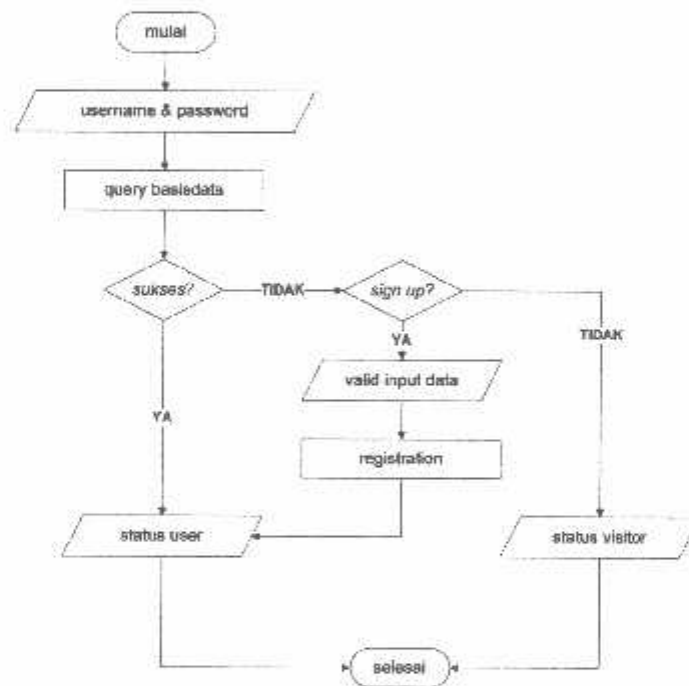
Pada saat pertama kali, status *visitor* diberikan kepada pengguna (*user*) yang belum mendaftar. *Visitor* tersebut dapat memilih (*browsing*) item dalam sebuah kategori melalui katalog, kemudian diperiksa apakah *visitor* tersebut telah terdaftar atau belum.

Jika *visitor* telah terdaftar, maka status *visitor* menjadi pengguna (*user*) yang dapat memberikan rating kepada item yang dia sukai melalui proses beri rating. Ketika pengguna (*user*) memutuskan untuk tidak merating maka diperiksa apakah *user* telah memberi paling tidak sebuah rating pada sebuah item. Jika kondisi benar, *user* akan diberi rekomendasi *personalized*. Namun jika belum, *user* akan diberi rekomendasi *non-personalized*.

Bagi pengguna (*user*) yang tidak terdaftar atau tamu (*visitor*) , dapat mendaftarkan dirinya melalui proses login, sukses atau tidaknya proses tersebut akan menentukan status *user* tersebut, menjadi *user* atau tetap menjadi *visitor*. *Visitor* akan selalu diberikan rekomendasi *non-personalized*.

3.3.2 Proses Login

Berikut ini merupakan alur kerja dari proses login yang sederhana:

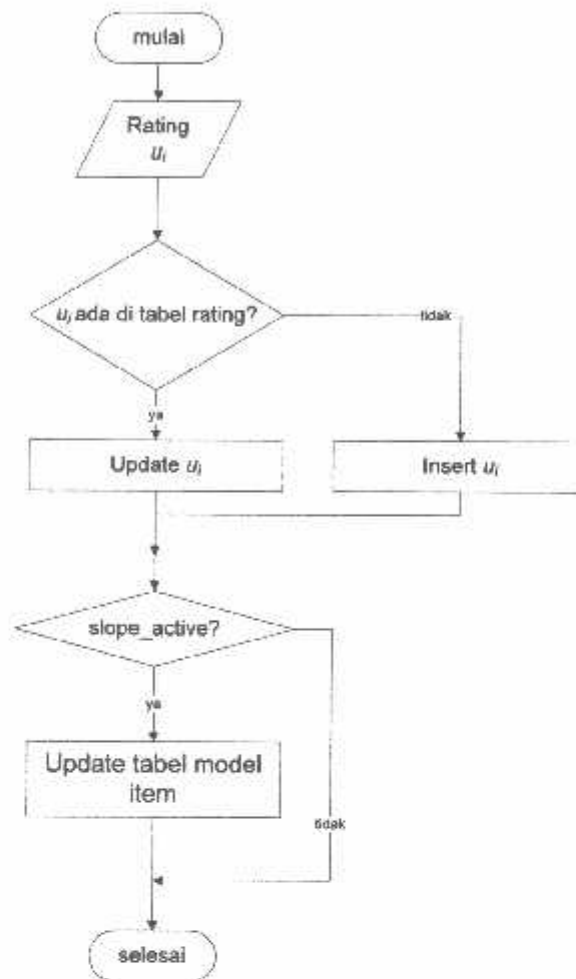


Gambar 3.2 Alur kerja proses login

Proses ini menentukan apakah *user* tersebut tetap menjadi *visitor* hingga hanya diberikan rekomendasi *non-personalized* atau berubah menjadi *user*.

3.3.3 Proses Pemberian Rating

Berikut adalah diagram alir dari alur kerja Beri Rating:



Gambar 3.3 Alur kerja proses pemberian rating

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. Apabila *slope_active* bernilai 1 (TRUE) maka sebuah *stored procedure* untuk meng-*update* tabel *rss_slope_one* akan dijalankan setiap kali seorang *user* member rating terhadap sebuah item, data model item ter-*update* secara otomatis.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

4.1.1 Persiapan Data

Untuk mengetahui bagaimana kinerja *Recommender Systems* (RSs), dibutuhkan sebuah *dataset* rating yang telah terisi dengan *record* rating item dari *user*. Dataset ini berisi 96 *user* dari total 109 *user*, 204 item dari 932 item, dan 1.445 data rating dengan skala rating antar 1 sampai 5.

Tabel 4.1 Statistik dataset

Jumlah user	96
Jumlah item	204
Banyaknya rating	1.445
Rata-rata rating per item	3,97
Rata-rata banyaknya rating per user	15,05
Rata-rata banyaknya rating per item	7,09

4.1.2 Implementasi Basis Data

Tabel utama yang dirancang untuk pembuatan rekomendasi berbasis metode *item-based* ada 2, yaitu tabel *rss_user_ratings* dan *rss_slope_one*. Tabel *rss_user_ratings* menyimpan data rating item dari pengguna, sedangkan tabel *rss_slope_one* dirancang untuk menyimpan data korelasi antar item atau data model item yang nantinya akan digunakan dalam perhitungan prediksi.

Tabel *rss_user_ratings* adalah tabel utama dalam proses *collaborative filtering* dan digunakan sebagai dasar pembuatan model item pada metode *item-based*. Cuplikan kode

dalam bentuk query di bawah ini bertujuan membuat tabel `rss_ratings` beserta *field-field* yang dibutuhkan.

```
CREATE TABLE IF NOT EXISTS `rss_user_ratings` (
  `user_id` int(11) DEFAULT NULL,
  `id` int(11) DEFAULT NULL,
  `rating` int(5) DEFAULT NULL,
  KEY `user_id` (`user_id`),
  KEY `item_id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Berdasarkan skema Slope One yang sudah dibahas, maka dalam tabel model item dirancang 2 field, yakni `diff_slope` dan `cnt_slope`. Field `diff_slope` menyimpan besarnya selisih rating setiap pasangan item, sedangkan `cnt_slope` digunakan untuk menyimpan banyaknya rating yang didapat pasangan item tersebut. Dengan demikian, nilai korelasi antara 2 item yang berpasangan dapat diketahui dengan membagi `diff_slope` dengan `cnt_slope`. Nilai korelasi yang dihasilkan adalah berupa rata-rata selisih rating. Query untuk membuat tabel `rss_slope_one` adalah sebagai berikut.

```
CREATE TABLE IF NOT EXISTS `rss_slope_one` (
  `item_id1` int(11) NOT NULL,
  `item_id2` int(11) NOT NULL,
  `times` int(11) NOT NULL,
  `rating` decimal(14,4) NOT NULL,
  KEY `item_id1` (`item_id1`),
  KEY `item_id2` (`item_id2`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

4.1.3 Deskripsi Program

4.1.3.1 Pembuatan Model Item

Model item dibuat berdasarkan data rating item dalam tabel `rating`. Untuk tujuan tersebut dibuat sebuah *stored procedure* untuk membuat tabel `rss_slope_one`, sebagai berikut :

```
CREATE PROCEDURE `sp_rss_slope_one`()
begin
  DECLARE tmp_item_id int;
  DECLARE done int default 0;
```

```

        DECLARE mycursor CURSOR FOR select distinct id
from rss_user_ratings;
        DECLARE CONTINUE HANDLER FOR NOT FOUND set
done=1;

        open mycursor;
        while (!done) do
            fetch mycursor into tmp_item_id;
            if (!done) then
                insert into rss_slope_one (select a.id
as item_id1,b.id as item_id2,count(*) as times, sum(b.rating-
a.rating) as rating from rss_user_ratings a,rss_user_ratings b where
a.id = tmp_item_id and b.id != a.id and a.user_id=b.user_id group by
a.id,b.id);
            end if;
        END while;
        close mycursor;
    end
//
DELIMITER //

```

Sebuah CURSOR digunakan untuk menyimpan sementara query hasil dari memilih item secara unik (DISTINCT):

```

DECLARE mycursor CURSOR FOR select distinct id from
rss_user_ratings;

```

Sebuah looping while hingga done bernilai 1, memasukkan item berdasarkan rata-rata selisih rating item 1 terhadap item 2 ($dev_{j,i}$) seperti persamaan (1).

4.1.3.2 Pemberian Rating

Pemberian rating adalah salah satu proses utama dalam sistem sebagai bentuk interaksi antara pengguna dengan sistem. Pada proses ini pengguna menyimpan data rating ke dalam tabel rss_user_rating, fungsi PHP sebagai berikut:

```

function rate($db, $u_id, $id){
    //$rating = $_GET['rating'];
    $text = strip_tags($_GET['rating']);
    $update = "UPDATE `rss_user_ratings` SET `rating` =
" . $_GET['rating'] . " WHERE `user_id` = " . $u_id . " AND `id` =
" . $id . ";";

    //$result = @mysql_query($update);
    $result = $db->query($update);
    //if(mysql_affected_rows() == 0){
    if ($result){

```

```

        $insert = "INSERT INTO `rss_user_ratings`
        (`user_id`,`id`,`rating`) VALUES (".$u_id.", ".$id.",
        ".$_GET['rating'].");";

        // $result = @mysql_query($insert);
        $result = $db->query($insert);
    }
}

```

Sebuah Struktur kontrol digunakan apabila pengguna (*user*) telah merating item tersebut maka proses hanya akan meng-*update* tabel tersebut.

4.1.3.3 Update Model Item

Proses ini bertujuan agar tabel model item selalu *ter-update* secara otomatis. Proses ini dilakukan dengan sebuah *constructor* pada class *SlopeOne* sehingga proses ini langsung dijalankan ketika sebuah *instance* dari *SlopeOne* dijalankan.

```

function __construct($db, $active/*, $categ_on*/){
    $this->db = $db;
    $this->active = $active;
    // $this->categ_on = $categ_on;
    if ($this->active == 1){
        $this->truncateSlopeOne();
        $this->callSlopeOne(/*$this->categ_on*/);
    }
}

```

Fungsi *truncateSlopeOne*, menghapus seluruh isi tabel *rss_slope_one*:

```

function truncateSlopeOne(){
    $sql = "TRUNCATE TABLE `rss_slope_one`";
    $this->db->query($sql);
}

```

Fungsi *callSlopeOne*, yang berfungsi untuk memanggil *stored procedure* *rss_sp_slope_one*:

```

function callSlopeOne(/*$categ_on*/){
    $this->procedure_call = "CALL `sp_rss_slope_one`";
    $this->db->prepareProcedure($this->procedure_call);
}

```

4.1.3.4 Pemberian Rekomendasi *non-personalized*

Fungsi dibawah ini memberikan rekomendasi berdasarkan korelasi antar item kepada pengguna yang belum terdaftar atau belum memberikan rating, dengan sebuah rating min dan batas item yang ditampilkan

```
function non personalizedRecomendation($db, $item_id,
$rat_min, $limitNon){
    $this->sql = "SELECT item_id2 AS item FROM rss_slope_one
WHERE item_id1= ? AND times >= ? GROUP BY item_id2 ORDER BY
SUM(ABS(rating/-times)) DESC LIMIT ? ;";
    $this->item_id = $item_id;
    $this->rat_min = $rat_min;
    $this->limitNon = $limitNon;
    $this->db->prepareStatement($this->sql, $this->item_id,
$this->rat_min, $this->limitNon);
    $value = array();
    foreach ($this->db->bind_array as $line){
        $value[] = $line;
    }
    return $value;
}
}
```

Dimana \$rat_min adalah jumlah banyak rating, dan \$limitNon merupakan batas item yang ingin ditampilkan.

4.1.3.5 Pemberian Rekomendasi *personalized*

Fungsi dibawah ini memberikan rekomendasi berdasarkan korelasi antar item kepada pengguna yang telah terdaftar dan sudah memberikan rating, dengan sebuah batas item yang ditampilkan.

```
function personalizedRecomendation($db, $user_id,
$limitTopN) {
    $this->sql = "SELECT s.item_id2 AS item FROM
rss_slope_one s,rss_user_ratings u WHERE u.user_id = ? AND
s.item_id1 = u.id AND s.item_id2 != ALL (SELECT u.id FROM
rss_user_ratings u WHERE u.user_id = ?) GROUP BY s.item_id2 ORDER BY
SUM(u.rating * s.times - s.rating)/SUM(s.times) DESC LIMIT ? ;";
    $this->user_id = $user_id;
    $this->user_idDump = $user_id;
    $this->limitTopN = $limitTopN;
    $this->db->prepareStatement($this->sql, $this->user_id,
$this->user_idDump, $this->limitTopN);
    $value = array();
    foreach ($this->db->bind_array as $line){
```

```

        $value[] = $line;
    }
    return $value;
}

```

Dimana `$limitTopN` merupakan batas item yang ingin ditampilkan.

4.2 Penerapan Aplikasi

4.2.1 Pembuatan Model Item

Pembuatan model item dilakukan secara *offline* dan *online* (update) untuk mencari semua pasangan item sehingga dapat digunakan untuk perhitungan prediksi dan pembuatan rekomendasi secara *online*.

Pembuatan awal model item dilakukan dengan menjalankan sebuah *stored procedure* yang menghitung selisih rating dan jumlah rating masing-masing pasangan item dari tabel `rss_user_rating`, kemudian menyimpannya dalam tabel `rss_slope_one`.

Spesifikasi komputer sebagai alat uji untuk memproses pembuatan model secara *offline* adalah sebagai berikut:

- Processor : AMD Athlon® 64 X2 4200+ (2 CPUs), ~2.2GHz
- Memory : 2048MB RAM
- Sistem Operasi : Windows 7 Ultimate 32-bit
- Web Server : Apache 2.2.14
- Database Server : MySQL version: 5.1.41

Spesifikasi server *shared-hosting* sebagai alat uji untuk memproses pembuatan model secara *online* adalah sebagai berikut:

- Processor : Intel® Core i7® 920 Quad Core
- Memory : 8 GB DDR3 ECC RAM
- Server : UNIX
- Web Server : Apache & Nginx
- Database Server : MySQL version: 5.0.91

Tabel 4.2 Waktu komputasi pembuatan model item

Server	Banyaknya Item	Banyak Korelasi	Waktu Komputasi
<i>Offline</i>	204	9.178	0,68 dtk
<i>Online</i>	204	9.178	0,1613 dtk

4.2.2 Web *e-commerce* penjualan dvd games

Sebuah web *e-commerce* penjualan dvd games yang sederhana dengan manajemen pengguna dibuat bertujuan untuk melakukan simulasi *Recommender Systems*.

4.2.3 Katalog item

Katalog item menampilkan daftar item yang terurut berdasarkan kategori dan abjad, dengan script sebagai berikut.

```
foreach($html['items'] as $item) {?>
<h1><?php echo $item["title"]?></h1>
<p>
    <a href="?id=<?php echo $item["id"]?>" >" alt="<?php echo $item["title"]?>"
class="float-left"/></a>

    <?php echo $item["comment"]?>
</p>
```



Gambar 4.1 Tampilan katalog berdasarkan kategori

Apabila pengguna telah terdaftar (*logged-in*) atau baru saja mendaftar

```
if (isset($_POST['username']) && isset($_POST['password'])) {
    //user logged-in
    $username = $_POST['username'];
    $password = $_POST['password'];
    $db_prefix = $config['db_prefix'];
```

```

$key = $config['key'];

    $sql = "SELECT * FROM `".$db_prefix."users` WHERE user_name
= '".$username.'" AND password =
AES_ENCRYPT('".$password."','".$key."')";
    $result = $db->query($sql);
    foreach($result as $rs){
        $user_id = $rs['user_id'];
    }
    if ($result){
        $_SESSION['rss_valid_user'] = $username;
        $_SESSION['user_id'] = $user_id;
    }
}
}

```

PENDAFTARAN - REGISTRATION

Username:
 Password:
 Ulangi Password:
 Alamat e-mail:

Daftar

[Home](#) [Action](#) [Action-2](#)

USER ACCOUNT

[Belum terdaftar?](#)

[Logout](#)

[Masuk](#)

17

Gambar 4.2 Halaman Pendaftaran dan Login

maka dengan struktur kontrol berikut:

```

<?php if (isset($_SESSION['rss_valid_user'])) {
    $u_id = $_SESSION['user_id'];
}
?>

```

tampilan katalog akan disertai dengan rating bintang. Disini pengguna yang telah terdaftar dapat merating item tersebut. Skala rating angka 1- 5 diberi penjelasan dari "Payah" hingga "Hebat!!".

```

<span class="ratelinks" id="<?php echo $item['id'];?>">
    <span id="<?php echo $u_id;?>">
        <li><a href="javascript:void(0)" title="Payah"
class="one-star">1</a></li>
        <li><a href="javascript:void(0)" title="Kurang"
class="two-stars">2</a></li>
        <li><a href="javascript:void(0)" title="Lumayan"
class="three-stars">3</a></li>
        <li><a href="javascript:void(0)" title="Baik"
class="four-stars">4</a></li>

```



Gambar 4.3 Tampilan katalog pada user yang telah terdaftar

4.2.4 Detil item

Detil item memperlihatkan detail dari sebuah item yang disertai dengan dengan rating bintang (*star-rating*) bagi pengguna yang telah terdaftar. Apabila pengguna tersebut memberikan rating dengan meng-klik bintang, maka secara otomatis akan segera *ter-update* ke dalam database dan rating segera ditampilkan dengan skrip javascript berikut:

```

$('.ratelinks li a').click(function(){
    // get the parent id
    var idUser = $(this).parent().parent().attr('id');
    var idStar =
    $(this).parent().parent().parent().attr('id');
    $.ajax({
        type: "GET",
        url: "update.php",
        data:
        "rating="+$(this).text()+"&do-rate&u_id="+idUser+"&id="+idStar,
        cache: false,
        async: false,
        success: function(result) {
            // remove id #ratelinks element to
            prevent another rate

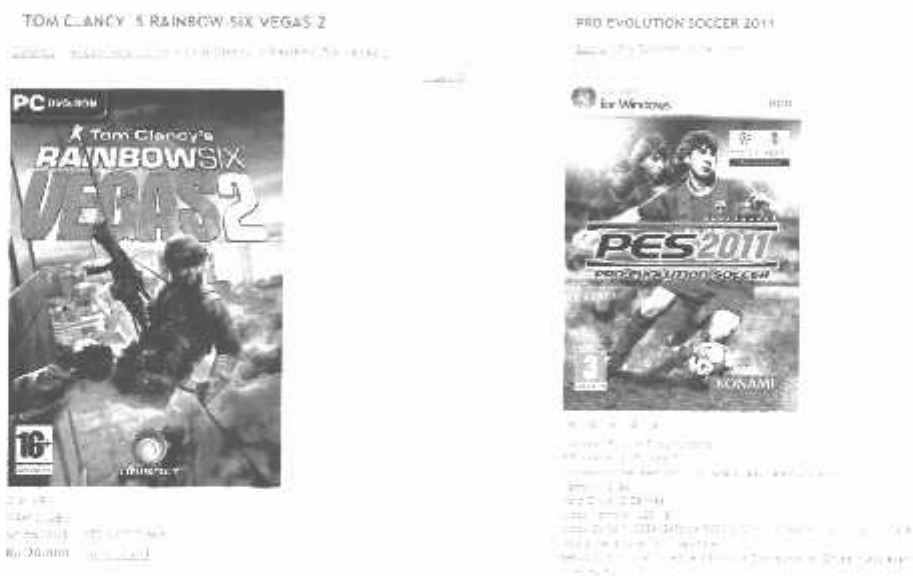
```

```

// class .ratelinks tidak bisa per id ->
hapus per id Bintang dari DOM
$("#"+idStar).remove();
// get rating after click
getRating(idUser,idStar);
},
error: function(result) {
    alert("some error occured, please try
again later");
}
});
});

```

Detail item terdapat fasilitas untuk melihat detail item sebelum ataupun sesudahnya.



Gambar 4.4 Tampilan katalog per-item

4.2.5 *Recommender Systems*

Fasilitas ini bekerja untuk membantu pengguna dalam memilih item yang akan dibeli dengan cara memberi informasi atau sugesti berdasarkan tingkat ketertarikan atau selera masing-masing pengguna.

4.2.5.1 Rekomendasi *non-personalized*

Baik pengguna yang belum mendaftar atau telah mendaftar, diberikan rekomendasi berdasarkan item yang sedang dilihat. Ditampilkan 3 item teratas dari hasil rekomendasi.



Gambar 4.5 Rekomendasi *non-personalized*

Tampilan yang diberikan kepada pengguna, menggunakan cuplikan kode berikut ini:

```
<h2>Rekomendasi <i>Non-Personalized</i> untuk <?php
    $trim_title =
substr_replace($html["item"]['title'],'...',23);
    echo $trim_title;
?></h2>
<p><?php
    $item_recomendation = $slope-
>non_personalizedRecomendation($db, $html["item"]['id'],
$config['rat_min'], $config['limitNon']);
    if ($item_recomendation == null); ?>
<h3>Maaf belum ada rekomendasi untuk item ini</h3>
<?php }
```

```

        for ($i=0; $i<count($item_recomendation);$i++){
            //echo $item_recomendation[$i]."<br>";
            $html["item"] = $catalog->items-
>get_by_id($item_recomendation[$i]);
            //echo $html["item"]["title"];
        ?>
        <h3><?php
            $trim_title =
substr_replace($html["item"]["title'],'...',23);
            echo $trim_title;
        ?></h3>
        <p>
            <a href="?id=<?php echo $html["item"]["id"]?>" >
                " alt="<?php echo
$html["item"]["comment"]?>" class="float-left"/></a>
                <br><br><br><br><br><br>
            </p>
        <?php
        }
    ?>
</p>

```

4.2.5.2 Rekomendasi *personalized*

Rekomendasi ini diberikan hanya kepada pengguna yang telah mendaftar dan telah merating item. Dari hasil rekomendasi hanya ditampilkan 5 item teratas, dengan potongan kode berikut:

```

<h2>Anda mungkin menyukai item ini: </h2>
<p>
<?php
    $item_recomendation = $slope-
>personalizedRecomendation($db, $_SESSION['user_id'],
$config['limitTopN']);
    for ($i=0; $i<count($item_recomendation);$i++){
        //echo $item_recomendation[$i]."<br>";
        $html["item"] = $catalog->items-
>get_by_id($item_recomendation[$i]);
        //echo $html["item"]["title"]."<br>";
    ?>
    <h3><?php
        $trim_title =
substr_replace($html["item"]["title'],'...',23);

```

```

        echo $trim_title;
    ?></div>
</div>
<div data-bbox="183 295 600 520" data-label="Image">


```

Gambar 4.6 Tampilan rekomendasi *personalized*

4.2.6 Shopcart

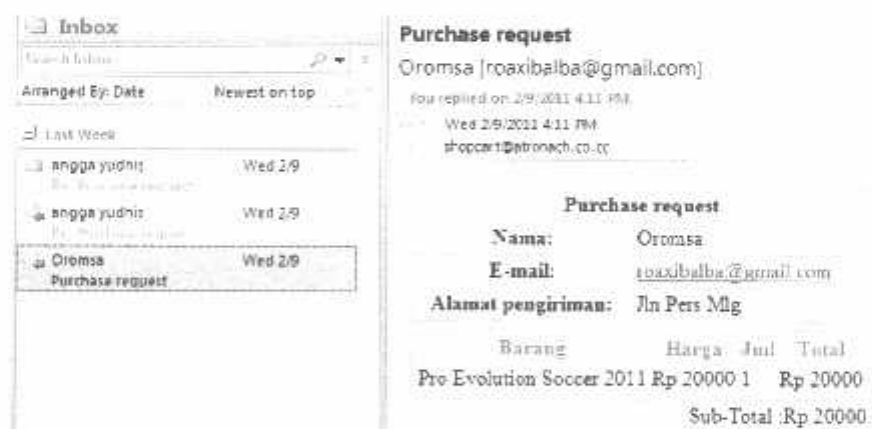
Meskipun web *e-commerce* penjualan dvd games ini hanya bersifat simulasi, pengguna dapat mengekspresikan tingkat kepuasan atau kesukaan dengan membeli item yang diinginkan. Item-item yang telah dibeli tersebut selanjutnya dapat dipergunakan untuk menganalisa dampak dari *Recommender Systems* yang telah berjalan.



Gambar 4.7 Keranjang Belanja

Permintaan belanja, akan ditujukan kepada e-mail yang sudah ditentukan dengan potongan kode berikut:

```
func_send_mail($postget['payer_business_name'],$postget['payer_email'],
'$config["admin_email"]',$subject,$message);
```



Gambar 4.8 Tampilan permintaan pada Microsoft® Outlook

4.3 Analisa Hasil

Jika seorang pengguna belum merating satu pun item dalam sebuah kategori, maka pengguna tersebut tidak memperoleh rekomendasi *personalized*. Hal ini disebabkan karena sistem tidak dapat mengetahui preferensi rating dari pengguna yang

bersangkutan sehingga tidak ada input \$itemID1\$ untuk diikutsertakan dalam perhitungan prediksi pada kategori tersebut, sebagaimana skema Slope One. Sebagai solusinya, pengguna masih dapat menerima rekomendasi berupa daftar item-item yang berkorelasi dengan item yang sedang dilihat, seperti halnya pada rekomendasi *non-personalized*.

Untuk mengukur keakuratan prediksi, digunakan ukuran *Mean Absolut Error* (MAE) dengan melakukan langkah-langkah sebagai berikut:

1. Mengumpulkan 1.000 rating dari pengguna (*user*) (disebut evaluasi u) dari tabel rating sebagai training set X .
2. Membuat model item dari rating dalam training set X .
3. Membuat tabel `rss_analyze_predicted_rating` untuk menyimpan hasil prediksi rating item.

```
CREATE TABLE IF NOT EXISTS `rss_analyze_predicted_rating` (
  `user_id` int(11) DEFAULT NULL,
  `id` int(11) DEFAULT NULL,
  `rating` float DEFAULT NULL,
  KEY `user_id` (`user_id`),
  KEY `item_id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

4. Memilih 100 rating lainnya sebagai test set X' .
5. Melakukan prediksi rating item $P(u)^{(i)}$ untuk semua rating u_i dalam test set X' .
6. Menyimpan hasil prediksi ke dalam tabel `rss_analyze_predicted_rating`.
7. Menghitung rata-rata error antara u_i di tabel rating dengan $P(u)^{(i)}$ hasil prediksi di tabel `rss_analyze_predicted_rating`. Berdasarkan persamaan (3) dengan N adalah banyak rating dalam test set $card(X')$, maka dapat dihitung besarnya *MAE*.

$$\begin{aligned} MAE &= \frac{1}{card(X')} \sum_{i \in S(u)} |P(u)^{(i)} - u_i| \\ &= \frac{1}{100} \sum_{i \in S(u)} |P(u)^{(i)} - u_i| \\ &= 1.742 \end{aligned}$$

Tabel 4.3 Contoh 10 rating hasil prediksi

No	user_id	item_id	u_i	$P(u)^{(i)}$	$ P(u)^{(i)} - u_i $
1	221	890	4	5	1
2	129	807	4	3.556	0.444
3	111	796	3	3	0
4	133	1059	5	5	0
5	215	942	4	4.5	0.5
6	162	186	5	4	1
7	206	1081	4	2	2
8	133	1059	5	5	0
9	175	932	3	1	2
10	111	796	3	3	0

Dari hasil pengujian yang telah dilakukan, besarnya *MAE* yang didapatkan adalah sebesar 1,742, sehingga dapat disimpulkan bahwa rating setiap item yang diprediksi mengalami rata-rata penyimpangan sebesar 1,742 dari rating pengguna (*user*) yang sebenarnya dalam skala rating 1- 5.

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

Tabel 4.4 Hasil uji coba waktu komputasi prediksi secara *online*

Uji Coba	user_ID	Item_ID yang dilihat	Jumlah item yang diprediksi	Waktu Komputasi
1	-	1073	69	0,0012 dtk
2	-	1100	67	0,0007 dtk
3	129	-	137	0,0126 dtk
4	162	-	149	0,0177 dtk
5	126	829	-	0,0006 dtk
6	210	913	-	0,0003 dtk
			RATA-RATA	0,0056 dtk

Percobaan 1 dan 2 merupakan rekomendasi *non-personalized* karena pada dasarnya sistem hanya memberikan rekomendasi berdasarkan item, sedangkan ke-3 dan ke-4 sistem memberikan rekomendasi berdasarkan pengguna yang telah mendaftar atau dikenal dengan rekomendasi *personalized*.

Pada percobaan 5 dan 6 tidak menghasilkan sebuah prediksi (null) dikarenakan minimnya dataset yang dimiliki hanya 204 item dari 932 saja yang dirating.

Hasil waktu komputasi rata-rata sangat cepat dikarenakan beberapa faktor, antara lain:

1. Kemampuan server shared hosting sangat kuat.
2. Dataset yang dimiliki masih sedikit, sehingga item yang dievaluasi belum banyak.

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari implementasi dan uji coba yang dilakukan adalah sebagai berikut:

1. Sistem Rekomendasi (*Recommender System*) pada dasarnya bekerja dengan cara memberikan rekomendasi atau sugesti kepada pengguna (*user*) berdasarkan tingkat ketertarikan pengguna yang diekspresikan melalui bobot rating tiap-tiap barang (*item*).
2. Rekomendasi yang dibuat dengan metode *item-based collaborative filtering* menggunakan algoritma slope-one dengan masukan eksplisit berupa rating, menghasilkan keluaran yakni rekomendasi atau sugesti dengan tingkat personalisasi *personalized* dan *non-personalized*.
3. 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.
4. Keakurasian hasil perhitungan prediksi dicapai dengan nilai *Mean Absolut Error* (MAE) yang didapatkan sebesar 1,742.
5. Hasil perhitungan prediksi secara *online* dalam sebuah halaman web dapat ditampilkan dalam kisaran waktu kurang dari 1 detik sehingga mampu memberikan rekomendasi *real-time* kepada pengguna.
6. Dengan adanya sistem rekomendasi (*Recommender System*) ini diharapkan dapat membantu pengguna (*user*) untuk menemukan barang (*item*) yang diinginkan dan dapat membentuk sebuah komunitas yang saling memberi masukan.

5.2 Saran

Berikut ini adalah beberapa saran untuk pengembangan implementasi *Recommender Systems* lebih lanjut:

1. Untuk meningkatkan tingkat akurasi diperlukan sebuah dataset dengan skala besar yang melingkupi hampir semua korelasi antar-item, sebagai contoh GroupLens⁽¹⁷⁾ memiliki dataset yang dapat digunakan untuk domain film dengan lebih dari 1 juta rating untuk 3900 judul film dari 6040 *user*.
 2. Untuk membuat *Recommender Systems* yang lebih akurat, metode *item-based* yang digunakan dapat dipadukan dengan metode-metode lain seperti *market basket analysis* atau *content-based filtering*.
 3. Untuk mempertajam hasil rekomendasi, diperlukan data demografi dari pengguna, sehingga akan dapat diketahui pola seperti misalnya: pengguna laki-laki cenderung memilih game olahraga.
 4. *E-commerce* yang digunakan dalam Skripsi ini hanya bertujuan untuk simulasi, untuk dikemudian hari alangkah baiknya terdapat fasilitas pencetakan faktur untuk pengguna, konfirmasi pembayaran dan alternatif cara pembayaran yang terintegrasi dalam situs *e-commerce* tersebut.
-

DAFTAR PUSTAKA

- [1] *Amazon*. [Online] [Dikutip: 06 Maret 2010.] <http://www.amazon.com>.
- [2] *IMDb. The Internet Movie Database*. [Online] [Dikutip: 20 Oktober 2010.] <http://www.imdb.com>.
- [3] **Mlajenic, Dunja**. *Intelligent Systems and their Applications*. s.l. : IEEE, 1999. 1094-7167.
- [4] **Lin, Weiyang**. *Efficient Adaptive Support Association Rule Mining for Recommender Systems*. 2000.
- [5] **Sarwar, B, et al**. *Item-Based Collaborative Filtering Recommendation Algorithms*. Minneapolis : University of Minnesota, 2001.
- [6] *Gamefaqs*. [Online] Maret 26, 2010. <http://www.gamefaqs.com>.
- [7] **Larose, Daniel T**. *Discovering Knowledge in Data: An Introduction to Data Mining*. New Jersey : John Wiley & Sons, Inc, 2005.
- [8] **Pramudiono, Iko**. Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data. *Ilmu Komputer*. [Online] 2003. [Dikutip: 27 September 2009.] <http://www.ilmukomputer.com>.
- [9] **Schafer, Ben, Konstan, Joseph A dan Riedl, John**. *E-Commerce Recommendation Applications*. s.l. : Springer Netherlands, 2001. 1573-756X.
- [10] **Dürsteler, Juan C**. Web Mining. *InfoVis*. [Online] 18 September 2005. [Dikutip: 17 Februari 2010.] <http://www.infovis.net/printMag.php?lang=2&num=172>.
- [11] **Lemire, Daniel dan Maclachlan, Anna**. *Slope One Predictors for Online Rating-Based Collaborative Filtering*. 2005.
- [12] **Oracle**. MySQL 5.1 Reference Manual. Januari 1997 - 2011 .
- [13] **Vaswani, Vikram**. *MySQL : Database Usage & Administration*. Singapore : Mc Graw Hill, 2010. ISBN: 978-0-07-160549-6.
- [14] **Oracle**. MySQL Benchmarks. *MySQL*. [Online] [Dikutip: 22 01 2011.] <http://www.mysql.com/why-mysql/benchmarks/>.
- [15] **PHP Documentation Group**. PHP Manual. 1997 - 2010.
- [16] **Vaswani, Vikram**. *PHP: A Beginner's Guide*. Singapore : McGraw Hill, 2009. 0-07-154902-1.

- [17] Group Lens Dataset. *Grouplens*. [Online] [Dikutip: 27 01 2011.] <http://www.grouplens.org>.
 - [18] **Welling, Luke dan Thomson, Laura.** *PHP and MySQL. Web Development - 4th ed.* Singapore : Addison-Wesley, 2009. ISBN 978-0-672-32916-6.
 - [19] **Lemire, Daniel and McGrath, Sean.** *Implementing a Rating-Based Item-to-Item Recommender System in PHP/SQL.* 2005.
-

LAMPIRAN



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

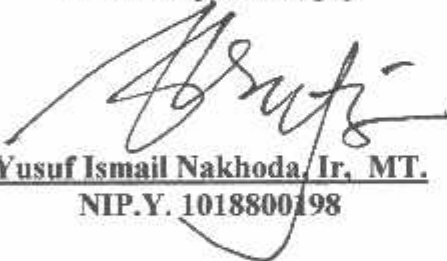
Nama : Aldi K. Dipasanta
NIM : 0512720
Jurusan : Teknik Elektro S-1
Konsentrasi : Komputer dan Informatika
Judul : IMPLEMENTASI *RECOMMENDER SYSTEM* DENGAN
MENGUNAKAN ALGORITMA *SLOPE-ONE* PADA WEB
E-COMMERCE PENJUALAN DVD GAMES

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada:


Hari : Jum'at
Tanggal : 18 Februari 2011
Nilai : 83,95 (A) *α*

Panitia Ujian Skripsi :

Ketua Majelis Penguji



Yusuf Ismail Nakhoda, Ir, MT.
NIP.Y. 1018800198

Sekretaris Majelis Penguji



Dr. Eng. Arvanto Soetedjo, ST, MT
NIP.Y. 1030800417

Anggota Penguji :

Penguji Pertama


Sandy Nataly M, S.Kom.
NIP.P. 1030800418

Penguji Kedua


Irmalia Suryani F, ST, MT.
NIP.P. 1030000365



PT. BNI (PERSERO) MALANG
BANK N. AGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : J. Bondungan Sgura-gura No. 2 Telp. (0341) 66143; (Hunting), Fax. (0341) 663315 Malang 65145
Kampus II : J. Raya Kalungo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : ALDI K. DIPASANTA
NIM : 0512720
Jurusan : Teknik Elektro S-1
Konsentrasi : Komputer dan Informatika
Masa Bimbingan : 17 Desember 2010 s/d 17 Juni 2011
Judul Skripsi : IMPLEMENTASI *RECOMMENDER SYSTEM* DENGAN MENGGUNAKAN ALGORITMA *SLOPE-ONE* PADA WEB *E-COMMERCE* PENJUALAN DVD GAMES

Tanggal	Penguji	Uraian	Paraf
18 Februari 2011	Penguji 1	1. Penulisan pada BAB I dan BAB V. 2. Tata Bahasa Penulisan pada Kesimpulan dan Saran	
	Penguji 2	1. Tidak tercantumnya Hasil Penelitian pada Abstrak. 2. Tujuan Penelitian yang ditujukan pada user / komunitas. 3. BAB V Kesimpulan	

Disetujui,

Dosen Penguji I

Sandy Nataly M, S.Kom.
NIP.P. 1030800418

Dosen Penguji II

Irmalia Suryani F, ST, MT.
NIP.P. 1030000365

Mengetahui,

Dosen Pembimbing

I Komang Somawirata, ST, MT.
NIP.Y. 1030100361



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : Aldi K Dipasanta
 NIM : 05.12.720
 Semester : 10
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : **TEKNIK ELEKTRONIKA**
TEKNIK ENERGI LISTRIK
TEKNIK KOMPUTER DAN INFORMATIKA
 Alamat : Banyuwangi Asri D-13 Singosari Malang

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

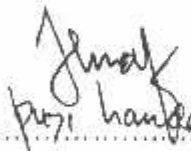
Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :


1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
 Recording Teknik Elektro


Malang,200
Pemohon

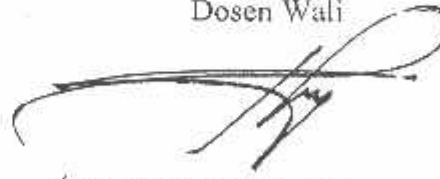

 (..... Puji Handayani))


 (..... Aldi K Dipasanta))

Disetujui
 Ketua Jurusan Teknik Elektro

Mengetahui
 Dosen Wali


 Ir. F. Yudi Limpraptono, MT
 NIP. P. 1039500274


 (.....))

Catatan :

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. IPK = $\frac{2 \times 459,5}{138} = 3,39$
2. Langkah
3.

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura-gura No.2
MALANG

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak **I Komang Somawirata, ST., MT.**
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini :

Nama : ALDI K DIPASANTA
Nim : 0512720
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika

Dengan ini mengajukan permohonan kiranya Bapak bersedia menjadi Dosen Pembimbing Utama / **Pendamping ***, untuk penyusunan Skripsi dengan judul (proposal terlampir) :

**IMPLEMENTASI *RECOMMENDER SYSTEM* DENGAN
MENGUNAKAN ALGORITMA *SLOPE-ONE* PADA WEB *E-COMMERCE* PENJUALAN DVD GAMES**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan Bapak kami ucapkan terima kasih.

Malang, 21 Mei 2010

Mengetahui
Ketua Jurusan Teknik Elektro

Hormat kami



Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039 5900274

Aldi K Dipasanta

*)coret yang tidak perlu

Form S-3 a

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : ALDI K DIPASANTA
Nim : 0512720
Semester : 10 (sepuluh)
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika

Dengan ini Menyatakan bersedia / tidak bersedia*) Membimbing Skripsi dari mahasiswa tersebut, dengan judul:

IMPLEMENTASI *RECOMMENDER SYSTEM* DENGAN MENGGUNAKAN ALGORITMA *SLOPE-ONE* PADA WEB *E-COMMERCE* PENJUALAN DVD GAMES

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 21 Mei 2010

**Kami yang membuat
pernyataan,**

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa/I yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.

*)coret yang tidak perlu



I Komang Somawirata, ST., MT.
NIP.P 103 0100361



FORMULIR BIMBINGAN SKRIPSI

Nama : ALDI K DIPASANTA
NIM : 05.12.720
Masa Bimbingan : 17 Desember 2010 s/d 17 Juni 2011 *AK*
Judul Skripsi : IMPLEMENTASI *RECOMMENDER SYSTEM* DENGAN MENGGUNAKAN
ALGORITMA *SLOPE-ONE* PADA WEB *E-COMMERCE* PENJUALAN DVD
GAMES

No	Tanggal	Uraian	Paraf Pembimbing
1	25 Januari 2011	Demo Program	<i>AK</i>
2	27 Januari 2011	BAB I, II, dan III	<i>AK</i>
3	31 Januari 2011	Flowchart Utama BAB IV	<i>AK</i>
4	31 Januari 2011	ACC Program	<i>AK</i>
5	3 Februari 2011	ACC BAB IV, V	<i>AK</i>
6	3 Februari 2011	Makalah Seminar	<i>AK</i>
7			
8			
9			
10			

Malang,
Dosen Pembimbing

(I Komang Somawirata, ST, MT)
NIP/P 103 0100361

FORM S-4b

LAMPIRAN

Core

db_connect.class.php

```
<?php

class DB_Connect{
    private $host;
    private $db_user;
    private $db_pass;
    private $db_name;
    private $mysqli;
    private $magic_quotes_active;
    private $real_escape_string_exists;
    public $bind_array;

    function __construct($host, $db_user, $db_pass, $db_name){
        $this->host = $host;
        $this->db_user = $db_user;
        $this->db_pass = $db_pass;
        $this->db_name = $db_name;
        $this->mysqliConnect(); //MySQLi
        $this->magic_quotes_active = get_magic_quotes_gpc();
        $this->real_escape_string_exists = function_exists(
"mysql_real_escape_string" );
    }

    function mysqliConnect(){
        $this->mysqli = new mysqli($this->host, $this->db_user,
$this->db_pass, $this->db_name);
        if ($this->mysqli->connect_error){
            die('Connect Error (' . $this->mysqli->connect_errno .
' ) : ' . $this->mysqli->connect_error);
        }
    }

    function query($sql){
        $result = $this->mysqli->query($sql);

        if(!preg_match("#^\s*SELECT #i", $sql))
            return $result;

        $res_array = array();

        if($result){
            while($line = $result->fetch_array(MYSQLI_BOTH)){
                $res_array[] = $line;
            }
            $result->close();
        }

        if(preg_match("#LIMIT +1(?:^\d)*$#is", $sql))
            return $res_array[0];

        return $res_array;
    }
}
```

```

function prepareStatement($sql, $param1, $param2, $param3){
    $stmt = $this->mysqli->prepare($sql);
    if (!$stmt) {
        echo "Error: ".$stmt->error."</br> Error no: ".$stmt->
>errno;
        exit;
    }
    $stmt->bind_param("iii", $param1, $param2, $param3);
    $stmt->execute();
    $stmt->bind_result($result);
    $this->bind_array = array(); //bind to array variable
    while ($stmt->fetch()){
        $this->bind_array[] = $result;
    }
    $stmt->close();
}

function prepareProcedure($sp_call){
    $stmt = $this->mysqli->prepare($sp_call);
    if (!$stmt) {
        echo "Error: ".$stmt->error."</br> Error no: ".$stmt->
>errno;
        exit;
    }
    $stmt->execute();
    $stmt->close();
}

function escape_value($value) {
    if( $this->real_escape_string_exists ) { // PHP v4.3.0 or
higher
        // undo any magic quote effects ->
mysql_real_escape_string
        if( $this->magic_quotes_active ; | $value =
stripslashes( $value ); )
            $value = $this->mysqli->real_escape_string($value );
    }
    return $value;
}
}
?>

```

slope_one.class.php

```

<?php

class Slope_One{
    private $db;
    private $active;
    private $user_id;
    private $user_idDump;
    private $item_id;
    private $limitNon;
    private $limitTopN;
    private $rat_min;
    private $sql;
    private $procedure;
    //private $categ_on;

    function __construct($db, $active/*, $categ_on*/){

```



```

        $this->db = $db;
        $this->active = $active;
        //$this->categ_on = $categ_on;
        if ($this->active == 1){
            $this->truncateSlopeOne();
            $this->callSlopeOne(/*$this->categ_on*/);
        }
    }

    function createProcedure_1(){ //MySQL pake 2 delimiter,
MySQL >5.xx
        $sp = <<<EOF

        CREATE PROCEDURE `sp_rss_slope_one`()
        begin
            DECLARE tmp_item_id int;
            DECLARE done int default 0;
            DECLARE mycursor CURSOR FOR select distinct id
from rss_user_ratings;
            DECLARE CONTINUE HANDLER FOR NOT FOUND set done=1;
            open mycursor;
            while (!done) do
                fetch mycursor into tmp_item_id;
                if (!done) then
                    insert into rss_slope_one (select a.id as
item_id1,b.id as item_id2,count(*) as times, sum(b.rating-a.rating) as
rating from rss_user_ratings a,rss_user_ratings b where a.id =
tmp_item_id and b.id != a.id and a.user_id=b.user_id group by
a.id,b.id);
                end if;
            END while;
            close mycursor;
        end

//
DELIMITER //
EOF;
    }

    function createProcedure_2(){ // slope_one berdasarkan
category, belum efektif karena minimnya data rating
        $sp = <<<EOF

        CREATE PROCEDURE `sp_rss_slope_one_cat`( IN categ INT)
        begin
            DECLARE tmp_item_id int;
            DECLARE done int default 0;
            DECLARE mycursor CURSOR FOR SELECT DISTINCT u.id
FROM rss_user_ratings u, rss_catalog_items c WHERE c.id = u.id AND
c.categ_id = categ;
            DECLARE CONTINUE HANDLER FOR NOT FOUND set done=1;
            open mycursor;
            while (!done) do
                fetch mycursor into tmp_item_id;
                if (!done) then
                    insert into rss_slope_one (select a.id as
item_id1,b.id as item_id2,count(*) as times, sum(b.rating-a.rating) as
rating from rss_user_ratings a,rss_user_ratings b where a.id =
tmp_item_id and b.id != a.id and a.user_id=b.user_id group by
a.id,b.id);
                end if;
            end if;
        end
    }

```

```

                END while;
                close mycursor;
            end
        //
        DELIMITER //
        EOF;
    }

    function truncateSlopeOne(){
        $sql = "TRUNCATE TABLE `rss_slope_one`";
        $this->db->query($sql);
    }

    function callSlopeOne(/*$categ_on*/){
        /*
        if ($categ_on == 1){
            $this->procedure_call = "CALL
`sp_rss_slope_one_cat`";
        }else{
            $this->procedure_call = "CALL
`sp_rss_slope_one`";
        }
        */
        $this->procedure_call = "CALL `sp_rss_slope_one`";
        $this->db->prepareProcedure($this->procedure_call);
    }

    function non_personalizedRecommendation($db, $item_id,
    $rat_min, $limitNon){
        $this->sql = "SELECT item_id2 AS item FROM rss_slope_one
WHERE item_id1 = ? AND times >= ? GROUP BY item_id2 ORDER BY
SUM(ABS(rating/times)) DESC LIMIT ? ";
        $this->item_id = $item_id;
        $this->rat_min = $rat_min;
        $this->limitNon = $limitNon;
        $this->db->prepareStatement($this->sql, $this->item_id,
    $this->rat_min, $this->limitNon);
        $value = array();
        foreach ($this->db->bind_array as $line){
            $value[] = $line;
        }
        return $value;
    }

    function personalizedRecommendation($db, $user_id, $limitTopN)
    {
        $this->sql = "SELECT s.item_id2 AS item FROM rss_slope_one
s, rss_user_ratings u WHERE u.user_id = ? AND s.item_id1 = u.id AND
s.item_id2 != ALL (SELECT u.id FROM rss_user_ratings u WHERE u.user_id
= ?) GROUP BY s.item_id2 ORDER BY SUM(u.rating * s.times -
s.rating)/SUM(s.times) DESC LIMIT ? ";
        $this->user_id = $user_id;
        $this->user_idDump = $user_id;
        $this->limitTopN = $limitTopN;
        $this->db->prepareStatement($this->sql, $this->user_id,
    $this->user_idDump, $this->limitTopN);
        $value = array();
        foreach ($this->db->bind_array as $line){
            $value[] = $line;
        }
    }

```

```
        return $value;
    }
}
?>
```

register.class.php

```
<?php
class Register{

    private $db;
    private $username;
    private $result;
    private $password;
    private $key;
    private $email;

    function __construct($db, $username){
        $this->db = $db;
        $this->username = $username;
        $this->checkUser();
    }

    function checkUser(){
        $sql = "SELECT * FROM `rss_users` WHERE `user_name` =
'".$this->username."'";
        $this->result = $this->db->query($sql);
    }

    function insertUser($db, $password, $email, $key){
        $this->password = $password;
        $this->email = $email;
        $this->key = $key;
        if($this->result){
            throw new Exception('Username yang digunakan sudah
terdaftar, silahkan mendaftarkan usorname baru');
        }
        $sql = "INSERT INTO `rss_users`
(`user_name`, `password`, `email`) VALUES ('".$this->
>username."',AES_ENCRYPT('".$this->password."', '".$this->
>key."', '".$this->email."'");";
        $result = $this->db->query($sql);
        if(!$result){
            throw new Exception("Tidak dapat mendaftarkan ke
Database - Silahkan coba kembali");
        }
    }

    function fetchID($db, $username){
        $this->db = $db;
        $this->username = $username;
        $this->checkUser();
        /* Gak perlu, kontrol ada di register.php
        if (!$this->result){ //->fetch row? coba
            echo "Belum Terdaftar";
        }
        */
        foreach ($this->result as $rs){
            $res = $rs[user id];
        }
    }
}
```

```
        return $res;
    }
}

?>
```

core.php

```
<?php
//require_once(dirname(__FILE__)."/db_sql.class.php");
require_once(dirname(__FILE__)."/func_send_mail.php");
require_once(dirname(__FILE__)."/func_translit.php");
require_once(dirname(__FILE__)."/misc.class.php");
require_once(dirname(__FILE__)."/base.mdl.php");
require_once(dirname(__FILE__)."/base.ctl.php");
require_once(dirname(__FILE__)."/catalog_categ.ctl.php");
require_once(dirname(__FILE__)."/catalog_item.ctl.php");
require_once(dirname(__FILE__)."/catalog.ctl.php");
//require_once(dirname(__FILE__)."/config.class.php");
require_once(dirname(__FILE__)."/shopcart.class.php");
//-----//
require_once(dirname(__FILE__)."/db_connect.class.php");
require_once(dirname(__FILE__)."/slope_one.class.php");
require_once(dirname(__FILE__)."/register.class.php");
require_once(dirname(__FILE__)."/func_filled_out.php");
require_once(dirname(__FILE__)."/func_valid_email.php");
require_once(dirname(__FILE__)."/func_rupiah.php");
require_once(dirname(__FILE__)."/func_substrwords.php");
?>
```

public

common.ini.php

```
<?php
session_start();
require_once(dirname(__FILE__)."/core/core.php");
require_once(dirname(__FILE__)."/config.ini.php");
date_default_timezone_set('UTC');
$postget = $_POST?$_POST:$_GET;
//var_dump($postget);
//$db = new
DB_sql($config["db_host"],$config["db_user"],$config["db_pass"],$config["db_name"],0);
$db = new
DB_Connect($config["db_host"],$config["db_user"],$config["db_pass"],$config["db_name"]);
```

```

/*
if($db->result != true)
{
    ?><a href="admin/catalog_db_install.php"><h1><?php echo $db-
>result?></h1></a><?php
        exit;
    }
*/
$catalog = new Catalog($config,$db,"catalog_categs","catalog_items");
$html = array();
$html['categ'] = $catalog->categs_get("WHERE
id=".(int)@$postget["categ_id"]);
$html["categs"] = $catalog->categs_gets();
$html["categs_parents"] = $catalog->categs-
>get_parents_($html["categ"]["id"]);
$html["categs_parents_ids"] = $catalog->categs-
>get_parents_ids($html["categs_parents"]);
//-----
//-----//
$slope = new Slope_One($db, $config['slope_active']);
?>

```

catalog.php

```

<?php
    include("common.ini.php");
    include("update.php");
    //-----Login-----//
    if (isset($postget['username']) && isset($postget['password']))
        //kalau user baru mau masuk
        $username = $postget['username'];
        $password = $postget['password'];
        $db_prefix = $config['db_prefix'];
        $key = $config['key'];

        $sql = "SELECT * FROM '". $db_prefix. "users' WHERE user_name =
        '". $username. "' AND password =
        AES_ENCRYPT('". $password. "', '". $key. "')";
        $result = $db->query($sql);
        foreach($result as $rs){
            $user_id = $rs['user_id'];
        }
        if ($result){
            $_SESSION['ras_valid_user'] = $username;
        }
    }

```

```

        $ SESSION['user_id'] = $user_id;
    }
}
//-----Tampilan-----//
if(isset($_POST['id'])) //per-id
{
    $html["item"] = $catalog->items-
>get_by_id((int)$_POST['id']);
    $html["id"] = $catalog->items->get_id_next($html["item"]);
    $html["head_title"] = "- ".$html["categ"]["title"]." -
".$html["item"]["title"];
    $html["meta_description"] = $html["item"]["comment"];
    include("catalog_div_head.php");
    include("catalog_div_left.php");
    include("catalog_main_item.php");
    include("catalog_div_right.php");
    include("catalog_foot.php");
}
else //per-category
{
    $html_temp = $catalog-
>items_gets_by_categ((int)$_POST['categ_id']);
    $html['count_total'] = $html_temp['count_total'];
    $html["pagination"] = $html_temp["pagination"];
    $html["items"] = $html_temp["items"];
    $html["head_title"] = "- ".$html["categ"]["title"];
    $html["meta_description"] = $html["categ"]["description"];

    if(isset($_GET["search"])) //search
    {
        $_GET["search"] = $db->escape_value($_GET["search"]);
        $result = $catalog->search($_GET["search"]);
        $html['items'] = $result['items'];
        $html['pagination'] = $result['pagination'];
        $html["head_title"] = "- search - ".$_GET["search"];
    }

    include("catalog_div_head.php");
    include("catalog_div_left.php");
    include("catalog_main_items.php");
}

```

```
include("catalog_div_rights.php");
include("catalog_foot.php");
}
```

?>

catalog_div_head.php

```
<?php

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>

    <meta name="Description" content="Tugas Akhir: Implementasi Slope
One" />
    <meta name="Keywords" content="Slope One, DVD Games, Simulasi,
Tugas Akhir" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
    <meta name="Distribution" content="Global" />
    <meta name="Author" content="Aldi K" />
    <meta name="Robots" content="index, follow" />
    <!-- _____ Stylesheet _____ -->
    <link rel="stylesheet" href="stylesheets/main.css" type="text/css"
/>
    <link rel="stylesheet" type="text/css"
href="stylesheets/starrating.css" />
    <!-- _____ Javascript _____ -->
    <script type="text/javascript"
src="javascripts/jquery.js"></script>
    <script type="text/javascript"
src="javascripts/starrating.js"></script>

    <title>Atronach <?php echo $html["head_title"]?></title>

</head>

<body>
<!-- wrap -->
<div id="wrap">
```

```

<div id="header">

    <h1 id="logo">Atronach<span class="blue">DVD</span></h1>
    <h2 id="slogan">Simulasi <i>e-commerce</i> penjual dvd</h2>

    <form method="get" class="searchform" action="catalog.php">
        <p><input class="textbox" type="text" name="search"
value="<?php echo
isset($_GET["search"])?htmlspecialchars($_GET["search"],
ENT_QUOTES):""?>"/>
        <input type="submit" name="Submit" class="button"
value="Search" /></p>
    </form>

</div>

<div id="menu">
    <ul>
        <li ><a href="index.php">Home</a></li>
        <?php foreach($html["categs"] as $categ);
if($categ["id"]!=$_POSTGET["categ_id"]&id="";else $id='id="current"';
?>
        <li <?php echo $id?> ><a href="catalog.php?categ_id=<?php
echo $categ["id"]?>"><?php echo $categ["title"]?></a></li>
        <?php } ?>

        <!-- li><a href="contact.php">Contact</a></li -->
    </ul>
</div>

```

catalog_div_left.php

```

<!-- content-wrap -left sidebar -->
<div id="content-wrap">

<div id="sidebar">

    <h1>User Account</h1>
    <div class="left-box">
        <?php
            if (isset($_SESSION['ras_valid_user']))(
                echo '<h2>Selamat
Datang&nbsp;';.$_SESSION['ras_valid_user']. '</h2><br>';

```

```

                                <td><input type="password" name="password"
size="12"></td></tr>
                                <tr><td colspan="2" align="center"><input
type="submit" value="Log in"></td></tr>
                                </table>
                                </form>
                                <?php
                                ;
                                }
                                ?>
</div>

```

```

<a href="shopcart.php"><h1>Shopcart</h1></a><i>(hanya simulasi,
tidak aktif)</i>

```

```

<div class="left-box">
    <?php include("shopcart_preview.html") ?>
</div>

```

```

<h1>Sistem</h1>

```

```

<p>&quot;<i>Recommender system</i> (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 .&quot;</p>

```

```

<p class="align-right"><a href="http://www.daniel-
lemire.com/fr/abstracts/TRDC1.html"><i>RSs</i></a></p>

```

```

<h1>Disclaimer</h1>

```

```

<p><strong> Situs ini hanya digunakan untuk simulasi, tidak
bertujuan komersil.</strong><br></p>

```

```

<p><strong> NOT FOR COMMERCIAL USE, SIMULATION PURPOSE
ONLY.</strong><br></p>

```

```

</div>

```

catalog_main_items.php

```

<div id="main">
<!-- h1><?php echo $html["categ"]["title"]?></h1 -->
<div align=left>
    <?php
        $child_categs = $catalog->categs_gets($html["categ"]["id"]);
        for($i=0;$i<count($child_categs);$i++)
        ;
        $_category = $child_categs[$i];
        if($i>0)echo " | ";

```

```

        <li class="current-rating" id="current-rating-<?php echo
        $item['id'];?>" style="width: <?php echo getRating( $db, $u_id,
        $item['id']);?>%";<!-- will show current rating --></li>
        <?php
            /* we need to generate 'id' for star rating.. this
            'id' will identify which data to execute */
            /* we will pass it in ajax later */
            ?>
            <span class="rateLinks" id="<?php echo $item['id'];?>">
                <span id="<?php echo $u_id;?>">
                    <li><a href="javascript:void(0)" title="Payah"
                    class="one-star">1</a></li>
                    <li><a href="javascript:void(0)" title="Kurang"
                    class="two-stars">2</a></li>
                    <li><a href="javascript:void(0)" title="Lumayan"
                    class="three-stars">3</a></li>
                    <li><a href="javascript:void(0)" title="Bagus"
                    class="four-stars">4</a></li>
                    <li><a href="javascript:void(0)" title="Hebat!!"
                    class="five-stars">5</a></li>
                </span></span>
            </ul>
            <br>
            <?php
            }else{
            ?>
            <br><br>
            <?php } } ?>
        <br><br>
        <div align=center><?php
        include(dirname(__FILE__)."/pagination.html")?></div>
    </div>

```

catalog_main_items.php

```

<div id="main">
<h1><?php echo $html['item']['title'];?></h1>
<table border=0>
    <tr>
        <td>

            <table width="100%">
                <tr><td>
                    <!-- a href="catalog_categs.php">Catalog</a --
                ><?php

```

```

        $categs_parents = $html["categs_parents"];
        for($i=count($categs_parents)-1;$i>=0;$i--
    )
        {
            $_category = $categs_parents[$i];
            $_category["id"]?>"><?php echo $_category["title"]?></a><?php
        }
    ?>
        &raquo; <?php echo $html['item']['title']?>
    </td></tr>
</table>
</td></tr>
</table>

<table width="100%">
    <tr>
        <td align="left">
            <?php
                $back = "[&laquo;prev]";
                $forw = "[next&raquo;]";
            ?>
            <?php if($html["id"]["prev"]!=""){
                echo $back?></a>
            }?>
        </td>
        <td align="right">
            <?php if($html["id"]["next"]!=""){
                echo $forw?></a>
            }?>
        </td></tr>
</table>

<table>
    <tr>
        <td align="left">
            "
            alt="">
            <?php if (isset($_SESSION['rss_valid_user'])) {

```

```

        $u_id = $_SESSION['user_id'];
    ?>
    <?php //-----
Star-Rating----- ?>
        <ul class='star-rating' id="star-rating-<?php echo
$html["item"]["id"];?>">
            <?php /* getRating($id) is to generate current
rating '/?>
                <li class="current-rating" id="current-rating-
<?php echo $html["item"]["id"];?>" style="width: <?php echo getRating(
$db, $u_id, $html["item"]["id"])?>%"><!-- will show current rating --
></li>
            <?php
                /* we need to generate 'id' for star rating..
this 'id' will identify which data to execute */
                /* we will pass it in ajax later */
            ?>
            <span class="ratelinks" id="<?php echo
$html["item"]["id"];?>">
                <span id="<?php echo $u_id;?>">
                    <li><a href="javascript:void(0)"
title="Payah" class="one-star">1</a></li>
                    <li><a href="javascript:void(0)"
title="Kurang" class="two-stars">2</a></li>
                    <li><a href="javascript:void(0)"
title="Lumayan" class="three-stars">3</a></li>
                    <li><a href="javascript:void(0)"
title="Bagus" class="four-stars">4</a></li>
                    <li><a href="javascript:void(0)"
title="Hebat!!" class="five-stars">5</a></li>
                </span></span>
            </ul>
            <?php
            }
        ?>
        <div class="description"><?php echo
$html['item']['description']?></div>
    </td>
</tr>
<tr><td>
    <div class="price"><strong>
        Rp&nbsp;<?php
            $rp_price = $html["item"]['price'];
            $indonesian_price = func_rupiah($rp_price);

```

```

        echo $indonesian_price;
    ?>
    </strong>
    <a href="shopcart.php?action=add&item_id=?php
echo $html["item"]["id"]?>">[Add to Cart]</a></div>
    </td></tr>
</table>
</div>

```

catalog_div_right.php

```

<div id="rightbar">
    <?php
        $rated = scanRating($db, $_SESSION['user_id']);
        //echo $_SESSION['rss_valid_user'];

        if(isset($_SESSION['rss_valid_user'])) {
            if($rated == 1) { ?>
                <h2>Anda mungkin menyukai item ini: </h2>
                <p>
                    <?php
                        $item_title_left = $html["item"]["title"]; //dump
variable lst
                        $item_id_left = $html["item"]["id"];
                        $item_recomendation = $slope-
>personalizedRecommendation($db, $_SESSION['user_id'],
$config['limitTopN']);
                        for ($i=0; $i<count($item_recomendation);$i++){
                            //echo $item_recomendation[$i]. "<br>";
                            $html["item"] = $catalog->items-
>get_by_id($item_recomendation[$i]);
                            //echo $html["item"]["title"]. "<br>";
                        ?>
                    <h3><?php
                        echo substrwords($html["item"]['title'], 23);
                    ?></h3>
                    <p>
                        <a href="?id=?php echo $html["item"]["id"]?>" >
                            " alt="<?php echo
$html["item"]["comment"]?>" class="float-left"/></a>
                            <br><br><br><br><br><br>
                    </p>
                </div>

```

```

        <?php }
    ?> </p>
    <h2>Rekomendasi <i>Non-Personalized</i> untuk <?php
        echo substrwords($item_title_left, 23);
    ?></h2>
    <p><?php
        $item_recomendation = $slope-
>non_personalizedRecommendation($db, $item_id_left, $config['rat_min'],
    $config['limitNon']);
        if ($item_recomendation == null){ ?>
            <h3>Maaf belum ada rekomendasi untuk item ini</h3>
        <?php }
        for ($i=0; $i<count($item_recomendation);$i++){
            //echo $item_recomendation[$i]."<br>";
            $html["item"] = $catalog->items-
>get_by_id($item_recomendation[$i]);
            //echo $html["item"]["title"];
        ?>
        <h3><?php
            echo substrwords($html["item"]["title"], 23);
        ?></h3>
        <p>
            <a href="?id=<?php echo $html["item"]["id"]?>" >
                " alt="<?php echo
    $html["item"]["comment"]?>" class="float-left"/></a>
                <br><br><br><br><br><br>
            </p>
        <?php
    }
    ?>
</p>

    <?php } else{ ?>
        <h2>Anda belum merating, tidak bisa memberikan rekomendasi
    <i>Personalized</i></h2>
        <h2>Rekomendasi <i>Non-Personalized</i> untuk <?php
            echo substrwords($html["item"]["title"], 23);
        ?></h2>
    <p><?php

```

```

        $item_recomendation = $slope-
>non_personalizedRecomendation($db, $html["item"]['id'],
$config['rat_min'], $config['limitNon']);
        if ($item_recomendation == null){ ?>
<h3>Maaf belum ada rekomendasi untuk item ini</h3>
<?php }
        for ($i=0; $i<count($item_recomendation);$i++){
            //echo $item_recomendation[$i]."<br>";
            $html["item"] = $catalog->items-
>get_by_id($item_recomendation[$i]);
            //echo $html["item"] ["title"];
        ?>
<h3><?php
            echo substrwords($html["item"]['title'], 23);
        ?></h3>
<p>
            <a href="?id=<?php echo $html["item"] ["id"]?>" >
                " alt="<?php echo
$html["item"] ["comment"]?>" class="float-left"/></a>
                <br><br><br><br><br><br>
            </p>
<?php
    }
?>
</p>
<?php } ?>

<?php
} else { ?>
<h2>User yang menyukai <?php
    echo substrwords($html["item"] ['title'], 15);
    ?>, juga menyukai :</h2>
<p><?php
    $item_recomendation = $slope-
>non_personalizedRecomendation($db, $html["item"] ['id'],
$config['rat_min'], $config['limitNon']);
    if ($item_recomendation == null){ ?>
<h3>Maaf belum ada rekomendasi untuk item ini</h3>
<?php }
    for ($i=0; $i<count($item_recomendation);$i++){
        //echo $item_recomendation[$i]."<br>";

```

```

        $html["item"] = $catalog->items->
>get_by_id($item_recomendation[$i]);
        //echo $html["item"]["title"];
    ?>
    <h3><?php
        echo substrwords($html["item"]['title'], 23);
    ?></h3>
    <p>
        <a href="?id=<?php echo $html["item"]["id"]?>" >
            " alt="<?php echo
$html["item"]["comment"]?>" class="float-left"/></a>
            <br><br><br><br><br><br>
        </p>
    <?php
    }
?>
</p>
<?php }?>
</div>

```

catalog_div_rights.php

```

<div id="rightbar">
    <?php
        if (function_exists('scanRating')){
            $rated = scanRating($db, $_SESSION['user_id']);
            if(isset($_SESSION['rss_valid_user'])){
                if($rated == 1){ ?>
                    <h2>Anda mungkin menyukai item ini: </h2>
                    <p>
                        <?php
                            $item_recomendation = $slope->
>personalizedRecomendation($db, $_SESSION['user_id'],
$config['limitTopN']);
                            for ($i=0; $i<count($item_recomendation);$i++){
                                //echo $item_recomendation[$i]."<br>";
                                $html["item"] = $catalog->items->
>get_by_id($item_recomendation[$i]);
                                //echo $html["item"]["title"]."<br>";
                            ?>
                            <h3><?php

```

```

                echo substrwords($html["item"]['title'],
23);
                ?></h3>
                <p>
                <a href="?id=<?php echo $html["item"]['id']?>"
>
                " alt="<?php echo
$html["item"]['comment']?>" class="float-left"/></a>
                <br><br><br><br><br><br><br>
                </p>
                <?php }
                ?> </p>

                <?php }else{ ?>
                <h1>Anda belum merating, tidak bisa memberikan
rekomendasi <i>Personalized</i></h1>
                <?php } }else{ ?>
                <h2><i> Recommender Systems</i></h2>
                <p>Bekerja untuk membantu user dalam memilih iter yang
akan dibeli dengan cara memberi informasi atau sugesti berdasarkan
tingkat ketertarikan atau selera masing-masing user.<br>
                <h3>Silahkan Login, dan merating salah satu item</h3>
                </p>
                <?php } ?>

```

</div>

register.php

```

<?php
    include("common.ini.php");
    $html["head_title"] = " :: Registration ";
    $html["categs"] = $catalog->categs_gets();

    include_once("catalog_div_head.php");
    /*
    if(isset($_SESSION["rss_valid_user"])){
        include("catalog_div_left_user.php");
    } else{
        include("catalog_div_left.php");
    }
    */
    if (isset($_postget['daftar'])) {

```

```

                echo substrwords($html["item"]['title'],
23);
                ?></h3>
                <p>
                    <a href="?id=<?php echo $html["item"]['id']?>"
>
                        " alt="<?php echo
$html["item"]['comment']?>" class="float-left"/></a>
                    <br><br><br><br><br><br><br>
                </p>
                <?php }
            ?> </p>

            <?php }else{ ?>
                <h1>Anda belum merating, tidak bisa memberikan
rekomendasi <i>Personalized</i></h1>
                <?php }}}else{ ?>
                <h2><i> Recommender Systems</i></h2>
                <p>Bekerja untuk membantu user dalam memilih iter yang
akan dibeli dengan cara memberi informasi atau sugesti berdasarkan
tingkat ketertarikan atau selera masing-masing user.<br>
                <h3>Silahkan Login, dan merating salah satu item</h3>
                </p>
            <?php } ?>

</div>

```

register.php

```

<?php
    include("common.ini.php");
    $html["head_title"] = " :: Registration ";
    $html["cstegs"] = $catalog->categs_gets();

    include_once("catalog_div_head.php");
    /*
    if(isset($_SESSION:"rss_valid_user")){
        include("catalog_div_left_user.php");
    } else{
        include("catalog_div_left.php");
    }
    */
    if (isset($_postget:'daftar')){

```

```

$username = $postget['username'];
$pass_0 = $postget['pass_0'];
$pass_1 = $postget['pass_1'];
$email = $postget['email'];
$key = $config['key'] ;
$register = new Register($db,$username);
try{
    //cek isi
    if(!func_filled_out($postget)){
        throw new Exception('Form belum terisi, silahkan
daftar kembali');
    }
    //cek email
    if(!func_valid_email($email)){
        throw new Exception('Bukan email yang valid. Silahkan
kembali dan coba lagi');
    }
    //cek password
    if($pass_0 != $pass_1){
        throw new Exception('Password tidak sama. Silahkan
kembali dan coba lagi');
    }
    //cek panjang password
    if((strlen($pass_0) < 6) || (strlen($pass_0) > 16)){
        throw new Exception('Panjang password antara 6 hingga
16 karakter. Silahkan kembali dan coba lagi');
    }
    //register
    //func_register($username,$pass_0,$email,$key);
    $register->insertUser($db,$pass_0,$email,$key);
    $_SESSION['rss_valid_user'] = $username;
    $_SESSION["user_id"] = $register-
>fetchID($db,$_SESSION['rss_valid_user']);
    include_once("catalog_div_left.php");
    echo "Terima kasih ".$_SESSION['rss_valid_user']. " telah
mendaftar";
    include_once("catalog_div_rights.php");
    include_once("catalog_foot.php");

}catch(Exception $e){
    echo $e->getMessage();
}

```

```

        exit;
    }
}
else{
//fixme: form register diperbaiki?
<div id="main">
<h1>Pendaftaran - <i>Registration</i></h1>
<form action="register.php" method="post">
    <table>
        <tr>
            <td>Username</td>
            <td><input type="text" name="username"></td></tr>
        <tr>
            <td>Password</td>
            <td><input type="password" name="pass_0"></td></tr>
        <tr>
            <td>Ulangi Password</td>
            <td><input type="password" name="pass_1"></td></tr>
        <tr>
            <td>Alamat e-mail</td>
            <td><input type="text" name="email"></td></tr>
        <tr><td colspan="2" align="center"><input type="submit"
name="daftar" value="Daftar"></td></tr>
    </table>
</form>
</div>
<?php
}
?>

```

update.php

```

<?php
    include("common.ini.php");
    // $db = new DB_Connect($config['db_host'], $config['db_user'],
    $config['db_pass'], $config['db_name']);

    if($_GET['do']=='rate'){
        // do rate
        rate($db, $_GET['u_id'],$_GET['id']);
    }else if($_GET['do']=='getrate'){
        // get rating
        getRating($db,$_GET['u_id'],$_GET['id']);
    }
}

```

```

    )

    function scanRating($db, $u_id){ //if user ever rated
    $sql = "SELECT * FROM `rss_user_ratings` WHERE `user_id` =
    ".$u_id.";";
    $result = $db->query($sql);
    $result_val = 1;
    if ($result == null){
        $result_val = 0;
    }
    return $result_val;
}

// function to retrieve current rating from vote_ex and
predictRating() when zero it capable
function getRating($db, $u_id, $id){
    $sql = "SELECT * FROM `rss_user_ratings` WHERE `user_id` =
    ".$u_id." AND `id` = ".$id.";";

    //$result=@mysql_query($sql);
    $result = $db->query($sql);
    //$rs=@mysql_fetch_array($result);
    //if ($result){
    foreach ($result as $rs){
        $rating = $rs[rating] * 20;
    }
    //}else{
    //    $rating = 1 * 20;
    //}
    echo $rating;
}

// function to insert rating
function rate($db, $u_id, $id){
    //$rating = $_GET['rating'];
    $text = strip_tags($_GET['rating']);
    $update = "UPDATE `rss_user_ratings` SET `rating` =
    ".$_GET['rating']." WHERE `user_id` = ".$u_id." AND `id` = ".$id.";";

    //$result = @mysql_query($update);

```

```

$result = $db->query($update);
//if(mysql_affected_rows() == 0){
if ($result){
    $insert = "INSERT INTO `rss_user_ratings`
(`user_id`,`id`,`rating`) VALUES (".$u_id.", ".$id.",
".$_GET['rating'].");";

    //$result = @mysql_query($insert);
    $result = $db->query($insert);
}
}

```

?>

Javascript

starrating.js

```

//Coba jqSajax? weyah, kesel
//var data=$.x_slope_personalizedPrediction(var1,var2);
//var result=$.x_prediction(var1,var2);

// JavaScript Document
$(document).ready(function() {
    // get rating function
    function getRating(u_id,id){
        $.ajax({
            type: "GET",
            url: "update.php",
            data: "do=getrate&u_id="+u_id+"&id="+id,
            cache: false,
            async: false,
            success: function(result) {
                // apply star rating to element
                $("#current-rating-"+id+"").css({ width: ""
+ result + "%" });
            },
            error: function(result) {
                alert("some error occured, please try again
later");
            }
        });
    }
}

```

```

// link handler
$('.ratelinks li a').click(function(){
    // get the parent id
    var idUser = $(this).parent().parent().attr('id');
    var idStar =
$(this).parent().parent().parent().attr('id');
    $.ajax({
        type: "GET",
        url: "update.php",
        data:
"rating="+$(this).text()+"&do-rate&u_id="+idUser+"&id="+idStar,
        cache: false,
        async: false,
        success: function(result) {
            // remove id #ratelinks element to prevent
            // class .ratelinks tidak bisa per id ->
            // hapus per id Bintang dari DOM
            $("#"+idStar).remove();
            // get rating after click
            getRating(idUser,idStar);
        },
        error: function(result) {
            alert("some error ocured, please try again
later");
        }
    });
});
});
});

```

Stylesheet

main.css

```

/*****
HTML ELEMENTS
*****/

/* top elements */
* {
    margin: 0; padding: 0;

```

```
}
body {
    margin: 0; padding: 0;
    font: 71%/1.5em Verdana, 'Trebuchet MS', Arial, Sans-serif;
    background:url(headerbg-blue.gif) repeat-x;
    color: #666666;
    text-align: center;
}

/* links */
a {
    background: inherit;
    color: #3182C0;
}
a:hover {
    background: inherit;
    color: #72A545;
}

/* headers */
h1, h2, h3 {
    font: bold 1em 'Trebuchet MS', Tahoma, Sans-serif;
    text-transform: uppercase;
    color: #555;
}
h1 { font-size: 1.5em; }
h2 { font-size: 1.3em; }
h3 { font-size: 1.2em; text-transform: none;}

#main h1, #rightbar h1 {
    padding: 10px 0 5px 5px;
    margin: 0 0 0 10px;
    text-transform: uppercase;
    border-bottom: 1px solid #f2f2f2;
}

#rightbar h3{
    padding: 10px 0 5px 5px;
    margin: 0 0 0 10px;
}
```

```
    text-transform: none;
}

#sidebar h1 {
    padding: 10px 0px 5px 30px;
    background: url(square_arrow.gif) no-repeat 2px 12px;
    margin: 0;
    text-transform: uppercase;
}

p, h1, h2, h3 {
    margin: 10px 15px;
}

ul, ol {
    margin: 10px 30px;
    padding: 0 15px;
    color: #3162C0;
}

ul span, ol span {
    color: #666666;
}

/* images */
img {
    border: 2px solid #CCC;
}

img.float-right {
    margin: 5px 0px 5px 15px;
}

img.float-left {
    margin: 5px 15px 5px 0px;
}

a img {
    border: 2px solid #3791D7;
}

a:hover img {
    border: 2px solid #806B4D !important; /* IE fix*/
    border: 2px solid #3791D7;
}

}
```

```
code {
    margin: 5px 0;
    padding: 10px;
    text-align: left;
    display: block;
    overflow: auto;
    font: 500 1em/1.5em 'Lucida Console', 'courier new', monospace;
    /* white-space: pre; */
    background: #FAFAFA;
    border: 1px solid #f2f2f2;
    border-left: 3px solid #3182C0;
}
acronym {
    cursor: help;
    border-bottom: 1px solid #777;
}
blockquote {
    margin: 15px;
    padding: 0 0 0 32px;
    background: #FAFAFA url(quote.gif) no-repeat 5px 10px !important;
    background-position: 8px 10px;
    border: 1px solid #f2f2f2;
    border-left: 3px solid #3182C0;
    font-weight: bold;
}

/* form elements */
form {
    margin: 10px; padding: 0;
    border: 1px solid #f2f2f2;
    background-color: #FAFAFA;
}
label {
    display: block;
    font-weight: bold;
    margin: 5px 0;
}
input {
    padding: 4px;
}
```

```
border:1px solid #eee;
font: normal 1em/1.5em Verdana, sans-serif;
color:#777;
}
textarea {
width:350px;
padding:4px;
font: normal 1em/1.5em Verdana, sans-serif;
border:1px solid #eee;
height:100px;
display:block;
color:#777;
}
input.button {
margin: 0;
font: bold 1em Arial, Sans-serif;
border: 1px solid #ccc;
background: #fff;
padding: 2px 3px;
color: #333;
}

/* search form */
form.searchform {
background: transparent;
border: none;
margin: 0; padding: 0;
}
form.searchform input.textbox {
margin: 0;
width: 120px;
border: 1px solid #ccc;
background: #fff;
color: #333;
vertical-align: top;
}
form.searchform input.button {
width: 55px;
vertical-align: top;
```
