Jurnal Ilmu Komputer Agri-Informatika

Tersedia secara online di:
http://journal.ipb.ac.id/index.php.jika

Volume 2 Nomor 1 halaman 9-19
ISSN: 2089-6026

Pembuatan Modul Rekomendasi pada OpenCart Menggunakan Metode *Item-Based Collaborative Filtering*

Making Recommender Module in OpenCart Using Item-Based Collaborative Filtering Method

KIRANA NURYUNITA*, YANI NURHADRYANI

Abstrak

Penelitian ini bertujuan menambahkan modul rekomendasi pada *content management system* Opencart. Salah satu pendekatan dalam rekomendasi adalah *item-based collaborative filtering*. Metode *item-based collaborative filtering* dapat mengurangi waktu eksekusi perhitungan. Metode *item-based collaborative* filtering pada penelitian ini menggunakan perhitungan *adjusted cosine similarity* untuk menghitung nilai kemiripan antarbuku dan *weighted sum* untuk menghitung nilai prediksi *rate* buku. Untuk mendapatkan rekomendasi, pengguna harus melakukan *login* dan memberikan *rate* pada buku. Berdasarkan *rate* pengguna, nilai kemiripan dihitung menggunakan *adjusted cosine similarity*. Berdasarkan kemiripan antarbuku, nilai prediksi *rate* buku dicari menggunakan *weighted sum.* Sebelum buku direkomendasikan kepada pengguna, kategori prediksi buku dicocokkan dengan kategori buku yang telah diberi *rate* oleh pengguna. Penelitian ini menggunakan 300 buku dan 30 pengguna sebagai data. Dari hasil penelitian, hanya 17 pengguna yang mendapatkan rekomendasi. Pengujian dilakukan dengan menganalisis waktu eksekusi dan keakuratan rekomendasi. Waktu eksekusi dalam pengujian ini adalah 1.60 detik. Untuk menghitung keakuratan rekomendasi, penelitian ini menggunakan *mean absolute error* dengan hasil perhitungan 0.15.

Kata kunci: *e-commerce, item-based collaborative filtering, recommender system*.

Abstract

This research aims to add recommender module in OpenCart content management system. One of the recommender method is item-based collaborative filtering. Item-based collaborative filtering method can reduce execution time of calculation. The method of item-based collaborative filtering in this study uses adjusted cosine similarity to calculate the similarity between the books and uses weighted sum method to calculate the books rate prediction. To get the recommendation, user have to login and give ratings to the books first. Based on user's rate, adjusted cosine similarity calculate the similarity between books. Based on the similarities between books, weighted sum method calculate the books rate prediction. Before a book is recommended to the user, type of the book from prediction are then matched with the book type rated by a user. This research uses 300 books and 30 users as the data. From the result, only 17 users will get recommendations based on the calculation from adjusted cosine similarity and weighted sum method. Testing is done by analyzing execution time and calculation error. The result of execution time is 1.60 seconds. To count calculation error, this research use mean absolute error and the result of calculation is 0.15.

Keywords: *e-commerce, item-based collaborative filtering,recommender system*.

PENDAHULUAN

Informasi dengan berbagai macam kriteria terdapat di internet, salah satunya adalah informasi tentang penjualan produk dan jasa secara *online* atau biasa disebut *e-commerce*.

Faktor yang dapat mendorong penjualan dalam *e-commerce* ialah menyangkut personalisasi dari pembeli di internet. Sistem rekomendasi merupakan salah satu bentuk dari personalisasi *website*. Sistem rekomendasi merupakan model penyelesaian masalah yang menerapkan teknik-teknik tertentu pada pembuatan rekomendasi untuk pemilihan suatu informasi, produk, dan jasa (Goldberg *et al*. 2001).

Terdapat dua pendekatan dalam mengembangkan sebuah sistem rekomendasi, yaitu *content-based* dan *collaborative filtering* (Balabanovic dan Shoham 1997). Pendekatan *content-based* menyediakan rekomendasi dengan cara membandingkan representasi konten yang terkandung dalam sebuah produk dengan representasi konten yang diinginkan pengguna dengan membangun *user thematic profile*, sedangkan pendekatan *collaborative filtering* bekerja dengan cara mempelajari masukan *rate* dari pengguna pada kumpulan produk, mengenali kesamaan berdasarkan masukan pengguna, dan menghasilkan rekomendasi baru (Candillier *et al.* 2007). Rekomendasi berdasarkan pendekatan *collaborative filtering* merupakan pendekatan yang paling banyak digunakan untuk membangun sebuah sistem rekomendasi (Burke 2002). Teknik *collaborative filtering* terbagi menjadi dua, yaitu *user-based collaborative filtering* dan *item-based collaborative filtering* (Karypis 2001). Pada *user-based collaborative filtering*, sistem mencari sejumlah pengguna yang memiliki korelasi tinggi, sedangkan pada *item-based collaborative filtering*, sistem mencari sejumlah produk yang memiliki korelasi tinggi. *User-based collaborative filtering* memerlukan data pengguna. Pengguna lama dapat mengubah pola perilaku mereka dan pengguna baru dapat memasuki sistem setiap saat. Banyaknya data pengguna membutuhkan proses yang cukup lama untuk menghasilkan rekomendasi. *Item-based collaborative filtering* memerlukan data koleksi produk yang relatif lebih sedikit dibanding data pengguna bersifat statis sehingga memungkinkan pengurangan perhitungan (Sarwar *et al*. 2001).

Pembuatan situs *e-commerce* dapat dilakukan dengan menggunakan *content management system* (CMS). Salah satu CMS yang populer di masyarakat adalah OpenCart (Masalov 2007). OpenCart telah memiliki beberapa modul seperti *account*, *affiliate*, *banner*, *bestsellers*, *carousel*, *category*, *featured*, *googletalk*, *information*, *latest*, *slideshow*, *specials*, *store*, dan *welcome*. Namun, OpenCart belum memiliki modul yang dapat menghasilkan rekomendasi untuk pengguna.

Penelitian ini menambahkan modul baru, yaitu modul rekomendasi pada CMS OpenCart. Modul rekomendasi dikembangkan menggunakan metode *item-based collaborative filtering*. Metode *item-based collaborative filtering* pada penelitian ini menggunakan perhitungan *adjusted cosine similarity* untuk menghitung nilai kemiripan antarproduk dan *weighted sum* untuk menghitung nilai prediksi *rate* produk.

METODE PENELITIAN

Metode yang dilakukan pada penelitian ini terdiri atas beberapa tahap, yaitu analisis kebutuhan, perancangan sistem, implementasi *item-based collaborative filtering*, dan pengujian *item-based collaborative filtering*.

Analisis Kebutuhan

Analisis meliputi analisis kebutuhan data dan analisis kebutuhan fungsi. Analisis kebutuhan data adalah analisis CMS yang cukup populer di masyarakat Indonesia. CMS dicari berdasarkan Indonesia *traffic rank* tertinggi melalui situs alexa.com. Data buku yang digunakan pada penelitian ini berjumlah 300 buku yang diambil dari situs bukukita.com. Data pengguna pada penelitian ini berjumlah 30 pengguna mahasiswa IPB yang memberi *rate* untuk minimal 5 buku pada sistem. Tahapan pada analisis fungsi terdiri atas pemberian *rate* oleh pengguna, perhitungan rata-rata *rate* pengguna, perhitungan nilai kemiripan produk menggunakan *adjusted cosine similarity*, perhitungan prediksi produk menggunakan *weighted*

sum, dan pencocokan kategori produk prediksi dengan kategori produk yang diberi *rate* oleh pengguna.

Perancangan Sistem dan Implementasi *Item-Based Collaborative Filtering*

Perancangan sistem terdiri atas perancangan *database* dan perancangan antarmuka. Pada tahap implementasi, *adjusted cosine similarity* dan *weighted sum* diterjemahkan ke dalam sistem dan ditambahkan pada modul untuk menghasilkan sebuah rekomendasi. Persamaan *adjusted cosine similarity* digunakan untuk menghitung nilai kemiripan antarproduk. Persamaan *adjusted cosine similarity* adalah sebagai berikut (Karypis 2001):

$$\text{sim}_{(i,j)} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_u)^2}} \quad (1)$$

dengan:

$\text{sim}_{(i,j)}$ adalah nilai kemiripan antara produk *i* dan produk *j*,
$u \in U$ adalah himpunan pengguna u yang memberikan *rate* pada produk *i* dan produk *j*,
$r_{u,\text{i}}$ adalah *rating* pengguna *u* pada produk *i*,
$r_{u,j}$ adalah *rating* pengguna *u* pada produk *j*,
$\bar{r}_u$ adalah rata-rata *rating* pengguna *u*.
Dalam menghitung nilai kemiripan, nilai yang dihasilkan oleh persamaan *adjusted-cosine similarity* berkisar antara +1.0 dan -1.0. Informasi korelasi yang diketahui berdasarkan pada nilai kemiripan menurut Marmanis dan Babenko (2009): nilai kemiripan 0: kedua produk tidak berkorelasi (independen), Nilai kemiripan mendekati +1.0: kedua produk berkorelasi tinggi, dan nilai kemiripan mendekati -1.0: kedua produk saling bertolak belakang.

Setelah mendapatkan sekumpulan produk yang sangat mirip, dilakukan proses prediksi yang memperkirakan nilai *rate* dari pengguna. Prediksi yang diperkirakan adalah produk yang belum pernah diberi *rate* oleh pengguna. Teknik yang digunakan untuk mendapatkan nilai prediksi adalah dengan persamaan *weighted sum* (Karypis 2001) berikut:

$$\text{P}(u,j) = \frac{\sum_{i \in I}(\text{R}_{u,i} * \text{S}_{i,j})}{\sum_{i \in I}|\text{S}_{i,j}|} \quad (2)$$

dengan:

P(u,j) adalah prediksi untuk pengguna *u* pada produk *j*.
$i \in I$ adalah himpunan produk yang mirip dengan produk *j*.
$R_{u,i}$ adalah *rate* pengguna *u* pada produk *i*.
$S_{i,j}$ adalah nilai kemiripan antara produk *i* dan produk *j*.

Pengujian *Item-Based Collaborative Filtering*

Pengujian terhadap sistem dilakukan dengan menggunakan *mean absolute error* (MAE) untuk mengetahui nilai keakuratan dari rekomendasi yang dihasilkan. MAE yang ditunjukkan pada persamaan 3 merupakan persamaan yang digunakan untuk mengukur akurasi sistem dengan membandingkan nilai yang diprediksi dengan nilai yang sebenarnya (Karypis 2001). Persamaan MAE digunakan untuk mengevaluasi kualitas dari sistem dan perhitungan yang paling sering digunakan (Candillier *et al*. 2007). Semakin kecil nilai MAE yang dihasilkan, prediksi yang dihasilkan semakin baik.

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

dengan:

MAE adalah nilai rata-rata kesalahan hitung.

N adalah jumlah produk yang dihitung.
p_i adalah nilai prediksi produk ke-*i*.
q_i adalah nilai *rate* sebenarnya ke produk *i*.

Pengujian juga dilakukan dengan menghitung waktu eksekusi yang dibutuhkan sistem untuk menghasilkan sebuah rekomendasi. Pengujian dilakukan pada lingkungan pengembangan perangkat keras *ProcessorIntel Core* 2 *Duo* 1.6 GHz, RAM 2GB, dan *Hard drive* 120 GB.

HASIL DAN PEMBAHASAN

Analisis Kebutuhan

Kebutuhan Data

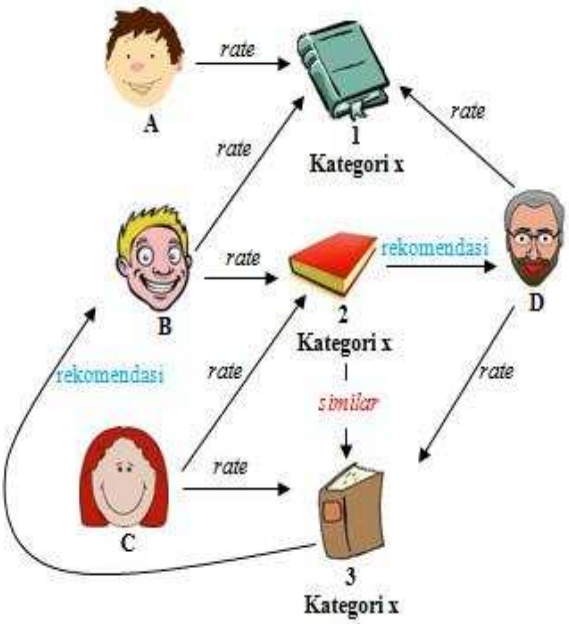
Analisis mengenai pemilihan CMS yang dipakai dilakukan dengan situs Alexa. Pemilihan Alexa sebagai *tool* analisis data website dikarenakan situs ini menyediakan layanan perangkingan secara otomatis untuk seluruh situs di dunia (Saidah 2012). Berdasarkan situs Alexa yang diakses dengan *queryopensource shopping cart*, lima global *traffic rank* tertinggi adalah magentocommerce, sitepoint, prestashop, rackspace hosting, dan OpenCart. Dari 5 peringkat berdasarkan global *traffic rank* tersebut, peringkat berdasarkan Indonesia *traffic rank* dicari. Situs yang memiliki Indonesia *traffic rank* tertinggi ialah OpenCart.

Aspek kesuksesan *e-commerce* menurut O'Brien (2006) terdiri atas sembilan aspek dengan 7 aspek merupakan proses yang dilakukan secara *online* dan 2 aspek lainnya merupakan proses yang dilakukan secara *offline*. OpenCart telah memiliki 7 aspek *online* berdasarkan O'Brien (2006), yaitu:

1 Pengendalian akses dan keamanan: terdapat verifikasi bagi pengguna yang *login*.
2 Manajemen pencarian: terdapat fungsi pencarian yang memudahkan pengguna dalam mencari suatu produk.
3 Manajemen isi: terdapat fungsi untuk memperbaharui maupun menambah produk.
4 Manajemen katalog: terdapat fungsi pembagian produk menjadi kategori-kategori tertentu.
5 Pembayaran: terdapat alur proses belanja sampai verifikasi pembayaran.
6 Pemberitahuan kegiatan: terdapat notifikasi apabila pengguna melakukan suatu proses pada sistem.
7 Pembuatan profil dan personalisasi: terdapat fungsi untuk menyimpan profil pengguna dan *history* pembelian bagi pengguna yang telah *login* ke dalam sistem. OpenCart hanya memiliki aspek pembuatan profil. Salah satu cara untuk menciptakan personalisasi ialah dengan adanya sistem rekomendasi.

Kebutuhan Fungsi

Gambar 1 menunjukkan simulasi rekomendasi yang terjadi pada sistem. Sistem memanfaatkan masukan *rate* dari pengguna untuk menghasilkan rekomendasi. Pada Gambar 1, pengguna A memberikan *rate* pada buku 1, pengguna B memberikan *rate* pada buku 1 dan 2, pengguna C memberikan *rate* pada buku 2 dan 3, dan pengguna D memberikan *rate* pada buku 1 dan 3. Berdasarkan masukan *rate* tersebut, dicari pasangan buku yang memiliki korelasi tinggi atau *similar* menggunakan *adjusted cosine similarity*. Pada Gambar 1, buku yang memiliki korelasi tinggi adalah buku 2 dan buku 3. Nilai prediksi *rate* dari pasangan buku yang memiliki korelasi tinggi dicari menggunakan *weighted sum*. Kategori prediksi buku lalu dicocokkan dengan kategori buku yang telah diberi *rate*. Buku 1, 2, dan 3 memiliki kategori yang sama pada Gambar 1. Pengguna yang memiliki rekomendasi adalah pengguna B dengan rekomnedasi buku 3 dan pengguna D dengan rekomendasi buku 2.



Gambar 1 Simulasi rekomendasi

Pada DFD konteks (Gambar 2) dapat dilihat bahwa pengguna memberi *rate* pada suatu buku. *Rate* tersebut lalu diolah oleh sistem untuk mencari buku yang memiliki kemiripan dengan buku yang telah diberi *rate* sebelumnya. Apabila terdapat buku yang memiliki kemiripan, sistem memberikan rekomendasi buku pada pengguna.

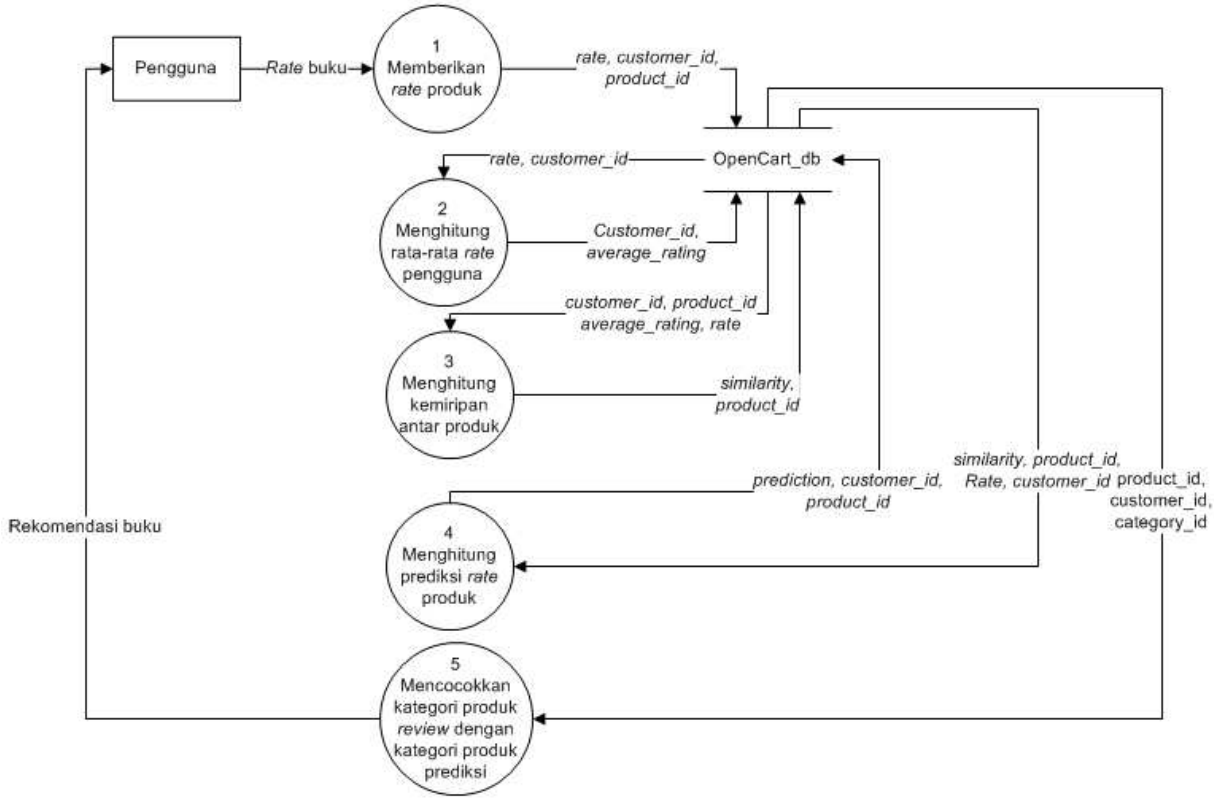
Gambar 2 DFD konteks sistem rekomendasi

DFD level 1 (Gambar 3) menjelaskan alur rekomendasi. Atribut yang terkait dengan pemberian *rate* pada produk, yaitu *rate*, *customer_id*, dan *product_id*, disimpan dalam *database*. Kemudian, sistem menghitung rata-rata *rate* pengguna dan disimpan dengan atribut *average_rating* dan *customer_id*. Setelah rata-rata *rate* dihitung, sistem menghitung kemiripan antarproduk dan menyimpannya dengan atribut *similarity*, *product_id*, dan *customer_id*. Sistem lalu menghitung nilai prediksi *rate* produk dan menyimpannya dengan atribut *prediction rate*, *product_id*, dan *customer_id*. Kemudian, kategori produk yang telah diberi *rate* oleh pengguna dicocokkan dengan kategori produk prediksi. Kategori yang memiliki kesamaan direkomendasikan kepada pengguna.

Berdasarkan pada Gambar 3, simulasi dari proses rekomendasi dibuat. Proses pertama dan kedua dalam menghasilkan rekomendasi adalah mengumpulkan data pengguna yang telah memberi *rate* pada buku dan menghitung rata-rata *rate* dari pengguna. Pemberian *rate* dilakukan oleh pengguna yang telah *login*. Simulasi data pengguna yang telah memberi *rate* dapat dilihat pada Tabel 1. Pada tabel, *U* melambangkan pengguna, *I* melambangkan buku, dan $\bar{U}$ melambangkan nilai rata-rata *rate* pengguna.

Proses ketiga ialah menghitung nilai kemiripan antarbuku menggunakan *adjusted cosine similarity*. Simulasi nilai kemiripan antarbuku dapat dilihat pada Tabel 2, dengan *I* melambangkan buku. Berdasarkan Tabel 1, terdapat empat pasang buku yang memiliki nilai kemiripan lebih besar dari 0.5 yaitu I_1 dengan I_5, I_3 dengan I_5, I_3 dengan I_6, dan I_5 dengan I_6.

Proses keempat ialah menghitung nilai prediksi *rate* buku menggunakan *weighted sum*. Pasangan buku yang dijadikan rekomendasi adalah pasangan buku yang nilai kemiripannya lebih besar dari 0.5. Pengguna yang diberikan prediksi *rate* buku hanya pengguna yang baru memberikan *rate* pada salah satu buku (Tabel 3), dengan *U* melambangkan pengguna dan *I* melambangkan buku.



Gambar 3 DFD level 1sistem rekomendasi

Tabel 1 *Rate* buku dari pengguna

Pengguna	I_1	I_2	I_3	I_4	I_5	I_6	I_7	$\overline{U}$
U_1	3	5	3	-	-	-	2	3.25
U_2	5	-	-	2	5	5	-	4.25
U_3	1	-	2	-	-	-	2	1.67
U_4	-	-	-	4	5	5	-	4.67
U_5	4	2	3	3	3	-	-	3.00
U_6	3	4	-	-	-	1	4	3.00
U_7	-	-	-	-	-	4	4	4.00
U_8	-	5	5	2	-	5	-	4.25
U_9	-	-	5	-	5	-	4	4.67
U_{10}	-	4	-	-	5	-	3	4.00
U_{11}	-	-	-	2	4	-	3	3.00
U_{12}	-	-	-	4	3	-	5	4.00

Sebelum prediksi diberikan kepada pengguna, terdapat proses kelima yaitu pencocokan prediksi buku dengan kategori buku yang telah diberi *rate* oleh pengguna. Apabila kategori buku dari prediksi yang dihasilkan sama dengan kategori buku berdasarkan *rate* dari pengguna, buku prediksi tersebut direkomendasikan.

Tabel 2 Nilai kemiripan antarbuku

Buku1	Buku2	Nilai kemiripan
I_1	I_2	-0.62
I_1	I_3	-0.31
I_1	I_4	-0.60
I_1	I_5	0.60*
I_1	I_6	0.35
I_1	I_7	0.08
I_2	I_3	0.07
I_2	I_4	-0.60
I_2	I_5	0.00
I_2	I_6	-0.54
I_2	I_7	-0.31
I_3	I_4	-1.00
I_3	I_5	1.00*
I_3	I_6	1.00*
I_3	I_7	0.26
I_4	I_5	-0.69
I_4	I_6	-0.99
I_4	I_7	0.00
I_5	I_6	1.00*
I_5	I_7	-0.80
I_6	I_7	-1.00

*Pasangan buku yang memiliki nilai *similarity* lebih besar dari 0.5

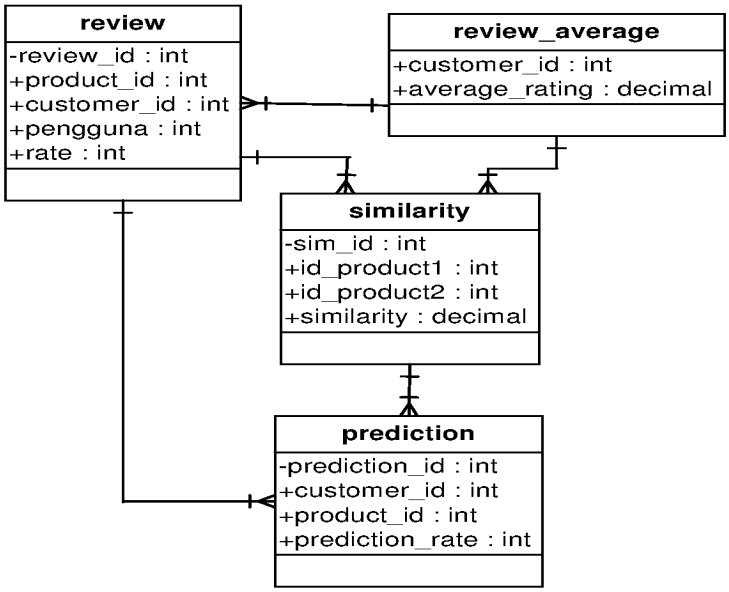
Tabel 3 Prediksi *rate* buku

Pengguna	Prediksi *rate* buku
U_1	(I_5, 3)(I_6, 3)
U_2	(I_3, 5)
U_3	(I_5, 2)(I_6, 2)
U_4	(I_1, 5)(I_3, 5)
U_5	(I_6, 3)
U_6	(I_3, 1)(I_5, 2)
U_7	(I_3, 4)(I_5, 4)
U_8	(I_5, 5)
U_9	(I_1, 5)(I_6, 5)
U_{10}	(I_1, 5)(I_3, 5)(I_6, 5)
U_{11}	(I_1, 4)(I_3, 4)(I_6, 4)
U_{12}	(I_1, 3)(I_3, 3)(I_6, 3)

Perancangan Sistem

Perancangan *Database*

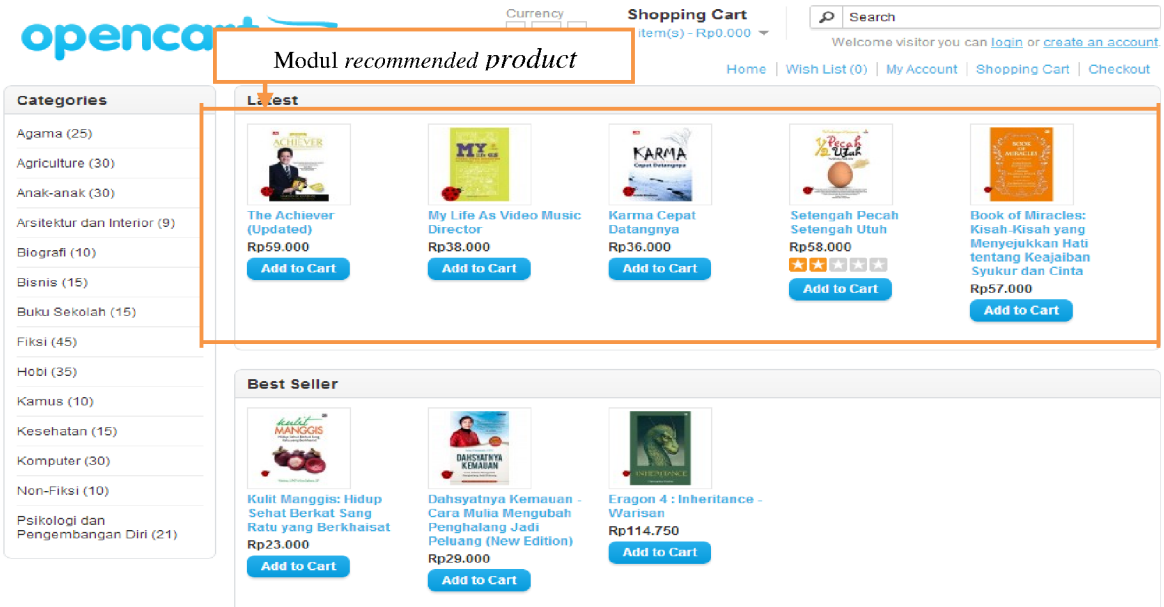
CMS OpenCart yang digunakan telah memiliki *database* agar sistem dapat berjalan dengan baik. Walaupun OpenCart telah memiliki *database* yang cukup lengkap, untuk menambahkan modul rekomendasi diperlukan beberapa tambahan tabel. Pada pembahasan pemodelan proses rekomendasi diperlukan empat tabel untuk menyimpan data yang dibutuhkan dalam menghasilkan rekomendasi. Tabel yang dibutuhkan adalah tabel untuk menyimpan nilai *rate* dari pengguna, tabel untuk menyimpan nilai rata-rata *rate* dari pengguna, tabel untuk menyimpan nilai kemiripan antarbuku, dan tabel untuk menyimpan prediksi buku (Gambar 4). Tabel yang digunakan untuk menyimpan nilai *rate* dari pengguna telah ada pada OpenCart, yaitu tabel *review*. Tabel lainnya tidak terdapat pada OpenCart sehingga ditambahkan ke dalam *database*. Tabel *review_average* ditambahkan untuk menyimpan nilai rata-rata *rate* dari pengguna. Tabel *similarity* ditambahkan untuk menyimpan nilai kemiripan antarbuku. Tabel *prediction* ditambahkan untuk menyimpan nilai prediksi buku.



Gambar 4 *Entity relationship diagram* sistem rekomendasi

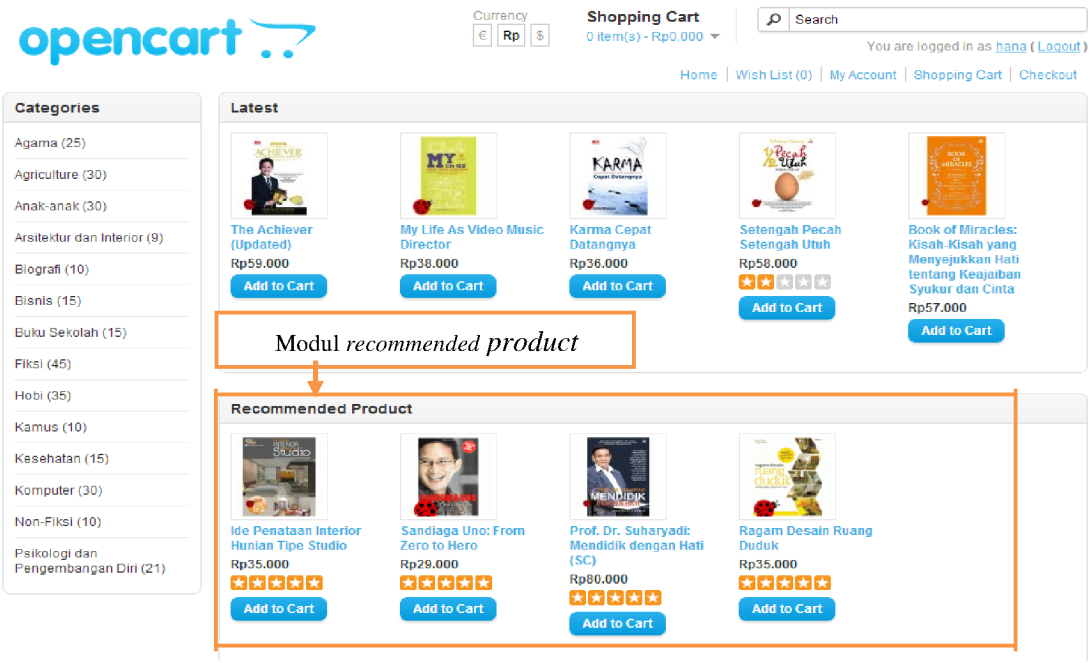
Perancangan Antarmuka

Antarmuka pada halaman *home* ditambah dengan modul baru, yaitu *recommended product*. Untuk pengguna yang belum *login* dan pengguna yang telah *login* tetapi tidak memiliki rekomendasi, modul *recommended product* menampilkan data buku berdasarkan *best seller* (Gambar 5).



Gambar 5 Halaman *home* untuk pengguna yang belum *login* dan pengguna yang telah *login* tetapi tidak memiliki rekomendasi

Data *best seller* didapat dari perhitungan pemesanan buku terbanyak yang dilakukan pada sistem. Untuk pengguna yang telah *login* dan memiliki rekomendasi buku, modul *recommended product* menampilkan data buku hasil rekomendasi (Gambar 6).



Gambar 6 Halaman *home* pengguna yang telah *login* dan memiliki rekomendasi

Implementasi *Item-Based Collaborative Filtering*

Adjusted Cosine Similarity

Adjusted cosine similarity digunakan untuk menghitung nilai kemiripan antarbuku. Data yang diperlukan dalam perhitungan *adjusted cosine similarity* adalah data *rate* pengguna dan data rata-rata *rate* pengguna. Data *rate* untuk menghitung kemiripan diambil dari tabel *review* pada *database*, sedangkan data rata-rata *rate* dari pengguna diambil dari tabel *review_average* pada *database*. Data pada tabel *review_average* dicari dengan menghitung rata-rata *rate* dari pengguna, yang dihitung dengan menggunakan *query* sebagai berikut:

```
SELECT
   customer_id, AVG (rating) AS average_rating
   FROM review
   GROUP BY (customer_id).
```

Implementasi *adjusted cosine similarity* pada *query* ialah sebagai berikut:

```
SELECT
(SUM((item1.rating-item1.average_rating)*(item2.rating-
   item2.average_rating))/(SQRT(SUM(POW((item1.rating-
   item1.average_rating                    ),2)))*SQRT(SUM(POW((item2.rating-
   item2.average_rating),2))))) AS similarity
FROM
(SELECT r.customer_id, r.product_id, r.rating, ra.average_rating FROM review r,
   review_average ra WHERE r.customer_id=ra.customer_id AND r.product_id =
   $prod ) AS item1,
(SELECT r.customer_id, r.product_id, r.rating, ra.average_rating FROM review r,
   review_average ra WHERE r.customer_id=ra.customer_id AND r.product_id =
   $product ) AS item2
WHERE
item1.customer_id=item2.customer_id).
```

Pada *query adjusted cosine similarity*, $prod dan $product merupakan parameter masukan dari id buku yang telah diberi *rate*. Nilai kemiripan ini disimpan dalam tabel *similarity* pada *database*. Nilai kemiripan yang disimpan dalam *database* hanya yang memiliki nilai kemiripan lebih besar dari 0.5. Berdasarkan *query adjusted cosine similarity*, terdapat 93 pasangan buku yang berkorelasi tinggi. Pasangan buku yang berkorelasi tinggi disimpan dalam tabel *similarity* pada *database*. Data pada tabel *similarity* semuanya memiliki

nilai kemiripan 1.0. Nilai kemiripan yang seragam ini disebabkan oleh: (1) pengguna yang memberi *rate* pada buku yang berbeda dengan nilai yang sama, dan (2) sebuah buku hanya diberi *rate* oleh satu atau dua pengguna.

Weighted Sum

Weighted sum digunakan untuk menghitung nilai prediksi *rate* buku. Data yang diperlukan dalam perhitungan *weighted sum* adalah data *rate* pengguna dan data kemiripan antarbuku yang berkorelasi tinggi. Data *rate* pengguna diambil dari tabel *review* dan data kemiripan antarbuku diambil dari tabel *similarity*. Implementasi *weighted sum* pada *query* adalah sebagai berikut:

```
SELECT
   (SUM(r.rating*s.sim)/SUM(ABS(s.sim))) AS prediction
FROM
   (SELECT product1, product2, sim FROM similarity)s,
   (SELECT customer_id, product_id, rating
      FROM review
      WHERE customer_id=$customer)r
WHERE
   (s.product1 = $product1 . "' AND s.product2 = r.product_id) XOR (s.product1
   = r.product_id AND s.product2 = $product2)).
```

Pada *query weighted sum*, $customer merupakan parameter masukan dari customer_id. $product1, dan $product2 merupakan masukan dari pasangan buku yang terdapat pada tabel *similarity*. Berdasarkan *query weighted sum* terdapat 297 nilai prediksi *rate* buku. Prediksi yang diberikan kepada pengguna adalah prediksi buku yang belum pernah diberi *rate* oleh pengguna dan kategorinya sama dengan kategori buku yang pernah diberi *rate* oleh pengguna yang bersangkutan. Pengguna yang memiliki rekomendasi hanya 17 pengguna, yaitu pengguna dengan *customer_id* 6, 7, 8, 10, 11, 15, 17, 18, 20, 23, 24, 25, 28, 30, 31, 32, dan 33. Beberapa pengguna tidak mendapatkan rekomendasi karena pengguna hanya memberi *rate* pada kategori tertentu saja atau *rate* yang diberikan seorang pengguna bertolak belakang dengan *rate* yang diberikan oleh pengguna lainnya.

Pengujian *Item-Based Collaborative Filtering*

Pengujian keakuratan rekomendasi

Pengujian dilakukan dengan menghitung *Mean Absolute Error* (MAE) dan waktu eksekusi yang dibutuhkan untuk menjalankan *query adjusted cosine similarity* dan *query weighted sum*. Pengujian yang pertama dilakukan adalah melakukan perhitungan MAE dengan membandingkan nilai *rate* prediksi dan nilai *rate* yang diberi oleh pengguna. Implementasinya pada *query* ialah sebagai berikut:

```
SELECT
SUM(ABS(p.prediction-r.rating))/COUNT(*)  FROM  prediction  p,  review  r  WHERE
   p.customer_id = r.customer_id AND p.product_id = r.product_id).
```

Berdasarkan perhitungan MAE, tingkat kesalahan perhitungan yang dihasilkan relatif kecil, yaitu 0.145. Semakin kecil nilai MAE, semakin akurat rekomendasi yang dihasilkan.

Pengujian waktu eksekusi

Pengujian yang dilakukan selanjutnya merupakan penghitungan waktu eksekusi untuk menjalankan *query* persamaan 1 dan 2. Penghitungan waktu eksekusi dilakukan dengan aplikasi tambahan yang terdapat pada *web browser*, yaitu *page load time*. Berdasarkan perhitungan *page load time*, rata-rata waktu eksekusi yang dibutuhkan adalah 60.7 detik. Tingginya waktu eksekusi ini disebabkan oleh proses pembuatan tabel yang cukup lama. Namun, lamanya waktu eksekusi ini tidak mengganggu pengguna karena algoritme diletakkan pada bagian *admin* sistem. Waktu eksekusi rata-rata dari 30 pengguna adalah 1.57 detik. Waktu eksekusi pada bagian pengguna digunakan untuk membandingkan kategori dari buku

yang telah diberi *rate* dan kategori dari prediksi buku. Menurut Miller (1968), respons pengguna terhadap waktu eksekusi halaman dibagi menjadi tiga, yaitu: (i) 0.1 detik yaitu pengguna merasa sistem bekerja sangat cepat, (ii) 1 detik yaitu batas pengguna tidak merasakan adanya gangguan, dan (iii) 10 detik yaitu batas memungkinkan pengguna untuk tetap fokus dan tidak meninggalkan halaman *website*. Waktu eksekusi untuk menghasilkan rekomendasi pada penelitian ini berada pada bagian ketiga yang masih memungkinkan pengguna untuk tetap fokus dan tidak meninggalkan halaman *website*.

SIMPULAN

Metode *item-based collaborative filtering* telah berhasil diimplementasikan menggunakan CMS OpenCart. Keakuratan metode *item-based collaborative filtering* dihitung menggunakan persamaan MAE. Tingkat kesalahan hitung metode *item-based collaborative filtering* pada sistem ini relatif kecil, yaitu 0.15. Semakin kecil nilai MAE, semakin akurat rekomendasi yang dihasilkan. Waktu eksekusi untuk menghasilkan prediksi membutuhkan waktu yang relatif lama, yaitu 61.00 detik. Waktu eksekusi untuk menampilkan rekomendasi pada pengguna relatif cepat dengan rata-rata 1.60 detik.

DAFTAR PUSTAKA

Balabanovic M, Shoham Y. 1997. Fab: Content-based collaborative recommendation. *Communication of the ACM* 40(3):66-72.

Burke R. 2002. Hybrid Recommender Systems: survey and experiments. *User Modelling and User-Adapted Interactioni.* 12(4):331-370.

Candillier L, Meyer F, Boullé M. 2007. Comparing state-of-the-art collaborative filtering systems. Di dalam: *International Conference on Machine Learning and Data* Mining; 2007 Jul 18-20; Leipzig, Jerman. hlm 548-562.

Goldberg K, Roeder T, Gupta D, Perkins C. 2001. Eigenstate: A constant time collaborative filtering algorithms. *Information Retrieval Journal* 4:133-151.

Karypis G. 2001. Evaluation of ttem-based top-N recommendation. Di dalam: *10th International Conference of Information and Knowledge Management (CIKM)*; 2001 Nov 5-10; Atlanta, Amerika Serikat. hlm 247-254.

Marmanis H, Babenko D. 2009. *Algorithms of the Intelligent Web*. Greenwich(UK): Manning Publ.

Masalov K. 2007. Developing Country E-commerce portal.*Honours project report*. University of Cape Town.

Miller RB. 1968. Response time in man-computer conversational transactions. Di dalam: *Proceedings of AFIPS Fall Joint Computer Conferencel*; 1968 Des 9-11; San Fransisco, Amerika Serikat. hlm 267-277.

O'Brien JA. 2006. *Introduction to Information System.*Ed ke-12. New York(US): McGraw Hill.

Saidah HT. 2012. Kajian *Usability website e-commerce* Indonesia berdasarkan perspektif tipe pengguna *browser* dan *evaluator* [skripsi]. Bogor(ID): Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor

Sarwar B, Karypis G, Konstan JA, Riedl J. 2001. Item-based collaborative filtering recommendation algorithms. Di dalam: *Proceedings of the 10thInternational World Wide Web Conference*; 2001 Mei 1-5; Hong Kong. hlm 285-295.