

repository.ub.ac.id

SISTEM REKOMENDASI PENCARIAN BUKU DIPERPUSTAKAAN MENGGUNAKAN METODE *ITEM-BASED COLLABORATIVE FILTERING* {Studi Kasus: Perpustakaan Universitas Brawijaya}

Rahman Anam¹, Rekyan Regasari Mardi Putri, S.T, M.T.², Agus Wahyu Widodo, S.T, M.Cs³
Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya Malang
Email: rahmananam36@gmail.com¹, rekyan.rmp@ub.ac.id², a_wahyu_w@ub.ac.id³

ABSTRACT

Daya baca mahasiswa di Indonesia masih sangat rendah. Banyak faktor yang melatarbelakangi hal ini, salah satunya mahasiswa kekurangan referensi mengenai buku apa yang sesuai dengan minat maupun kebutuhan mereka. Perpustakaan Universitas Brawijaya adalah perpustakaan yang memiliki literatur yang cukup lengkap. Namun karena banyaknya koleksi buku tersebut mahasiswa sering mendapat kesulitan menemukan buku yang sesuai dengan minat dan kebutuhannya. Dengan permasalahan tersebut maka diusulkan penelitian tentang sistem rekomendasi pencarian buku dengan menggunakan metode *item-based collaborative filtering* yang akan memberikan rekomendasi buku kepada mahasiswa berdasarkan korelasi peminjaman yang pernah dilakukan *user* dengan yang dilakukan *user* lain. Data peminjaman akan diubah menjadi nilai kemudian dilakukan perhitungan dengan algoritma *Adjusted Cossine Similarity* dan *Weight Sum* sehingga menghasilkan prediksi nilai *rating* terhadap buku yang belum pernah dipinjam oleh mahasiswa. Hasil pengujian menunjukkan nilai MAE terendah didapatkan ketika tingkat *sparsity* sebesar 30% yaitu rata-rata 0,44251. Sistem dapat mengatasi tingkat *sparsity* hingga 60% dengan rata-rata MAE yang diperoleh sebesar 0,8376. Dari hasil pengujian Relevansi dengan nomor DDC menghasilkan rekomendasi buku yang relevan dengan *user* sebanyak 57%. Dari pengujian waktu eksekusi sistem yang dibutuhkan ternyata dipengaruhi oleh jumlah *item* dan tingkat *sparsity*. Meningkatnya jumlah *item* diiringi dengan peningkatan waktu eksekusi yang dibutuhkan. Sedangkan untuk tingkat *sparsity* semakin tinggi tingkat *sparsity*-nya waktu eksekusi yang dibutuhkan relatif lebih sedikit.

Kata kunci: Sistem Rekomendasi, Perpustakaan, *Item-base Collaborative Filtering*.

ABSTRACT

Literacy of college student in Indonesia is very low. There are many factors behind this case, one of the factor is college students are lack of reference about which books that suitable with their interest and needs. Library of Brawijaya University is library which has a fairly comprehensive literature. However, the numerous collection of books make college students often have trouble finding books that match with their interest and needs. Based on this problem. the proposed research on a book search recommendation system using the item-based collaborative filtering which will give book recommendations to students based on the correlation of borrowing he had done with the other people do. Data loan will be converted into a value and then calculating the Adjusted algorithm Cossine Similarity and Weight Sum resulting in a rating of the predictive value of a book that has never been borrowed by college students. The test results showed the lowest MAE values obtained when the sparsity rate of 30% with an average of 0.44251. The system can cope with the level of sparsity of up to 60% with an average MAE obtained at 0.8376. The relevance of the test results with the initial tiaiit DDC numbers resulted in a recommendation of books that are relevant to the user as much as 57%. From the test system execution time required is influenced by the number of items and the level of sparsity. Increasing the number of items accompanied by an increase in execution time required. As for the higher level of sparsity execution time required is relatively less.

Keywords: Recommendation System, Library, Item-base Collaborative Filtering.

1. PENDAHULUAN

Daya baca masyarakat Indonesia bisa dibilang cukup rendah. Rata-rata secara nasional mengatakan bahwa tak sampai satu buku pun yang dibaca orang dalam setahun. Dari survey yang dilakukan oleh Badan Perpustakaan Nasional Dan Arsip Daerah (BPAD) pada tahun 2012, di Yogyakarta BPAD menyatakan bahwa indeks baca di kota tersebut merupakan yang tertinggi di

Indonesia, dengan angka 0.049. Sedangkan rata-rata secara nasional adalah 0.01. Namun ini terhitung kecil jika dibandingkan dengan Singapura yang sudah mencapai 0.45. Rendahnya daya baca masyarakat dapat diindikasikan melalui dua hal. Pertama adalah rendahnya jumlah buku yang terbit di Indonesia. Jika di Inggris bisa mencapai 100.000 judul pertahun, maka di Indonesia hanya bisa menerbitkan antara 5000 hingga 10.000 saja buku

per tahunnya. Kedua adalah pilihan masyarakat untuk mendapatkan informasi melalui media cetak(23,5%) kalah dengan televisi yang mencapai(85,9%), dan radio(40,3%)(Kriswijayanti, 2009). Bahkan untuk kalangan mahasiswa yang tergolong kaum mintelektual pun daya bacanya masih rendah. Banyak faktor yang melatarbelakangi hal ini, seperti sistem pembelajaran di Indonesia yang kurang membuat mahasiswanya harus mencari informasi melalui buku. Selain itu, mahasiswa juga kekurangan referensi mengenai buku apa yang sesuai dengan minat maupun kebutuhan mereka.

Universitas Brawijaya sebagai salah satu universitas negeri terbesar di Indonesia tentu memiliki perpustakaan yang memiliki literatur yang cukup lengkap. Semakin banyak koleksi buku di perpustakaan akan semakin membantu kegiatan riset dan akademis mahasiswa. Namun karena banyaknya koleksi buku tersebut mahasiswa sering mendapat kesulitan menemukan buku yang sesuai dengan minat dan kebutuhannya. Karena itu perlu adanya upaya untuk lebih meningkatkan lagi daya baca mahasiswa terutama di perpustakaan. Salah satu caranya adalah dengan membuat suatu sistem yang memunculkan suatu rekomendasi buku. Rekomendasi ini bisa diberikan ketika mahasiswa melakukan pencarian di katalog. Jadi, mahasiswa akan diberikan rekomendasi sesuai buku yang dipilih ketika melakukan pencarian. Dengan mendapatkan rekomendasi literatur lain, maka buku yang dibaca pun semakin banyak dan daya baca mahasiswa pun meningkat dan juga akan berguna untuk studinya. Selain itu, dengan sistem rekomendasi ini dapat menjadi sebuah stimulus agar mahasiswa meminjam lagi buku sesuai dengan minat atau kebutuhannya.

Sistem rekomendasi mencoba memprediksi sesuatu yang mungkin disukai oleh *user* tentang suatu *item* berdasarkan informasi yang diberikan oleh *user*. Secara umum ada tiga pendekatan yang digunakan dalam sistem rekomendasi, yaitu, *Content-based filtering*, *Collaborative filtering* dan *Hybrid* (Adomavicius, & Tuzhilin, 2005). *Content-based filtering* memberikan rekomendasi berdasarkan deskripsi dari *item* dengan membandingkan karakteristik informasi suatu *item* yang disukai oleh *user* sebelumnya. Metode ini memiliki kekurangan yaitu *item* tidak dapat direkomendasikan jika *user* memilih *item* yang memiliki konten berbeda dengan *item-item* yang pernah dipilih oleh *user* sebelumnya dan juga ketika jumlah *item* yang akan direkomendasikan menjadi bertambah besar, sistem akan membutuhkan waktu yang lama untuk melakukan ekstraksi pada

setiap *item* yang akan direkomendasikan (Pazzani, 2007).

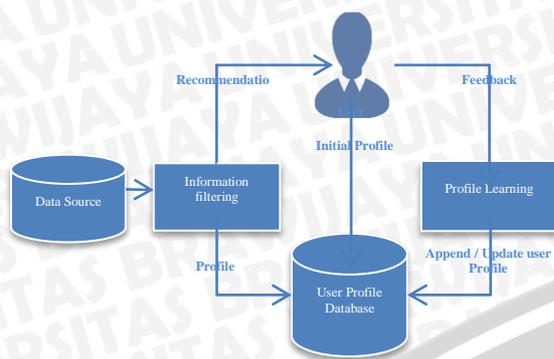
Berbeda dengan metode *Collaborative filtering*, metode *Collaborative filtering* menghasilkan rekomendasi berdasarkan keterkaitan antar *user* yang menyukai suatu *item* tertentu. *Collaborative Filtering* dibagi menjadi dua pendekatan yaitu *item-based* dan *user-based*. Perbedaannya, jika *user-based* menghitung kesamaan di antara *user* sebagai parameter untuk menghasilkan rekomendasi. Maka *item-based* akan menghitung kesamaan di antara *item*, dilihat dari penilaian yang diberikan *user* untuk *item* tersebut. Dalam kasus ini terdapat juga pendekatan dalam mengkategorikan buku berdasarkan subjek menggunakan *Dewey Decimal Classificatio*(DDC). DDC adalah sebuah sistem klasifikasi/pengkategorian buku perpustakaan berdasarkan subjek. Subjek pada DDC diklasifikasi pada kesepuluh kelas yang meliputi seluruh ilmu pengetahuan manusia. Menurut sistem klasifikasi Persepuluh *Dewey*, ilmu pengetahuan manusia dapat dibagi ke dalam sepuluh kelas utama(Subrata, 2009). Perbedaan sistem DDC dengan penelitian ini adalah pada pengkategorian. Penelitian ini pengkategorian berdasarkan prediksi perilaku *user* terhadap suatu *item*(buku).

Dengan menggunakan metode *item-base Collaborative Filtering* diharapkan dapat memiliki tingkat akurasi yang tinggi dalam merekomendasikan buku dan bermanfaat bagi mahasiswa dalam membantu mempermudah mahasiswa dalam menemukan referensi bacaan yang serupa, dan juga mempermudah pencarian suatu referensi bacaan dengan memberikan saran atau rekomendasi buku yang sesuai berdasarkan ketertarikan mahasiswa dengan memanfaatkan metode *Item-based Collaborative filtering*.

2. LANDASAN KEPUSTAKAAN

2.1 Sistem Rekomendasi

Sistem rekomendasi merupakan sistem yang dirancang untuk memprediksi sekumpulan *item* yang sesuai dengan preferensi *user* dimana *item* tersebut akan direkomendasikan pada *user* (Sanjung, 2011). Perkiraan informasi ini diperoleh melalui profil *item*, preferensi *item* dan aktivitas yang terjadi pada sistem. Profil *item* dapat berisi tentang informasi *item*, ketertarikan *item* pada suatu *item* dan juga history interaksi antara *item* dengan *item*. Misalnya ketika *item/* pelanggan melakukan pembelian rumah maka data *history* pembelian ini akan dicatat dan menjadi profil *item*.



Gambar 1. Arsitektur sistem rekomendasi

Sistem rekomendasi pada umumnya melakukan dua proses yaitu *information filtering* dan *profile learning*. *Information filtering* akan menghasilkan rekomendasi *item* yang mungkin sesuai untuk *item* untuk kemudian dinilai oleh *item* dengan memberikan *feedback*. *Feedback* ini yang digunakan oleh sistem untuk meng-*update* profil *item*. Dalam perkembangan sistem, *feedback* yang diberikan *item* kemudian disimpan dan diproses untuk menghasilkan rekomendasi yang lebih baik.

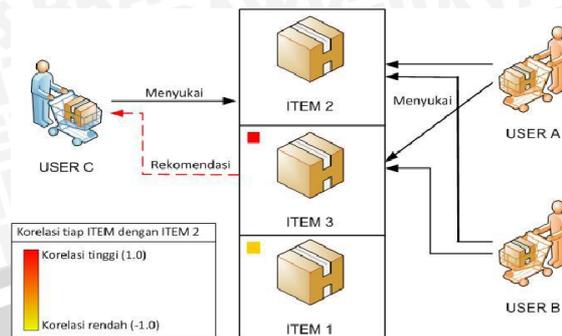
2.2 Collaborative Filtering

Metode *Collaborative Filtering* merupakan salah satu metode yang diterapkan pada sistem rekomendasi untuk memberikan prediksi otomatis terhadap *user* dengan cara mengoleksi informasi dari banyak *user* (Ampazis, 2012). Pada *Collaborative filtering*, rekomendasi yang diberikan berdasarkan keterkaitan (preferensi) antar *item* satu dengan *item* lainnya keterkaitan.

2.2 Item-based Collaborative Filtering

Pendekatan *Collaborative Filtering* pada dasarnya dibagi menjadi dua kategori yaitu *item-based collaborative filtering* yang disebut juga *model-based* dan *user-based collaborative filtering* disebut juga *memory based* (Uyun, Agus., 2011).

Pendekatan *item-based collaborative filtering* memberikan rekomendasi berdasarkan kemiripan antar *item*. Metode ini merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian *rating* terhadap suatu *item* dengan *item* yang pernah di-*rating* *user* lain. *Item* yang telah di *rating* oleh *user* akan menjadi patokan untuk mencari sejumlah *item* lainnya yang berkorelasi dengan *item* yang telah di-*rating* *user*.



Gambar 2. Skema Item-Based Collaborative Filtering

Gambar diatas menggambarkan bagaimana *item* 3 direkomendasikan kepada *user* c yang telah menyukai *item* 2, dimana *item*2 yang disukai *user* c memiliki korelasi yang kuat dengan *item* 3.

Secara umum proses pemberian rekomendasi pada *collaborative filtering* terdiri atas 2 langkah yaitu (Sarwar, at.al., 2001).

1. Penemuan similar item.

Salah satu tahap penting dalam *item-based collaborative filtering* adalah menghitung *similarity* atau kemiripan antar *item* untuk kemudian memilih beberapa *item* yang paling mirip. Ilustrasi proses ini dapat dilihat pada Gambar 2.

	Item 1	Item 1	...	Item 1	...	Item 1	...	Item n-1	Item n
User 1				R		R			
User 1				-		R			
...									
...									
User 1				R		R			
...									
...									
User m-2				R		-			
User m-1				R		R			
...User m				-		-			

Similarity antar item dihitung dengan melihat item oleh pengguna. Pada contoh ini, similaritas sij dari item i dan j didapatkan dengan melihat nilai rating dari user u1, u, dan m-1

Gambar 3. Proses Komputasi Similarity

Persamaan *adjusted cosine similarity* digunakan untuk menghitung nilai kemiripan antar *item*. Perhitungan kemiripan ini merupakan penyempurnaan dari perhitungan kemiripan berbasis vektor. Berdasarkan hasil penelitian Sarwar Badrul yang meneliti tentang algoritma-algoritma yang dapat digunakan untuk menghitung kemiripan, algoritma *adjusted-cosine similarity* merupakan algoritma yang dapat menghasilkan nilai MAE (*mean absolute error*) paling rendah (Sarwar, at.al., 2001).

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u) (R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}} \quad (2.1)$$



Keterangan:

- $sim(i, j)$ = Nilai kemiripan antara *item* i dan *item* j
- $u \in U$ = Himpunan *user* u yang me-rating *item* i dan j
- $R_{u,i}$ = Rating *user* u pada *item* i
- $R_{u,j}$ = Rating *user* u pada *item* j
- \bar{R}_u = Nilai rata-rata rating *user*

Untuk menghitung nilai kemiripan (*similarity*) antar 2 *item*, diperlukan himpunan *user* yang me-rating *item* tersebut. Nilai yang dihasilkan pada persamaan *adjusted-cosine similarity* adalah berkisar antara +1.0 dengan -1.0. *Item* dianggap saling berkorelasi jika nilai *similarity* antara kedua *item* tersebut mendekati +1, begitu juga sebaliknya *item* dianggap tidak berkorelasi apabila nilai *similarity*-nya mendekati -1

2. Penghitungan prediksi.

Setelah mendapatkan nilai kemiripan, maka langkah selanjutnya adalah proses perhitungan prediksi. Proses prediksi yang dilakukan adalah dengan memperkirakan nilai *rating* dari *user* terhadap suatu *item* yang belum pernah di-rating sebelumnya oleh *user* tersebut. Algoritma yang dapat digunakan untuk mendapatkan prediksi dari suatu *item* yaitu (Sarwar, at.al., 2001):

Algoritma *Weighted Sum* digunakan untuk menghitung prediksi. Algoritma ini mendapatkan nilai prediksi dengan menghitung total *rating* yang diberikan terhadap *item* yang mirip dengan *item* yang ingin diprediksi. Teknik ini memprediksi *item* j untuk *user* u dengan menghitung jumlah *rating* yang diberikan oleh *user* terhadap *item* yang berkorelasi dengan *item* j . Setiap *rating* yang diberikan *user* pada *item* yang berkorelasi dengan *item* j akan dikalikan dengan nilai kemiripannya. Kemudian dibagi dengan jumlah nilai absolut kemiripan seluruh *item* yang berkorelasi.

$$P(u, j) = \frac{\sum_{i \in I} (R_{u,i} * S_{i,j})}{\sum_{i \in I} |S_{i,j}|} \quad (2.2)$$

Keterangan:

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

Prediksi dengan persamaan *weighted sum* diperoleh dengan menghitung total keseluruhan dari perkalian antara *rating user* terhadap suatu *item* dan nilai *similarity Item* tersebut. Selanjutnya menghitung total keseluruhan dari nilai *similarity Item* yang mirip dengan *item* yang akan prediksi langkah terakhir yaitu membagi langkah pertama dengan langkah kedua, yang mana akan menghasilkan nilai prediksi untuk *user* terhadap *item*.

2.3 DDC(Dewey Decimal Classification)

DDC adalah salah satu sistem klasifikasi berdasarkan subjek. Pada pengembangannya, simbol notasi pada sistem klasifikasi ini menggunakan sistem desimal angka Arab (Subrata, 2009). *Iteman* bilangan desimal untuk membagi semua bidang ilmu pengetahuan menjadi sepuluh kelas utama. Bidang ilmu tersebut tertulis pada Tabel 1.

Tabel 1. Klasifikasi Dewey Decimal Number (DDC)

Kode	Keterangan
000	Komputer, Informasi dan Referensi Umum
100	Filsafat dan Psikologi
200	Agama
300	Ilmu Sosial
400	Bahasa
500	Sains dan Matematika
600	Teknologi
700	Kesenian dan Rekreasi
800	Sastra
900	Sejarah dan Geografi

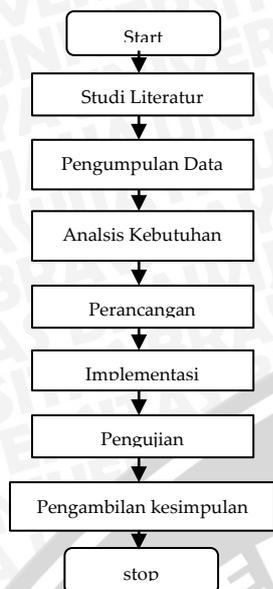
2.4 Skala Guttman

Skala Guttman merupakan skala yang menginginkan tipe jawaban tegas, seperti jawaban benar-salah, ya-tidak, pernah-tidak pernah, positif-negative, tinggi-rendah, baik-buruk, dan seterusnya. Pada skala Guttman, hanya ada dua interval, yaitu setuju dan tidak setuju (Babbie, 2012).

Skala Guttman dapat dibuat dalam bentuk pilihan ganda maupun daftar checklist. Untuk jawaban positif seperti benar, ya, tinggi, baik, dan semacamnya diberi skor 1; sedangkan untuk jawaban negative seperti salah, tidak, rendah, buruk, dan semacamnya diberi skor 0 (Babbie, 2012).

3. METODOLOGI

Metodologi penelitian menjelaskan mengenai langkah-langkah yang akan digunakan dalam pembuatan aplikasi sistem rekomendasi pencarian buku menggunakan *Item-based Collaborative Filtering*. Gambar 4 berikut ini menunjukkan langkah-langkah yang dilakukan dalam penelitian skripsi.



Gambar 4. Diagram alir Penelitian.

4. ANALISIS DAN PERANCANGAN

Pada tahap ini akan dijelaskan perancangan sistem rekomendasi yang akan dibangun dan dijabarkan dalam sebuah diagram alir sesuai dengan analisis yang dilakukan sebelumnya.

4.1 Analisis Kebutuhan Data

Jumlah data yang terhimpun sebanyak 4000 data buku, data mahasiswa yang terhimpun sebanyak 4131 data, peminjaman sebanyak 31 Data transaksi. Untuk detail data transaksi peminjaman dapat dilihat pada lampiran C.1. Pada pembuatan aplikasi sistem rekomendasi ini keseluruhan data yang dibutuhkan dapat dijabarkan secara umum pada Tabel 2.

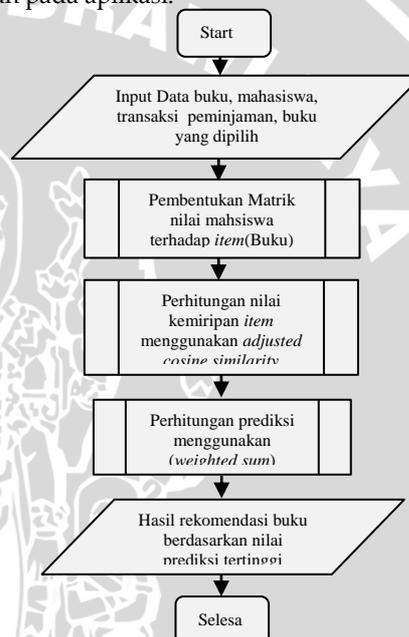
Tabel 2. Analisis Kebutuhan Data

Data	Keterangan
Mahasiswa	Mahasiswa yang menggunakan sistem untuk memperoleh rekomendasi buku. Data ini mencakup nomor mahasiswa/NIM (Nomor Induk Mahasiswa, nama, jurusan, dan program studi).
Buku	Buku berisi data buku-buku yang akan direkomendasikan dan dipinjam oleh mahasiswa. Data ini mencakup id buku, judul buku, jumlah eksemplar, tanggal <i>input</i> , ISBN, penerjemaah, editor, penulis, cetakan, penerbit, tahun terbit, kota, jumlah halaman, panjang lebar, subjek, daftar isi/pembahasan dan catatan umum, dan nomor rak.

Transaksi Peminjaman	Peminjaman berisi catatan peminjaman mahasiswa terhadap suatu buku. Data ini nantinya akan dikonversi sebagai data penilaian. Data peminjaman ini akan dijadikan penilaian mahasiswa terhadap suatu buku.
----------------------	---

4.2 Diagram Alir Sistem Rekomendasi Pencarian Buku Diperustakaan

Sistem rekomendasi buku menggunakan metode *Item-Based Collaborative Filtering* ini sendiri digambarkan pada diagram alir perancangan sistem. Gambar 5 Berikut ini menunjukkan tahapan-tahapan proses sistem rekomendasi buku yang berjalan pada aplikasi.



Gambar 5. Diagram Alir Sistem Rekomendasi Pencarian Buku dengan Metode *Item-based Collaborative Filtering*.

Tahapan-tahapan dalam sistem rekomendasi buku dengan menggunakan *item-base collaborative filtering* dimulai dengan menginputkan buku, data mahasiswa, data peminjaman, dan data buku. Kemudian dilanjutkan dengan proses membentuk matrik penilaian mahasiswa terhadap buku yang berisikan data buku dan data mahasiswa yang satu jurusan. Kemudian disimpan kedalam database. Nantinya data dari matriks ini akan digunakan untuk menghitung kemiripan antar *user*. Karena metode yang digunakan adalah *Item-Based Collaborative Filtering*, maka nantinya sistem akan menghitung kemiripan berdasarkan kesamaan buku yang pernah dipinjam oleh mahasiswa lain dengan menggunakan algoritma *adjusted cosine similarity*. Dapat lihat persamaan 2.1.

Setelah menemukan kemiripan antar buku, maka sistem akan menghitung nilai prediksi untuk menentukan buku apa saja yang sesuai untuk direkomendasikan dengan menggunakan algoritma *weighted sum* sehingga menghasilkan nilai prediksi yang paling tinggi. Persamaan dari algoritma *weighted sum* bisa dilihat pada persamaan 2.2.

Sistem selesai ketika sistem mengeluarkan output berupa nilai prediksi tertinggi. Nilai tersebut yang dijadikan hasil rekomendasi.

5. IMPLEMENTASI

5.1 Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam perancangan sistem ini adalah sebagai berikut:

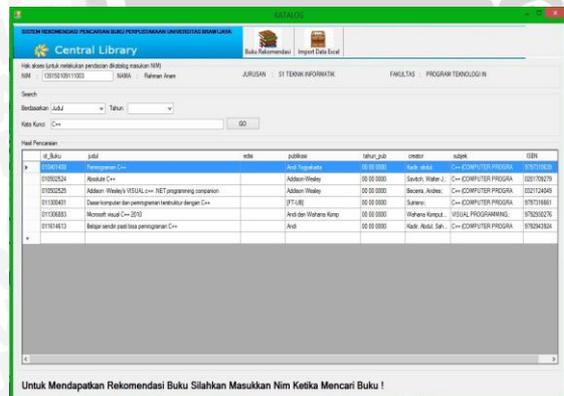
1. *Prosesor Intel(R) Core i3 CPU M350 @2.27GHz(4 CPUs)*
2. *RAM 4096 MB*
3. *Harddisk 300 GB*
4. *Monitor 14"*.
5. *Mouse*
6. *Keyboard*

5.2 Lingkungan Perangkat Lunak

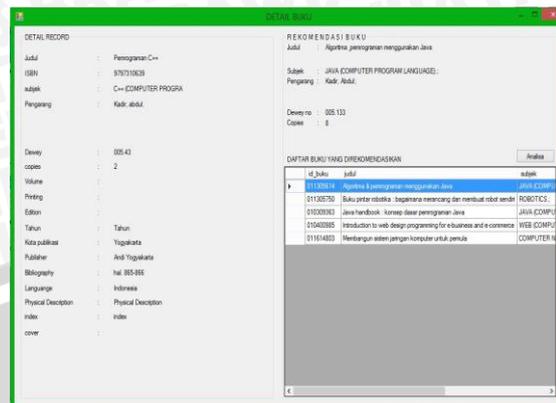
Adapun spesifikasi perangkat lunak yang digunakan pada penelitian ini adalah:

1. *Operating System Windows 8.1 64-bit.*
2. *Visual Studio 2010*
3. *Bahasa Pemrograman C#*
4. *Database MySql*

5.3 Implementasi Antarmuka



Gambar 6. Implementasi Antarmuka Pencarian



Gambar 7. Tampilan Detail Buku dan Hasil Rekomendasi

6. PENGUJIAN

Ada beberapa pengujian yang dilakukan pada penelitian ini untuk mengetahui tingkat akurasi dan efisiensi waktu eksekusi sistem. Adapun pengujian diantaranya:

6.1 Pengujian MAE (Mean Absolute Error)

Pengujian MAE ini bertujuan yaitu menghitung tingkat *error* dari hasil rekomendasi yang diberikan system, dengan menggunakan persamaan dibawah ini:

$$MAE = \frac{\sum_{i=1}^n |p_i - q_i|}{N}$$

Keterangan:

MAE = nilai rata-rata kesalahan hitungan

N = jumlah *item* yang dihitung

p_i = nilai prediksi *item* ke i

q_i = nilai *rating* sebenarnya *item* ke i

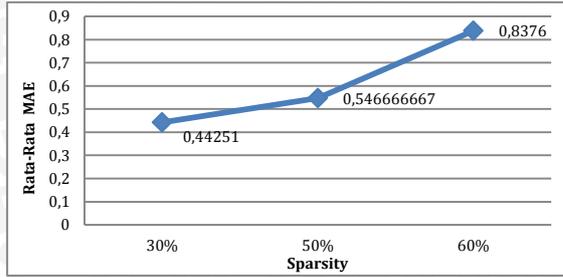
Semakin rendah nilai MAE yang didapat, maka sistem semakin akurat dalam memprediksi *rating* untuk setiap *user*.

Pengujian ini dilakukan pada 5 mahasiswa yang diambil secara acak dengan tingkat kekosongan data transaksi yang berbeda yang bisa disebut *sparsity*. Tingkatan *sparsity* data yang diujikan diantaranya 30%, 50%, 60%.

Adapun Tabel 3 merupakan hasil pengujian akurasi MAE

Tabel 3. Tabel Hasil pengujian MAE

Tingkat <i>sparsity</i>	Percobaan mahasiswa ke-					Rata-rata
	Mhs 1	Mhs 2	Mhs 3	Mhs 4	Mhs 5	
30%	0,4398	0,56475	0,4895	0,267	0,4515	0,44251
50%	0,4255	0,5485	1	0,295	0,464333	0,546667
60%	1	0,594	1	1	0,594	0,8376



Gambar 8 Grafik Hasil Pengujian MAE

Dapat dilihat hasil pengujian dataset yang dilakukan pada data uji dengan tingkat *sparsity* sebesar 30% 50% 60%. Dari hasil pengujian dapat diketahui bahwa nilai rata-rata MAE yang terendah adalah 0,44251 pada tingkat *sparsity* 30% sedangkan nilai rata-rata MAE yang tertinggi adalah 0,8376 pada tingkat *sparsity* 60%. Hal ini menunjukkan bahwa pada pengujian tingkat *sparsity* dimana nilai MAE akan meningkat ketika tingkat kekosongan data dinaikkan. Peningkatan ini disebabkan karena ketika data dalam keadaan sparse yang tinggi jumlah data *rating* yang berkurang akan mempengaruhi proses pertimbangan dalam pemberian prediksi.

6.2 Pengujian Akurasi relevansi presisi dengan DDC(Dewey Decimal Clasification)

Pengujian akurasi ini akan melakukan perbandingan relevansi antara nomor DDC buku yang direkomendasikan dengan nomer DDC buku yang dipilih. Pengujian ini bertujuan untuk menguji tingkat akurasi. Hasil pengujian akurasi terdapat pada Tabel 4.

Tabel 4. Hasil pengujian akurasi relevansi DDC

No	Mahasiswa	Jumlah Buku Yang Direkomendasikan	Jumlah Buku Yang Relevan dengan DDC	Jumlah Buku Yang Tidak Relevan dengan DDC	Precision relevan dengan DDC
1	12515020011116	5	1	4	0,2
2	12515020011119	4	4	0	1
3	125150207111011	8	3	5	0,375
4	125150200111105	7	4	3	0,571429
5	125150200111130	8	6	2	0,75
				Rata Rata	0,579286

Dari hasil pengujian Relevansi dengan nomor DDC menghasilkan kesimpulan bahwa nilai rata-rata precision untuk buku yang direkomendasikan sistem kepada *user* dengan rata-rata 0,579286. Dari hasil nilai precision tersebut dapat dikatakan bahwa sistem rekomendasi buku perpustakaan dengan metode *item-based collaborative filtering* memberikan

hasil rekomendasi buku yang relevan dengan *user* sebanyak 57%.

6.3 Pengujian Waktu Eksekusi Sistem

Pada pengujian waktu eksekusi dataset dipilih secara acak dengan jumlah tertentu. Pengujian dilakukan berdasarkan dua parameter yaitu pengujian berdasarkan jumlah data buku dan pengujian berdasarkan tingkat *sparsity*. Pada setiap parameter dilakukan percobaan sebanyak 5 kali. Berikut merupakan prosedur pengujian untuk tiap-tiap parameter yang digunakan :

1. Berdasarkan Jumlah Item.

Proses pengujian ini dilakukan untuk setiap data *item*/buku yang telah ditentukan jumlahnya, yaitu berjumlah 700 *item*, 1000 *item*, 2000 *item*, 4000 *item*. Berikut hasil pengujian berdasarkan jumlah *item* dapat dilihat pada tabel 5.

Tabel 5. Pengujian waktu eksekusi berdasarkan jumlah item

Jumlah Item/Buku	Percobaan Ke-					Rata-rata (detik)
	1	2	3	4	5	
700	4,541	4,583	4,270	4,281	4,578	4,451
1000	17,700	18,704	18,685	18,938	18,652	18,536
2000	34,922	35,158	35,044	35,153	35,495	35,154
4000	70,101	70,131	69,523	70,165	69,550	69,894



Gambar 9 Pengujian Waktu Eksekusi Berdasarkan Jumlah Item

Pada Tabel dan gambar diatas menjelaskan pengujian waktu eksekusi yang dibutuhkan untuk setiap pengujian yang dilakukan. Dari Hasil pengujian secara keseluruhan, waktu eksekusi meningkat jika jumlah *item* yang diolah bertambah banyaknya. Sehingga pengujian ini menghasilkan waktu tercepat yaitu ketika memproses data *item*/buku yang berjumlah 700 *item* yaitu dengan rata-rata 4,451 detik dan waktu terlama yaitu ketika memproses data *item* dengan total 4000 *item* yaitu dengan rata-rata 69,894 detik.

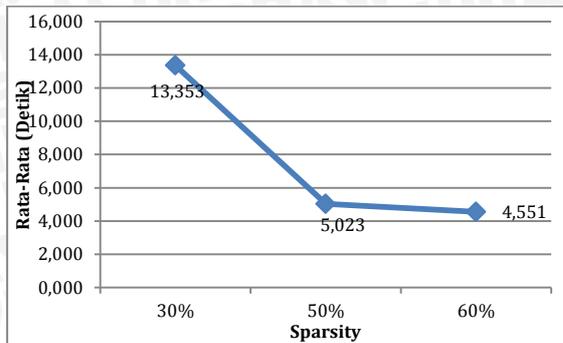
2. Berdasarkan Tingkat Sparsity.

Tahapan pengujian dilakukan dengan mengosongkan dataset peminjaman secara acak dengan tingkat *sparsity* sebanyak 30% 50% dan 60% dengan menggunakan data *item*/buku sebanyak 700 buku. Berikut hasil pengujian

berdasarkan tingkat *sparsity* dapat dilihat pada Tabel 6.

Tabel 6. Pengujian waktu eksekusi berdasarkan *sparsity* data

<i>Sparsity</i> %	Percobaan Ke-					Rata-rata
	1	2	3	4	5	
30%	13,399	13,342	13,448	13,277	13,300	13,353
50%	5,110	5,013	4,992	4,980	5,023	5,024
60%	4,557	4,533	4,549	4,547	4,573	4,552



Gambar 10. Hasil Pengujian Waktu Eksekusi Berdasarkan *Sparsity* Data

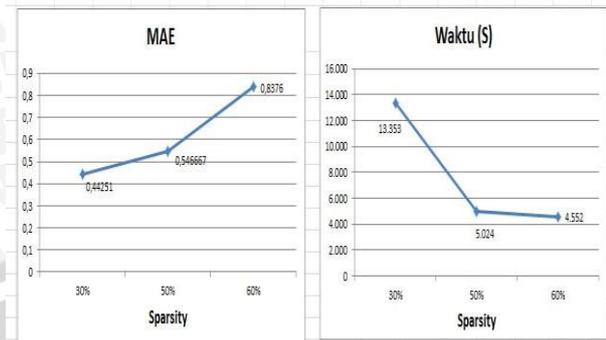
Tabel dan Gambar diatas menunjukkan pengaruh tingkat *sparsity* item pada waktu yang dibutuhkan untuk mengeksekusi sistem. Pada pengujian ini diketahui bahwa tingkat *sparsity* dataset berbanding terbalik dengan waktu eksekusi yang dibutuhkan. Dimana semakin banyak *item* yang belum di-*rating* (*sparsity* tinggi) maka waktu yang dibutuhkan untuk mengeksekusi semakin kecil. Hal tersebut terlihat pada pengujian ketika tingkat *sparsity*-nya mencapai 60% waktu yang dibutuhkan untuk mengeksekusi sistem rata-rata 4,552 detik.

6.4 Perbandingan Rata-Rata MAE dengan Waktu Eksekusi Sistem

Dataset dengan tingkat *sparsity* rendah memang membutuhkan waktu eksekusi yang cukup lama dibandingkan dengan tingkat *sparsity* yang lain, namun akan menghasilkan tingkat akurasi yang tinggi karena banyaknya data transaksi peminjaman yang diproses. Untuk leih jelasnya perbandingan rata-rata MAE dengan waktu eksekusi sistem dapat dilihat pada Tabel 7 dan Gambar 11 dibawah ini.

Tabel 7. Perbandingan Hasil MAE dengan Waktu Eksekusi Sistem

<i>Sparsity</i>	MAE	Waktu (S)
30%	0,44251	13,353
50%	0,546667	5,024
60%	0,8376	4,552



Gambar 11. Perbandingan Hasil Uji MAE dengan Hasil Uji Waktu Eksekusi Sistem

7. KESIMPULAN

Berdasarkan hasil penelitian, maka dapat disimpulkan bahwa:

1. Hasil pengujian yang ditunjukkan pada Tabel 3 membuktikan bahwa sistem dapat mengatasi tingkat *sparsity* data hingga 60% dengan nilai MAE 0,8376. semakin tinggi tingkat *sparsity* data maka nilai MAE juga akan semakin tinggi sehingga menghasilkan tingkat akurasi yang rendah.
2. Hasil pengujian yang ditunjukan pada Tabel 4 menggambarkan akurasi apabila dibandingkan dengan nomor DDC menghasilkan tingkat akurasi 57% dari buku yang direkomendasikan dengan data *sparsity* 30%.
3. Waktu eksekusi yang dibutuhkan dipengaruhi oleh jumlah data dan tingkat *sparsity*. Pada Tabel 3 dapat disimpulkan semakin besar jumlah data yang digunakan maka semakin besar juga waktu eksekusi yang dibutuhkan sedangkan untuk tingkat *sparsity* pada Tabel 6 dapat disimpulkan semakin tinggi *sparsity* maka waktu eksekusi yang dibutuhkan semakin kecil.

8. SARAN

Adapun saran-saran yang diajukan oleh penulis untuk sistem rekomendasi ini adalah:

1. Berdasarkan kesimpulan yang dijelaskan di atas bahwa sistem ini masih kurang tahan terhadap *sparsity* data yang melebihi 60%. Oleh sebab itu, penulis menyarankan untuk penelitian selanjutnya dapat menggunakan teknik *smoothing* pada dataset, sehingga penerapan teknik *smoothing* pada keadaan ini juga menyebabkan proses prediksi lebih banyak diisi oleh mesin rekomendasi dari pada jumlah penilaian yang sebenarnya.
2. Waktu eksekusi sistem paling tinggi adalah ketika jumlah 4000 buku dengan menghasilkan waktu 69 detik. Itu pun dikarenakan adanya proses *filtering* berdasarkan jurusan tiap

mahasiswa di awal proses. Kedepannya diharapkan adanya solusi yang lebih optimal dalam masalah waktu eksekusi.

3. Diharapkan adanya penggabungan dua metode atau lebih untuk dapat memaksimalkan hasil rekomendasi yang diperoleh.

9. DAFTAR PUSTAKA

Adomavicius., & Tuzhilin., 2005. *Toward the Next Generation of Recommender Systems: A survey of the state-of-the-art and possible Extentions*. IEEE Transaction of knowledge and Data Engineering, vol17.

Ampazis, Nicholas., 2012. *Collaborative Filtering via Concept Decomposition on the Netflix Dataset*. Tersedia di:<http://www.feeds2.com/netflix/ampazis_ECAI08.pdf> [Diakses 12 Februari 2016].

Babbie, Earl R., 2012. *The Practice of Social Research*. Tersediadi:<<http://books.google.co.id/books?id=kaz a3qSULoC&dq=Likert+and+Guttman+scaling>>, [Diakses 12 Februari 2016].

Budianto, Teguh., 2012. *Rancang Bangun Movie Recommender System Dengan Metode Cluster-Based Smoothing Collaborative filtering*. Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim, Riau.

Fitri, Aditya., 2015. *Sistem rekomendasi mata kuliah pilihan mahasiswa dengan content-based filtering dan collaborative filtering* (studi kasus: universitas brawijaya). Program Studi Informatika/Illmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Hakim, Irfan Aris Nur., 2010. *Sistem Rekomendasi Film Berbasis Web Menggunakan Metode Item-Based Collaborative Filtering Berbasis K-Nearest Neighbor*. Program Studi Ilmu Komputer, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam Universitas Pendidikan Indonesia.

Handrico, Aritha. 2012. *Sistem Rekomendasi Buku Perpustakaan Fakultas Sains dan Teknologi dengan Metode Collaborative Filtering*. Teknik Informatika, Fakultas Sains Dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim, Riau Pekanbaru.

Michael, P.O'Mahoney., Barry, Smyth., 2007. *A Recommender System for On-line Course Enrolment: An Initial Study*. School of Computer Science and Informatics, University College, Dublin.

Pazzani, Michael J., 2007. *Content-based Recommendation System*. Rutgers University.

Rachmat, Tsalaatsa., 2015 *Sistem Rekomendasi Film Menggunakan Item-Based Clustering Hybrid Method*. Program Studi Informatika/Illmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Sanjung, Ariyani., 2011. *Perbandingan Semantic Classification dan Cluster-based Smoothed pada Recommender System berbasis Collaborative filtering*. Fakultas Teknik Informatika, Institut Teknologi Telkom.

Sarwar, Badrul., George Karypis., Joseph Konstan., & Jhon Riedl., 2001. *Item-based Collaborative Filtering Recommendation Algorithms*. Departemen of Computer Science and Engineering, University of Minnesota.

Subroto, G. (2009). *Klasifikasi Bahan Pustaka*. Pustakawan Perpustakaan UM, 1-13.

Uyun, Shofwatul., Imam Fahrurrozi., & Agus Mulyanto., 2011. *Item Collaborative Filtering untuk Rekomendasi Pembelian Buku secara Online*. Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Sunan Kalijaga.

Wimmer, D. Roger, & Joseph R. Dominick., 2012. *Mass Media Research: An Introduction*. Halaman 55. Tersediadi:<<http://books.google.com/books?id=ay5l KmAw2UcC>>, [Diakses 14 Februari 2016].