

**SISTEM REKOMENDASI PENCARIAN BUKU
DIPERPUSTAKAAN MENGGUNAKAN METODE *ITEM-BASED
COLLABORATIVE FILTERING***

{Studi Kasus: Perpustakaan Universitas Brawijaya}

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rahman Anam
NIM: 135150109111003



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017

PENGESAHAN

SISTEM REKOMENDASI PENCARIAN BUKU DIPERPUSTAKAAN MENGGUNAKAN
METODE *ITEM-BASED COLLABORATIVE FILTERING*

{Studi Kasus: Perpustakaan Universitas Brawijaya}

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Rahman Anam

NIM: 135150109111003

Skripsi ini telah diuji dan dinyatakan lulus pada
29 Desember 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Rekyan Regasari Mardi Putri, S.T, M.T

NIK: 2011027704142001

Agus Wahyu Widodo, S.T, M.Cs

NIP: 197408052001121001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

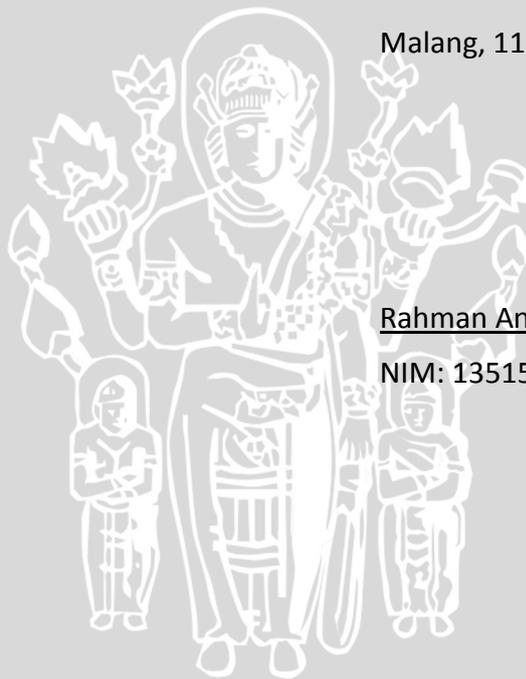
NIP: 197105182003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 11 November 2016



Rahman Anam

NIM: 135150109111003

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Allah SWT, karena atas rahmat dan hidayahNya, penulis dapat menyelesaikan skripsi yang berjudul “Sistem Rekomendasi Pencarian Buku di perpustakaan Menggunakan Metode *Item-based Collaborative Filtering*” ini dengan baik. Adapun maksud dan tujuan penulisan skripsi ini adalah sebagai syarat untuk menyelesaikan studi program sarjana Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya Malang.

Menyadari penulisan ini tidak lepas dari bantuan berbagai pihak, maka penulis mengucapkan terima kasih sebesar-besarnya kepada pihak-pihak yang telah membantu kelancaran dalam pengerjaan maupun penyusunan laporan skripsi ini yaitu kepada:

1. Ibu Rekyan Regasari Mardi Putri, S.T, M.T selaku dosen pembimbing 1 dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku dosen pembimbing 2 dan selaku Ketua Program Studi Informatika/Illmu Komputer Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya yang telah bersedia meluangkan waktu memberikan arahan, bimbingan serta saran selama penyusunan skripsi ini.
2. Ibu, Alm Bapak, dan seluruh keluarga, Penulis mengucapkan terima kasih atas dukungan, doa serta kasih sayang yang sangat luar biasa diberikan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya Malang.
4. Bapak Drs. Achmad Basuki, S.T., M.MG., Ph.D. selaku Dosen pembimbing akademik dari penulis.
5. Bapak dan Ibu Dosen serta seluruh Staf Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya yang telah memberikan ilmunya serta arahan selama masa perkuliahan.
6. Teman-teman Seleksi Alih Program (SAP) 2013 FILKOM atas motivasi, bantuannya.
7. Seluruh staf perpustakaan Universitas Brawijaya, perpustakaan FILKOM dan Universitas Brawijaya yang membantu penulis memperoleh referensi.

Penulis juga menyadari bahwa skripsi ini tidak luput dari kekurangan. Oleh karena itu penulis mengharapkan kritik dan saran dari pembaca untuk kesempurnaan skripsi ini. Saran dan kritikan yang bersifat membangun dapat disampaikan kepada penulis melalui email rahmananam36@gmail.com. Sebagai penutup, penulis mengucapkan banyak terima kasih, dan semoga skripsi ini dapat memberikan manfaat khususnya bagi penulis sendiri dan juga bagi kita semuanya.

Malang, 11, November 2016

Rahman Anam

ABSTRAK

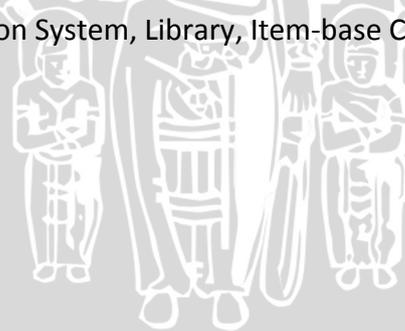
Daya baca mahasiswa di Indonesia masih sangat rendah. Banyak faktor yang melatarbelakangi hal ini, salah satunya mahasiswa kekurangan referensi mengenai buku apa yang sesuai dengan minat maupun kebutuhan mereka. Perpustakaan Universitas Brawijaya adalah perpustakaan yang memiliki literatur yang cukup 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.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR PERSAMAAN	xiii
DAFTAR KODE PROGRAM	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Sistem Rekomendasi.....	10
2.2.1 Arsitektur Sistem Rekomendasi.....	11
2.2.2 <i>Content Based Filtering</i>	12
2.2.3 <i>Collaborative Filtering</i>	12
2.3 <i>Dewey Decimal Classification (DDC)</i>	17
2.4 Skala Penilaian	18
2.4.1 Skala <i>Likert</i>	18
2.4.2 Skala <i>Guttman</i>	18
2.4.3 Skala <i>Rating (Rating Scale)</i>	18
2.5 Evaluasi Sistem Rekomendasi.....	18

2.5.1 Mean Absolute Error (MAE)	19
2.5.2 Presisi (<i>Precision</i>)	19
2.5.3 <i>Sparsity</i>	19
2.6 Perpustakaan Universitas Brawijaya	20
BAB 3 METODOLOGI	21
3.1 Diagram Alir Penelitian	21
3.2 Studi Literatur	22
3.3 Pengumpulan Data	22
3.4 Analisis Kebutuhan	22
3.5 Perancangan	23
3.6 Implementasi	24
3.7 Pengujian	24
3.8 Pengambilan Kesimpulan	25
BAB 4 ANALISIS DAN PERANCANGAN	26
4.1 Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras	26
4.2 Analisis Kebutuhan Data	27
4.3 Perancangan Sistem Rekomendasi	28
4.3.1 Diagram Blok Sistem Rekomendasi Pencarian Buku Perpustakaan	28
4.3.2 Alir Perancangan Sistem	29
4.3.3 Perancangan Perhitungan	32
4.3.4 Perancangan Tabel Data	36
4.3.5 Perancangan <i>Entity Relationship Diagram</i> (ERD)	38
4.3.6 Perancangan Antarmuka	39
BAB 5 IMPLEMENTASI	42
5.1 Spesifikasi sistem	42
5.1.1 Lingkungan Perangkat Keras	42
5.1.2 Lingkungan Perangkat Lunak	43
5.2 Batasan Implementasi	43
5.3 Implementasi Basis Data	43
5.4 Implementasi Algoritma	44
5.4.1 Implementasi Algoritma Pembentukan Matriks	44



5.4.2 Implementasi Algoritma Perhitungan Nilai Kemiripan(<i>Similarity</i>)	45
5.4.3 Implementasi Algoritma Perhitungan Nilai Prediksi	46
5.5 Implementasi Antarmuka	47
5.5.1 Halaman Utama Pencarian Buku	47
5.5.2 Tampilan Detail Buku Dan Hasil Rekomendasi Buku	47
5.5.3 Halaman Hasil Rekomendasi	48
5.5.4 Halaman Analisis Perhitungan	49
BAB 6 PENGUJIAN	50
6.1 Pengujian MAE (Mean Absolute Error)	50
6.2 Pengujian Akurasi Relevansi presisi dengan DDC (<i>Dewey Decimal Classifications</i>)	52
6.3 Pengujian Waktu Eksekusi Sistem	53
6.3.1 Hasil Pengujian Waktu Eksekusi Sistem Berdasarkan Jumlah <i>Item</i>	53
6.3.2 Hasil Pengujian Waktu Eksekusi Sistem Berdasarkan Tingkat <i>Sparsity</i>	54
6.4 Analisis Hasil Pengujian	55
6.5 Kesimpulan Hasil Pengujian	56
BAB 7 KESIMPULAN	58
7.1 Kesimpulan	58
7.2 Saran	58
DAFTAR PUSTAKA	59
LAMPIRAN A SKRIP PROGRAM	61
A.1 Skrip Program Perhitungan pada Sistem Rekomendasi	61
A.2 Skrip Program Pencarian Katalog di Menu Utama	65
LAMPIRAN B SURAT PENELITIAN	71
B.1 Surat Pengantar Penelitian di Perpustakaan Universitas Brawijaya	71
B.2 Surat Balasan Perijinan Penelitian di Perpustakaan Universitas Brawijaya	72
LAMPIRAN C DATA DAN HASIL PENELITIAN	73
C.1 Lampiran Data Transaksi Peminjaman	73



C.2 Lampiran Data Hasil Pengujian Relevansi dengan DDC dengan 30% Sparsity Data 74

C.3 Lampiran Data Hasil Pengujian MAE dengan 30% Sparsity Data 75

C.4 Lampiran Data Hasil Pengujian MAE dengan 50% Sparsity Data 76

C.5 Lampiran Data Hasil Pengujian MAE dengan 60% Sparsity Data 76



DAFTAR TABEL

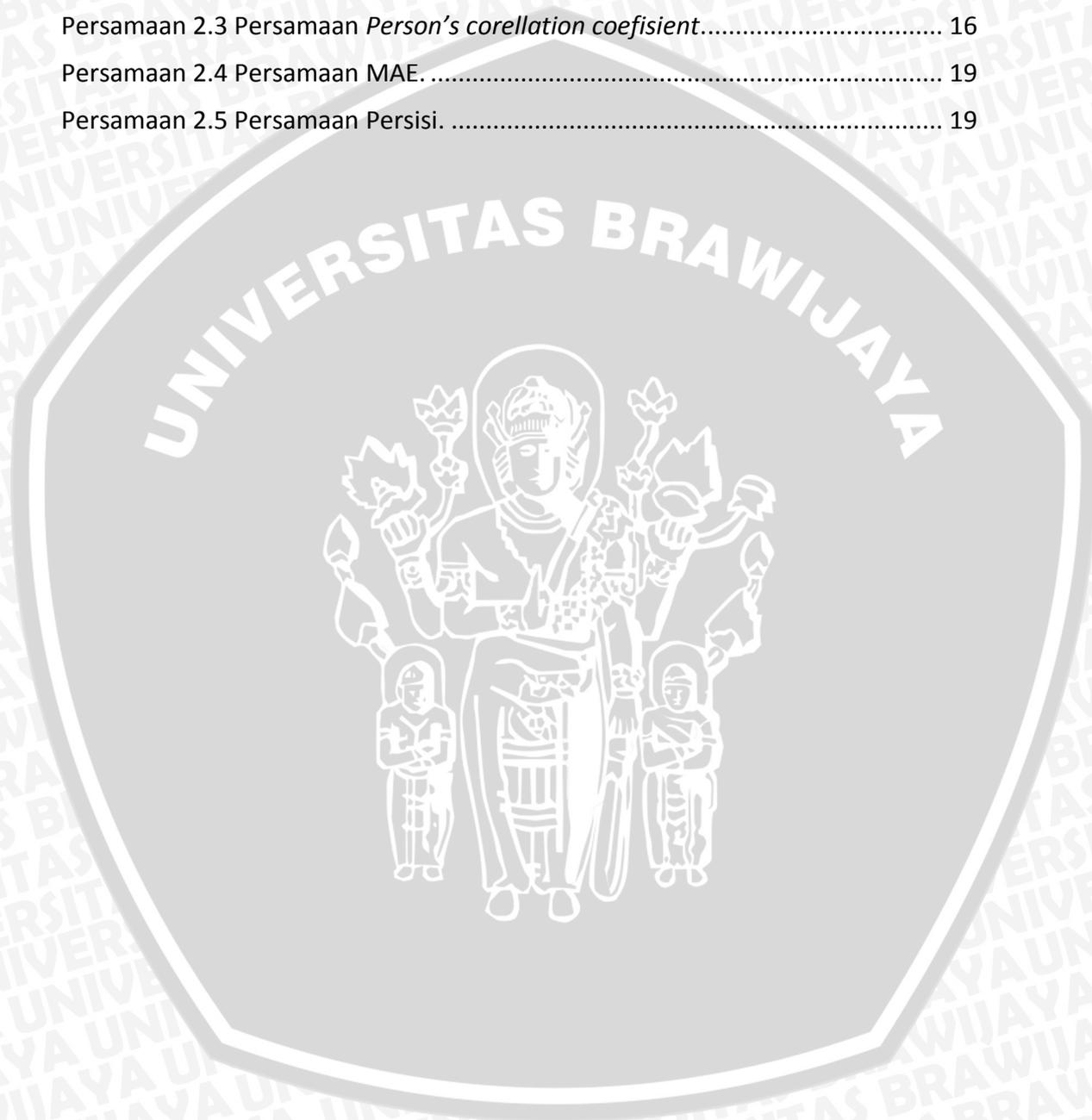
Tabel 2. 1 Kajian Pustaka.....	7
Tabel 2. 2 Contoh Data <i>Rating</i> Dari User	16
Tabel 2. 3 Klasifikasi Dewey Decimal Number (DDC).....	17
Tabel 3. 1 Penentuan Kebutuhan Data Penelitian	22
Tabel 4. 1 Analisis Kebutuhan Data.....	27
Tabel 4. 2 Matriks Hubungan Mahasiswa Terhadap Buku	33
Tabel 4. 3 Tabel Rata-rata penilaian Mahasiswa terhadap Buku	34
Tabel 4. 4 Tabel Hasil Perhitungan Kemiripan Antar Buku	34
Tabel 4. 5 Tabel Hasil Perhitungan Nilai Prediksi.....	35
Tabel 4. 6 Tabel Nilai Top N Prediksi.....	36
Tabel 4. 7 Tabel Mahasiswa	36
Tabel 4. 8 Tabel Buku	36
Tabel 4. 9 Tabel Peminjaman	37
Tabel 4. 10 Tabel Detail Peminjaman	37
Tabel 4. 11 Tabel <i>Rating</i>	38
Tabel 5. 1 Spesifikasi Perangkat Keras	43
Tabel 5. 2 Spesifikasi Perangkat Lunak	43
Tabel 6. 1 Tabel Hasil Pengujian MAE (<i>Mean Absolute Error</i>).....	51
Tabel 6. 2 Tabel Hasil Pengujian dengan Nomor DDC	52
Tabel 6. 3 Tabel Hasil Pengujian Waktu Eksekusi Berdasarkan Jumlah <i>Item</i>	53
Tabel 6. 4 Tabel Hasil Pengujian Waktu Eksekusi Berdasarkan Tingkat <i>Sparsity</i> .	54
Tabel 6. 5 Perbandingan Rata-Rata MAE Dengan Waktu Eksekusi Sistem.....	56

DAFTAR GAMBAR

Gambar 2. 1 Sistem Rekomendasi pada Youtube.com.....	10
Gambar 2. 2 Arsitektur Sistem Rekomendasi	11
Gambar 2. 3 Proses <i>Collaborative Filtering</i>	12
Gambar 2. 4 Skema <i>Item-based Collaborative Filtering</i>	13
Gambar 2. 5 Proses Komputasi <i>Similarity</i>	14
Gambar 3. 1 Diagram Alir Penelitian.....	21
Gambar 3. 2 Diagram Blok Sistem Rekomendasi Pencarian Buku Perpustakaan. 28	
Gambar 4. 1 Pohon Analisis dan Perancangan	26
Gambar 4. 2 Diagram Alir Sistem Rekomendasi Buku Perpustakaan Dengan Metode <i>Item-Based Collaborative Filtering</i>	29
Gambar 4. 3 Diagram Alir Perancangan Pembentukan Matrik	30
Gambar 4. 4 Diagram alir Perhitungan Nilai Kemiripan.....	31
Gambar 4. 5 Diagram Alir Perhitungan Nilai Prediksi	32
Gambar 4. 6 Rancangan ERD Sistem Rekomendasi Pencarian Buku	38
Gambar 4. 7 Halaman Pencarian Buku	39
Gambar 4. 8 Tampilan Detail Buku dan Hasil Rekomendasi.....	40
Gambar 4. 9 Halaman Hasil Rekomendasi.....	40
Gambar 4. 10 Tampilan Analisis Perhitungan	41
Gambar 5. 1 Pohon Implementasi	42
Gambar 5. 2 Implementasi Diagram ER Sistem	44
Gambar 5. 3 Halaman Utama Pencarian dan Hasil Rekomendasi Buku	47
Gambar 5. 4 Halaman Detail Buku	48
Gambar 5. 5 Halaman Hasil Rekomendasi	48
Gambar 5. 6 Halaman Analisis Perhitungan.....	49

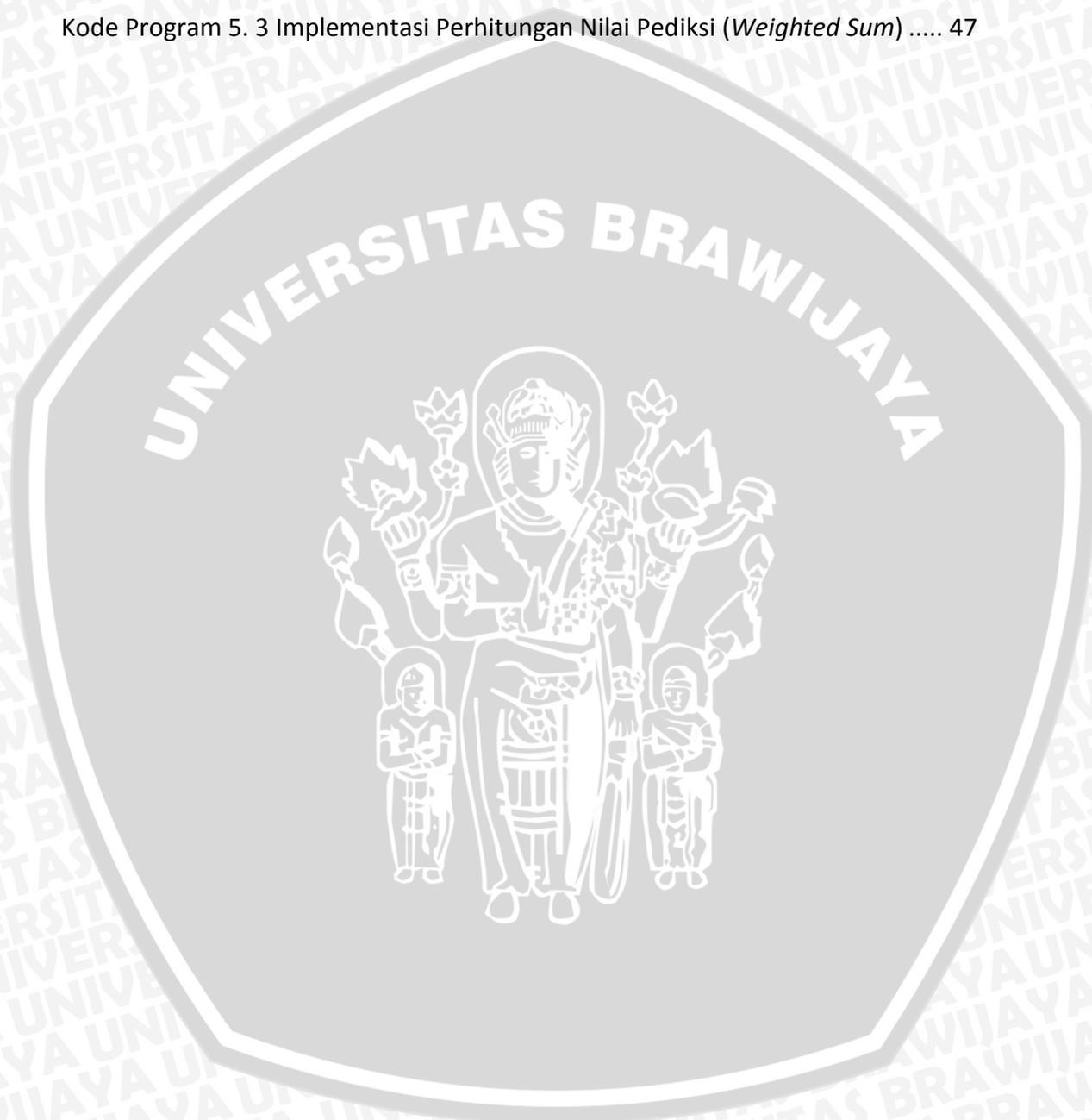
DAFTAR PERSAMAAN

Persamaan 2.1 Persamaan <i>Adjusted Cosine Similarity</i>	15
Persamaan 2.2 Persamaan <i>Weighted Sum</i>	15
Persamaan 2.3 Persamaan <i>Person's corellation coefisient</i>	16
Persamaan 2.4 Persamaan MAE.....	19
Persamaan 2.5 Persamaan Persisi.....	19



DAFTAR KODE PROGRAM

Kode Program 5. 1 Implementasi Pembentukan Matriks.....	45
Kode Program 5. 2 Implementasi Perhitungan Nilai Kemiripan (<i>Similarity</i>).....	46
Kode Program 5. 3 Implementasi Perhitungan Nilai Pediksi (<i>Weighted Sum</i>)	47



BAB 1 PENDAHULUAN

1.1 Latar belakang

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 Classification*(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 pengkategorianya. Penelitian ini pengkategorianya berdasarkan prediksi perilaku *user* terhadap suatu *item*(buku).

Penelitian tentang sistem rekomendasi sebelumnya telah banyak dilakukan dengan metode dan objek yang berbeda-beda. Pada penelitian yang dilakukan oleh Aditya membahas tentang perbandingan metode *item-based*, *user-based collaborative filtering* dengan *content-base filtering* dalam rekomendasi pemilihan matakuliah dengan studi Kasus Universitas Brawijaya (Aditya, 2015). Selanjutnya penelitian yang dilakukan oleh Rachmat dengan judul "*Sistem Rekomendasi Film Menggunakan Item-Based Clustering Hybrid Method*" penelitian ini menghasilkan rekomendasi film berdasarkan *rating item* dan *rating content* dengan menggabungkan metode IBCF dan *Content-Base Filtering* (Rachmat, 2015). Selanjutnya penelitian sistem rekomendasi dengan judul "*Rancang Bangun Music Recommender System Dengan Metode User-Based Collaborative Filtering*" (Budianto, 2012). Metode *Collaborative Filtering* juga diterapkan oleh Uyun dengan penelitiannya yang berjudul "*Item Collaborative Filtering untuk Rekomendasi Pembelian Buku Secara Online*" penelitian ini menghasilkan rekomendasi pembelian buku kepada *user* (Uyun, et.al., 2012).

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*.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat diuraikan rumusan masalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasi aplikasi sistem rekomendasi pencarian buku perpustakaan menggunakan metode *Item-based Collaborative Filtering*?
2. Bagaimana tingkat akurasi dari penerapan metode *Item-based Collaborative Filtering* pada aplikasi sistem rekomendasi pencarian buku perpustakaan?

1.3 Tujuan

Adapun Tujuan dari penelitian ini sebagai berikut:

1. Merancang dan mengimplementasikan aplikasi sistem rekomendasi pencarian buku perpustakaan dengan metode *Item-Based Collaborative Filtering*
2. Mengetahui hasil uji dari tingkat akurasi sistem rekomendasi yang dibangun yang menggunakan metode *Item-based Collaborative Filtering*.

1.4 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan kemudahan bagi mahasiswa untuk menemukan referensi buku yang sesuai dengan kebutuhan.
2. Memberikan gambaran kepada mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk dimanfaatkan pada penelitian selanjutnya tentang sistem rekomendasi.

1.5 Batasan masalah

Berdasarkan uraian dari latar belakang masalah, maka dapat disimpulkan bahwa ruang lingkup pembahasan dibatasi pada:

1. Penilaian mahasiswa dalam preferensi *user* terhadap *item* digantikan dengan *history* peminjaman buku yang dilakukan oleh mahasiswa,
2. Penilaian mahasiswa menggunakan skala penilaian *Guttman*. Skala penilaian *Guttman* merupakan skala yang menginginkan tipe jawaban tegas, seperti jawaban benar-salah, ya-tidak, pernah-tidak pernah. Pernah melakukan peminjaman diberi skor 1, sedangkan untuk jawaban, tidak pernah meminjam diberi skor 0 (Babbie, 2012).
3. Data transaksi peminjaman yang digunakan adalah data *Dummy* dengan 31 Data transaksi peminjaman.
4. *Filtering* yang digunakan hanya *filtering* jurusan.

1.6 Sistematika pembahasan

Untuk sistematika pembahasan yang diterapkan oleh penulis antara lain :

BAB 1 PENDAHULUAN

Bab ini Memuat latar belakang yang menjelaskan permasalahan yang dihadapi mahasiswa terutama dalam menemukan buku yang sesuai kebutuhan. serta rumusan masalah, batasan masalah, manfaat penelitian, dan sistematika penulisan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi kajian pustaka, teori-teori sistem rekomendasi dan metode *Item-based Collaborative Filtering* yang diantaranya algoritma *Adjusted Cosine Similarity* dan Algoritma perhitungan prediksi *Weighted Sum* yang akan digunakan pada sistem rekomendasi.

BAB 3 METODOLOGI

Membahas tentang metode yang digunakan dalam penulisan, yang terdiri dari studi literatur, pengumpulan data, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan pengambilan kesimpulan.

BAB 4 ANALISIS DAN PERANCANGAN

Membahas tentang analisis kebutuhan dari perancangan sistem rekomendasi pencarian buku perpustakaan seperti analisis kebutuhan data, kebutuhan perangkat lunak dan kemudian merancang hal-hal yang berhubungan dengan analisis tersebut.

BAB 5 IMPLEMENTASI

Membahas tentang hasil perancangan dari analisis kebutuhan dan implementasi sistem aplikasi sehingga dapat diketahui efektifitas sistem yang dibangun.

BAB 6 PENGUJIAN

Memuat tentang hasil pengujian dan analisis terhadap sistem rekomendasi yang telah direalisasikan sehingga didapat penilaian mengenai akurasi perhitungan dari sistem rekomendasi yang dibangun.

BAB 7 KESIMPULAN

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Berikut merupakan beberapa landasan kepustakaan yang digunakan penulis dalam penelitian untuk memberikan penjelasan tentang permasalahan yang diangkat. Kajian pustaka membahas penelitian sebelumnya tentang metode yang sama namun pada kasus yang berbeda. Dasar teori membahas teori penunjang yang berkaitan dengan penelitian diantaranya sistem rekomendasi, selanjutnya dibahas metode *Collaborative Filtering* yang diantaranya ada *item-based collaborative filtering*, Skala penilaian, dan evaluasi sistem rekomendasi.

2.1 Kajian Pustaka

Sistem rekomendasi saat ini banyak diimplementasikan dalam berbagai bidang, diantaranya adalah dalam pemberian rekomendasi produk pada pengunjung toko *online*, rekomendasi video pada mengunjungi situs Youtube serta rekomendasi film pada Netflix. Maka penelitian usulan ini akan mempelajari beberapa penelitian yang telah dilakukan sebelumnya yaitu berisikan beberapa jurnal yang dijadikan referensi. Jenis yang digunakan untuk mengkaji dari penelitian sebelumnya dan penelitian usulan ini adalah dengan mendekati objek penelitian yang berbeda namun metodenya sama dan juga objek yang sama dengan metode yang berbeda. Dan untuk penjelasan mengenai uraian dari penelitian sebelumnya ditunjukkan pada tabel kajian pustaka 2.1.

Referensi pertama adalah penelitian yang dilakukan oleh Aditya Fitri Hananta Putra pada tahun 2015, dari Universitas Brawijaya Malang telah melakukan studi perbandingan metode dalam sistem rekomendasi dengan studi kasus pemilihan matakuliah pilihan pada Universitas Brawijaya Malang dengan menggunakan metode *Content-based Filtering*, *Item-based Collaborative Filtering* dan *User-based Collaborative Filtering*. Hasil penelitian yang dilakukan menghasilkan performa rekomendasi bernilai 67,63% untuk *content-based filtering*, 88,69% untuk *item-based collaborative filtering*, dan 85,71% untuk *user-based collaborative filtering* (Aditya, 2015).

Referensi yang kedua dari Rachmat Tsalaatsa yang juga dari Universitas Brawijaya melakukan penelitian tentang sistem rekomendasi film menggunakan *Item-Based Clustering Hybrid Method* (ICHM) dengan melakukan kombinasi antara *Collaborative Filtering* dan *Content-Based Filtering*. Didapat nilai MAE terendah pada *cluster* dengan jumlah 100 dan memberikan nilai MAE = 0,739. Sistem rekomendasi menggunakan metode ICHM dapat memberikan prediksi pada *item* baru yang belum pernah di-*rating* sama sekali (Rachmat, 2015).

Referensi selanjutnya merupakan penelitian yang dilakukan Budianto, dari Universitas Komputer Indonesia. Telah melakukan penelitian mengenai sistem rekomendasi musik menggunakan metode *user-based collaborative filtering*, dengan menggunakan algoritma tambahan untuk mengatasi data *Sparsity* atau kekosongan *dataset* sehingga dapat memberikan tingkat akurasi yang lebih tinggi

karena meminimalkan data *Sparsity*. Hasil dari penelitian ini menghasilkan pengujian akurasi rekomendasi berdasarkan pengujian MAE sebesar 0,6713, MAE sendiri memiliki rentang nilai 0-1 (Budianto, 2012).

Aritha Handrico dari Universitas Islam Negeri Sultan Syarif Kasim Riau Pekanbaru melakukan penelitian mengenai sistem rekomendasi buku pada perpustakaan Fakultas Sains dan Teknologi UIN Sultas Syarif Kasim dengan menggunakan metode *Collaborative Filtering* di mana didalam metode itu digunakan pendekatan *Item-based Collaborative Filtering*. Penelitian ini menghasilkan hasil pengujian menggunakan teknik pengujian dengan UAT dan menghasilkan 30% buku yang relevan dengan kebutuhan *user*, sedangkan pengujian dengan MAE memiliki nilai 0,884307 dengan tingkat *Sparsity* mencapai 60%, hasil tersebut kurang akurat karena dipengaruhi banyaknya data *Sparsity* (Handrico, 2012).

O'Mahony dan Smyth dari University College Dublin telah melakukan studi mengenai pengembangan sistem rekomendasi mata kuliah dengan menggunakan *Collaborative Recommender*. Dengan memberikan rekomendasi mata kuliah yang sesuai untuk mahasiswa. Hasil dari penelitian O'Mahony dan Smyth ini memberikan performa *recall* sebesar 66% (Michael & Barry, 2007).



Tabel 2. 1 Kajian Pustaka

No	Judul	Obyek (<i>Input</i>)	Metode (<i>Proses</i>)	Hasil (<i>Output</i>)
1	Sistem Rekomendasi Mata Kuliah Pilihan Mahasiswa Dengan <i>Content-based Filtering</i> Dan <i>Collaborative Filtering</i> (Studi Kasus: Universitas Brawijaya) (Aditya, 2015).	Objek : Rekomendasi Mata Kuliah <i>Input</i> yang digunakan: - Data <i>History</i> pilihan matakuliah mahasiswa - Data matakuliah - Data nilai mahasiswa	Metode : <i>Content-Based Filtering</i> dan <i>Collaborative filtering</i> Proses <i>Content-based Filtering</i> : - Mempersiapkan data - Membentuk <i>decision tree</i> - Membentuk <i>Rules</i> untuk memperoleh rekomendasi matakuliah Proses <i>Item-based Collaborative filtering</i> dan <i>User-based Collaborative filtering</i> : - Menentukan Matriks <i>user</i> . - Melakukan perhitungan kemiripan menggunakan <i>Cosine Similarity</i> . - Melakukan perhitungan nilai prediksi menggunakan <i>weighted sum</i> dan <i>herlocker</i> untuk memberikan rekomendasi	Menghasilkan Daftar matakuliah yang direkomendasikan
2	Sistem rekomendasi film menggunakan <i>item-based clustering hybrid method</i> (Rachmat, 2015).	Objek: Rekomendasi Film <i>Input</i> yang digunakan: - Data <i>Rating user</i> terhadap <i>item</i> movie/len	Metode: <i>Item-based Clustering Hybrid Method</i> Proses : - Pembentukan Matriks - Melakukan <i>Clustering k-mean</i> - Perhitungan <i>Similarity</i> - Melakukan prediksi	Menghasilkan Daftar film yang direkomendasikan

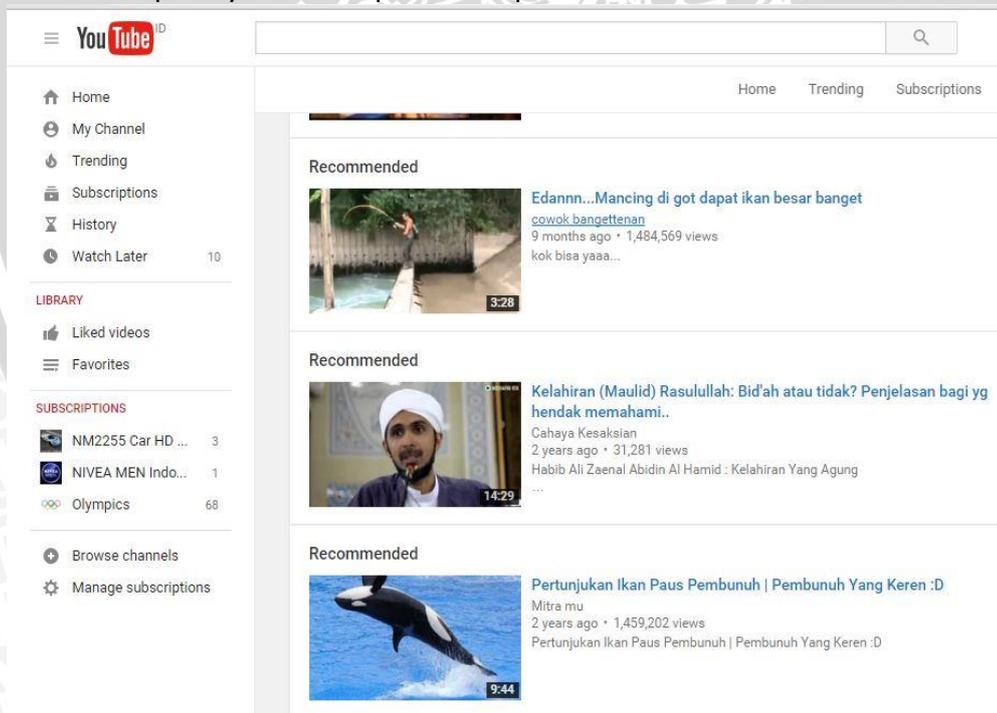
3	Rancang bangun music recommender system dengan metode <i>user-based collaborative filtering</i>] (Budianto, 2012).	<p>Objek: Rekomendasi musik</p> <p><i>Input</i> yang digunakan:</p> <ul style="list-style-type: none"> - <i>Rating user</i> terhadap <i>item</i> - <i>History user</i> Sebelumnya 	<p>Metode : <i>Collaborative Filtering</i> dan <i>K-means Clustering</i></p> <p>Proses :</p> <ul style="list-style-type: none"> - <i>Input data rating</i> - Proses <i>Smoothing</i> Pembentukan <i>Cluster</i> menggunakan <i>K-mean Clustering</i>. - Proses perhitungan menggunakan <i>User-base collaborative filtering</i> - Menentukan nilai prediksi dan memberikan rekomendasi 	Menghasilkan Daftar Musilk yang direkomendasikan
4	<i>Item Collaborative Filtering</i> untuk rekomendasi pembelian buku secara online (Uyun, et.al. 2011).	<p>Objek : Rekomendasi Buku</p> <p><i>Input</i> yang digunakan:</p> <ul style="list-style-type: none"> - <i>Data History</i> pembelian buku (dibeli atau tidak) - Penilaian buku 	Metode: <i>Item-Base Collaborative Filtering</i>	Menghasilkan Daftar buku yang direkomendasikan
5	<i>A Recommender System for On-line Course Enrolment: An Initial Study</i> (Michael & Barry, 2007)	<p>Objek : Rekomendasi Mata Kuliah</p> <p><i>Input</i> yang digunakan:</p> <ul style="list-style-type: none"> - <i>Data History</i> pilihan matakuliah - <i>Data Mahasiswa</i> 	<p>Metode: <i>Item-Base Collaborative Filtering</i></p> <p>Proses :</p> <ul style="list-style-type: none"> - Menentukan matriks <i>user</i> - Melakukan perhitungan kemiripan. - Melakukan perhitungan prediksi 	Menghasilkan Daftar matakuliah yang direkomendasikan

6	Penelitian yang diusulkan: Sistem Rekomendasi Pencarian Buku Pepustakaan Menggunakan Metode <i>Item-based Collaborative Filtering</i>	Objek : Rekomendasi Buku <i>Input</i> yang digunakan : <ul style="list-style-type: none">- Data Pinjaman- Data Buku- Data Mahasiswa- Pencarian katalog	Metode: <i>Item-Base Collaborative Filtering</i> Proses : <ul style="list-style-type: none">- Melakukan perhitungan kemiripan menggunakan <i>Cosine Similarity</i>.- Melakukan perhitungan nilai prediksi menggunakan <i>weighted sum</i> untuk memberikan rekomendasi	Menghasilkan Daftar Buku yang direkomendasikan
---	---	--	--	--

2.2 Sistem Rekomendasi

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

Pada sistem rekomendasi terdapat sekumpulan *item*. *Item* tersebut akan disaring (*filtering*) berdasarkan preferensi *item*, preferensi *user*, *rating*, dll. Sehingga menghasilkan beberapa *item* yang direkomendasikan kepada *user*. Saat ini sudah banyak yang menerapkan sistem rekomendasi pada *website* untuk mempermudah *user* mencari informasi seperti *website* penyedia video seperti youtube.com, toko *online* amazon.com dan lain lain. Gambar 2.1 merupakan halaman situs youtube.com. Pada halaman tersebut dapat dilihat pada sisi sebelah kanan terdapat beberapa video yang direkomendasikan sistem untuk *user*. Video-video tersebut direkomendasikan kepada *user* berdasarkan *history* pencarian yang pernah dilakukan *user* pada situs youtube.com. Sistem rekomendasi pada youtube dapat dilihat pada Gambar 2.1.

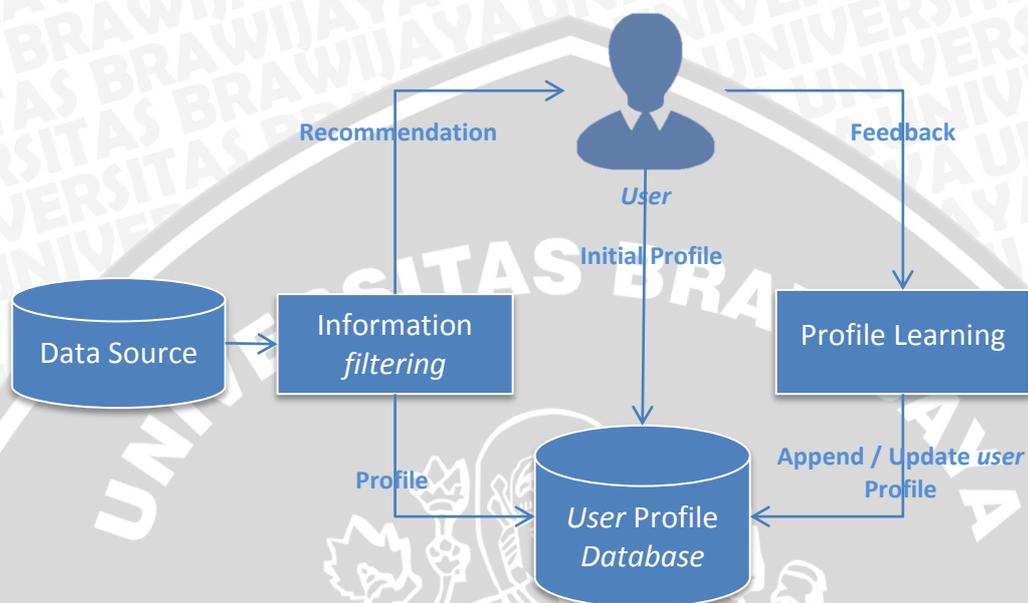


Gambar 2. 1 Sistem Rekomendasi pada Youtube.com

Sumber: youtube.com (2016)

2.2.1 Arsitektur Sistem Rekomendasi

Sistem rekomendasi bekerja dengan melakukan analisis terhadap sumber data yang ada. Data yang ada akan dihitung nilai kemiripan antar *item* maupun antar *user*/profil *user*.



Gambar 2. 2 Arsitektur Sistem Rekomendasi

Sumber: Bresee (1998)

Pada Gambar 2.2 terlihat bahwa sistem rekomendasi pada umumnya melakukan dua proses yaitu *information filtering* dan *profile learning*. *Information filtering* akan menghasilkan rekomendasi *item* yang mungkin sesuai untuk *user* untuk kemudian dinilai oleh *user* dengan memberikan *feedback*. *Feedback* ini yang digunakan oleh sistem untuk meng-*update* profil *user*. Dalam perkembangan sistem, *feedback* yang diberikan *user* kemudian disimpan dan diproses untuk menghasilkan rekomendasi yang lebih baik.

Menurut Bresee (1998), Algoritma sistem rekomendasi dapat dibagi menjadi dua kelompok umum, yaitu:

1. *Memory-based*

Algoritma *memory-based*, atau yang dikenal sebagai *heuristic-based* melakukan prediksi nilai *rating* dengan menggunakan seluruh data yang ada pada *database* nilai *rating*.

2. *Model-based*

Algoritma *model-based* melakukan prediksi *rating* dengan menggunakan model. Model yang dimaksud adalah sekelompok data yang biasanya diperoleh dari data latih dengan menggunakan teknik *data mining*. Model inilah yang disimpan oleh sistem untuk melakukan prediksi terhadap data yang *real*.

Secara umum terdapat 2 metode untuk membangun sebuah sistem rekomendasi yaitu, *Content based filtering* dan *Collaborative filtering*.

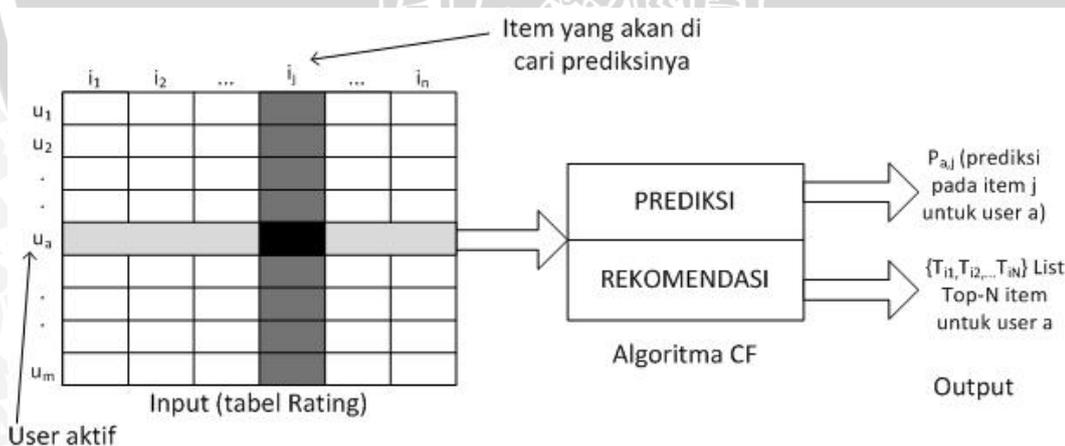
2.2.2 Content Based Filtering

Content based filtering memberikan rekomendasi suatu *item* untuk *user* berdasarkan deskripsi dari *item* yang akan direkomendasikan serta profil dari ketertarikan seorang *user* terhadap *item* (Pazzani, 2007). Sistem rekomendasi dengan metode *Content Based Filtering* biasa digunakan untuk merekomendasikan berita, artikel maupun situs *web*. Metode tersebut akan mengekstrak informasi yang terdapat pada *item* kemudian membandingkannya dengan informasi *item* yang pernah dilihat atau disukai oleh *user*.

2.2.3 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 yang didapat diantara *user* yang telah *me-rating* atau berinteraksi dalam sistem.

Sistem rekomendasi dengan metode *Collaborative Filtering* beroperasi di dalam sebuah ruang dua dimensi *User x Item*. *Rating* yang dapat diberikan oleh seorang *user* terhadap sebuah *item* dapat direpresentasikan sebagai R dan sistem rekomendasi mencoba untuk memprediksi *rating* yang akan diberikan seorang *user* untuk sebuah *item* yang belum pernah diberi *rating* sebelumnya. Misalnya, terdapat lima *user* $u_1, u_2, u_3, \dots, u_5$ dan lima *item* $i_1, i_2, i_3, \dots, i_5$. Sebuah sistem rekomendasi ingin memprediksi berapa *rating* yang akan diberikan oleh i_1 kepada i_5 . Berikut tabel matrix *user x item* pada sistem rekomendasi *Collaborative Filtering*.



Gambar 2. 3 Proses Collaborative Filtering

Sumber: Sarwar, at.al (2001)

Pada Gambar 2.3 merupakan skema dari metode *Collaborative Filtering*. Pada gambar tersebut direpresentasikan sejumlah m *user* x n *item* sebagai sebuah matriks *rating* di mana berisi nilai *rating* dari *user* untuk setiap *item*.

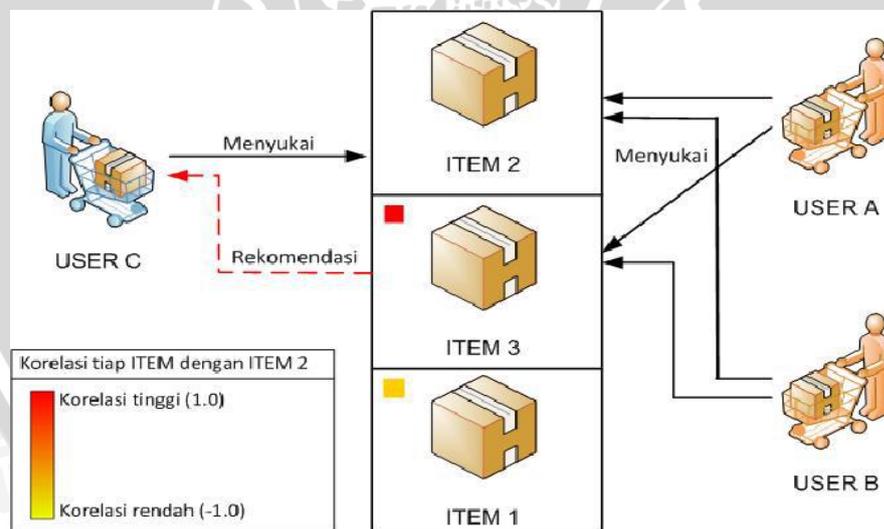
User aktif U_a pada skema tersebut adalah user yang akan diberikan rekomendasi *item* yang mungkin disukainya. *Item* tersebut direpresentasikan dalam dua bentuk yaitu (Ampazis, 2012):

1. Prediksi : merupakan nilai numerik, di mana $P_{a,j}$ adalah nilai prediksi *rating item* yang mungkin disukai oleh U_a .
2. Rekomendasi : merupakan daftar N *item* yang mungkin disukai oleh U_a . Dengan catatan bahwa *item* yang direkomendasikan belum pernah pinjam atau di *rating* oleh U_a . Hal ini sering juga disebut *top-N recommendation*.

2.2.3.2 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, et.al., 2011).

Pendekatan *item-based collaborative filtering* memberikan rekomendasi berdasarkan kemiripan antar *item*. Metode ini merupakan metode rekomendasi yang didasarkan 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. Motivasi kunci dibalik metode ini adalah user akan cenderung menyukai *item* yang sejenis atau mempunyai korelasi dengan *item* yang telah disukainya (Sarwar, at.al., 2001).



Gambar 2. 4 Skema *Item-based Collaborative Filtering*

Sumber: Hakim (2010)

Gambar 2.4 menggambarkan bagaimana *item* 3 direkomendasikan kepada user C yang telah menyukai *item* 2, di mana *item* 2 yang disukai user C memiliki korelasi yang kuat dengan *item* 3.

Item-based collaborative filtering melakukan analisis terhadap *matriks user-item* untuk mengetahui hubungan antar *item*, kemudian menggunakan

hubungan antar *item* tersebut untuk melakukan komputasi rekomendasi *user* (Sarwar, et.al., 2001).

	<i>Item₁</i>	<i>Item_i</i>	<i>Item_j</i>	<i>Item_n</i>
<i>User₁</i>		R	R	
<i>User₂</i>		R	-	
<i>User₃</i>		R	R	
<i>User_b</i>		-	R	
<i>User_m</i>		R	R	

Similarity antar *item* dihitung dengan melihat *item* oleh *user* pada contoh ini similaritas dari *item* *i* dan *j* didapat dengan melihat nilai *rating* *User₁*, *User₃*, *User_m*

Gambar 2. 5 Proses Komputasi *Similarity*

Sumber: Sarwar, et.al., (2001)

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 2.5.

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

1. Penemuan *similarity item*,
2. Penghitungan prediksi.

1. Penemuan *Similarity Item (Adjusted Cosine Similarity)*

Tahapan ini untuk mencari kemiripan antar *item*. *Item* yang pernah disukai *user* dibandingkan dengan *item* lain yang belum pernah disukai *user*. Hasilnya adalah nilai *similarity* (kemiripan) yang menggambarkan seberapa mirip suatu *item* dengan *item* lain. Berikut beberapa algoritma untuk menemukan *similarity item*, yaitu (Sarwar, et.al., 2001):

Persamaan *adjusted cosine similarity* digunakan untuk menghitung nilai kemiripan antar *item*. Perhitungan kemiripan ini merupakan penyempurnaan dari perhitungan kemiripan berbais 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 dan dengan waktu yang paling cepat (Sarwar, et.al., 2001).

Persamaan *adjusted cosine similarity* dapat dilihat pada Persamaan 2.1.

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

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* u

Untuk menghitung nilai kemiripan (*similarity*) antar *item*, diperlukan himpunan *user* yang me-*rating* *item* tersebut. Nilai yang dihasilkan pada persamaan *adjusted-cosine similarity* adalah berkisar antara +0.1 dengan -0.1. *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. Perhitungan 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 Algoritma *Wighted Sum* (Sarwar, at.al., 2001):

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.

Persamaan *Weighted Sum* dapat dilihat pada Persamaan 2.2.

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

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 selanjutnya yaitu membagi langkah pertama dengan langkah kedua, yang mana akan menghasilkan nilai prediksi untuk *user* terhadap *item*.

2.2.3.3 User-based Collaborative Filtering

Pendekatan *user-based collaborative filtering* ini merupakan salah satu pendekatan yang pertama kali digunakan dalam pembuatan sistem rekomendasi. Ide dasar dari pendekatan ini adalah dengan mengidentifikasi *user-user* yang memiliki similar *preferences* dari *user* aktif. Kemudian, diapatkan *items* yang dapat ditawarkan kepada *user* aktif. Masing-masing *item* tersebut akan dihitung prediksi *rating*-nya berdasarkan *rating* dari *user* sebelumnya.

Tabel 2. 2 Contoh Data Rating Dari User

User \ Item	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	1	5	2	3	2
User 2	2	3	4	2	2
User 3	4	4	3	4	3
User 4	2	4	4	4	1
User X	4	2	1	5	?

Sumber[Aditya, 2015]

Sebagai contoh, Tabel 2.1 menunjukkan matriks hubungan antara *user* dan *item* beserta nilai *rating* yang diberikan. Tugas dari sistem rekomendasi adalah memberikan prediksi nilai yang akan diberikan oleh *user* aktif X pada *item* yang belum pernah diberikan nilai *rating* olehnya. Untuk melakukannya, sistem melakukan pencarian *user-user* lain yang memiliki kesamaan *preferences*, dilihat dari nilai *rating* yang telah diberikan. Kemudian dari sekelompok *user-user* tersebut, sistem melakukan perhitungan untuk mengetahui apakah *user* aktif X akan menyukai *item* tersebut.

Salah satu cara yang banyak digunakan untuk menghitung kemiripan $sim(a,b)$ dari dua vektor *user* a, b adalah dengan menggunakan *Person's corellation coefisient*.

Persamaan *Person's corellation coefisient* dapat dilihat pada persamaan (2.3).

$$sim(a,b) = \frac{\sum_{i \in I} (R_{a,i} - \bar{R}_a)(R_{b,i} - \bar{R}_b)}{\sqrt{\sum_{i \in I} (R_{a,i} - \bar{R}_a)^2} \sqrt{\sum_{i \in I} (R_{b,i} - \bar{R}_b)^2}} \quad (2.3)$$

Keterangan :

$Sim(a,b)$ = similaritas *preferences* *user* a dan b

\bar{R}_a = rata-rata *rating* dari *user* a

$R_{a,i}$ = *rating* *user* a terhadap *item* i

Nilai kemiripan yang dihasilkan pada persamaan 2.3 berkisar antara -0 dan +0.1, di mana -0.1 berarti korelasi negatif dan +0.1 korelasi positif, mengindikasikan bahwa kedua *user* yang dibandingkan memiliki tingkat kesamaan/*preference* yang tinggi (Kwon, 2009).

Untuk memprediksi nilai *rating*, digunakan persamaan *wighted sum* (2.2). Persamaan tersebut menghasilkan nilai prediksi *rating* sesuai dengan *range* dan *rating* yang telah ada pada sistem.

2.3 Dewey Decimal Classification (DDC)

DDC adalah salah satu sistem klasifikasi berdasarkan subjek. Pada pengembangannya, simbol notasi pada sistem klasifikasi ini menggunakan sistem desimal angka Arab (Subroto, 2009). Terdapat tiga komponen yang penting dalam sistem klasifikasi ini, yaitu bagan (*schedules*), indeks relatif dan tabel-tabel. Bagan disini adalah penggunaan bilangan desimal untuk membagi semua bidang ilmu pengetahuan menjadi sepuluh kelas utama. Bidang ilmu tersebut tertulis pada Tabel 2.3.

Tabel 2. 3 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

Sumber : [Subroto, 2009]

Dari sepuluh kelas utama tersebut, nantinya masih akan terbagi-bagi lagi menjadi sub kelas untuk mengklasifikasikan suatu buku. Sebagai contoh, buku dengan tiga digit awal DDCnya 901 menandakan bahwa buku tersebut termasuk buku filsafat dan teori sejarah.

Pada penelitian ini, DDC akan digunakan untuk melakukan proses pengujian. Caranya, hasil rekomendasi yang keluar akan dibandingkan, apakah memiliki tiga digit DDC awal yang sama dengan buku yang dipilih sebagai *input*. Selain itu juga akan dilakukan perbandingan dengan satu digit awal buku yang merupakan kelas utama dari buku tersebut. Semakin banyak buku hasil rekomendasi yang satu kelas dengan buku yang dipilih, maka semakin tinggi pula akurasi yang dihasilkan.

2.4 Skala Penilaian

Banyak peneliti menggunakan skala penilaian untuk mengetahui pendapat responden terhadap penelitian yang dilakukan. Dengan cara meminta responden untuk memberikan penilaian terhadap sekumpulan *item*. Hal ini dilakukan untuk mengetahui seberapa besar kesukaan responden terhadap *item* tersebut.

Ada beberapa jenis skala penilaian yang dapat digunakan pada saat penelitian, yaitu: skala *Likert*, skala *Guttman* dan skala *Rating* (Babbie, 2012).

2.4.1 Skala *Likert*

Skala *Likert* adalah skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena pendidikan. Dalam skala *Likert* terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan negatif yang berfungsi untuk mengukur sikap negatif (Babbie, 2012). Skala yang digunakan pada skala *likert* yaitu antara 1 sampai 5 di mana tiap nilai diartikan sebagai berikut (Babbie, 2012):

1. Sangat tidak setuju, 2. Tidak setuju, 3. Netral, 5. Setuju, 6. Sangat setuju

2.4.2 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 sebagainya diberi skor 1; sedangkan untuk jawaban negative seperti salah, tidak, rendah, buruk, dan sebagainya diberi skor 0 (Babbie, 2012).

2.4.3 Skala *Rating (Rating Scale)*

Rating scale merupakan skala penilaian yang lebih fleksibel, skala penilaian ini tidak hanya untuk mengukur sikap tetapi dapat juga digunakan untuk mengukur persepsi responden terhadap fenomena lingkungan, seperti skala untuk mengukur status sosial, ekonomi, pengetahuan, kemampuan, dan lain-lain.

Pada skala penilaian *rating scale* menurut Wimmer & Joseph, (2012) keputusan peneliti yang menentukan skala mana yang digunakan untuk penilaian yang digunakan, apakah 0 sampai 5, 0 sampai 10, atau 0 sampai 100. Memilih jenis skala *rating* sebagian besar merupakan masalah preferensi pribadi.

2.5 Evaluasi Sistem Rekomendasi

Evaluasi digunakan untuk mengetahui seberapa bagus hasil rekomendasi yang diberikan sistem kepada *user*. Beberapa cara yang dapat dilakukan untuk mengevaluasi sistem rekomendasi yaitu dengan menguji tingkat akurasi sistem dengan menghitung tingkat *error* pada hasil rekomendasi dan juga menghitung nilai relevansi presisi dari hasil perbandingan dengan nomor DDC (*Dewey Decimal Clasification*).

2.5.1 Mean Absolute Error (MAE)

Dari beberapa jenis perhitungan *error*, MAE merupakan persamaan perhitungan nilai *error* yang paling sering digunakan untuk menghitung nilai *error* dari hasil rekomendasi. *Mean Absolute Error* (MAE) merupakan persamaan yang termasuk jenis *statistical accuracy metrics* di mana MAE akan menghitung nilai rata-rata selisih antara nilai prediksi dengan nilai yang sebenarnya (Sarwar, at.al., 2001).

Persamaan MAE dapat dilihat pada Persamaan 2.4.

$$MAE = \frac{\sum_{i=1}^n |p_i - q_i|}{N} \quad (2.4)$$

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*.

2.5.2 Presisi (Precision)

Sistem rekomendasi akan memberikan sekumpulan *item* sebagai hasil rekomendasi untuk *user*. Dari beberapa *item-item* tersebut tentu tidak semua *item* yang relevan atau yang sesuai dengan kebutuhan *user*. Untuk mengetahui kualitas hasil rekomendasi, dapat menggunakan rumus relevansi presisi yang membandingkan antara *item* yang relevan dengan total *item* yang dihasilkan atau yang direkomendasikan kepada *user* (Sarwar, at.al., 2001).

Persamaan Presisi dapat dilihat pada Persamaan 2.5.

$$Precision = \frac{\#relevant\ items\ retrieved}{\#retrieved\ items} = P(Relevant|Retrieved) \quad (2.5)$$

Relevant items retrieved adalah jumlah *item* relevan yang direkomendasikan sedangkan *retrieved items* adalah jumlah total *item* yang direkomendasikan.

2.5.3 Sparsity

Sparsity merupakan kondisi di mana terdapat banyak kekosongan *rating* pada *item* sehingga nilai *rating user* terhadap *item* menjadi jarang/renggang. Masalah *Sparsity* merupakan kendala yang ada pada sistem rekomendasi dengan metode *collaborative filtering* karena pada metode *collaborative filtering rating* merupakan hal yang paling penting dalam pemberian rekomendasi. Semakin banyak *rating* yang terkumpul maka semakin baik hasil rekomendasi yang diberikan. Semakin tinggi *Sparsity* data *rating* maka akan memberikan *poor accuracy* (akurasi yang rendah) terhadap hasil rekomendasi (Sanjung, 2011).

Jumlah *user* dan jumlah *item* yang besar akan sangat membantu proses perkomendasi, namun tidak mungkin semua *user* akan aktif menggunakan sistem dan memberikan penilaian pada setiap *item*. Terkadang *user* merasa

malas untuk memberi nilai *rating* pada *item* setelah memberi *rating* terhadap beberapa *item* saja. Pada umumnya *user* hanya melakukan *rating* terhadap *item-item* dalam jumlah yang lebih kecil dari jumlah *item* yang tersedia di *database*, sehingga menyebabkan matriks *user-item* mengalami kekosongan data yang sangat tinggi (Budianto, 2012).

Banyak cara yang dilakukan untuk mengatasi masalah *Sparsity* ini salah satunya yaitu dengan mengumpulkan data penilaian *rating user* secara implisit. *User* tidak perlu memberikan penilaian secara langsung dengan memilih *rating* penilaian yang disediakan. *User* hanya cukup melakukan aktivitas terhadap sistem seperti melakukan klik, *copy*, *drag*, *save* dan lain-lain di mana aktivitas-aktivitas ini dicatat sistem kemudian dikonversikan menjadi data *rating*. Selain itu ada juga teknik *smoothing*, teknik ini mengizinkan sistem untuk mengisi *rating-rating* yang masih kosong sehingga kekosongan *rating* tersebut dapat teratasi (Budianto, 2012).

2.6 Perpustakaan Universitas Brawijaya

Perpustakaan Universitas Brawijaya didirikan tanggal 5 januari 1963. Pada tahun 1965 perpustakaan pindah kekampus Universitas Brawijaya dengan menempati gedung seluas 400 m². Perpustakaan dilakukan perluasan hingga pada tahun 2008 luas perpustakaan mencapai 8.520 m²(Khoir, 2014).

Perpustakaan adalah salah satu tempat yang bisa kita gunakan untuk menemukan buku. Dengan semakin meningkatkan kesadaran akan pentingnya membaca, kini setiap daerah memiliki perpustakaan sendiri. Belum lagi adanya sekolah dan universitas yang memiliki perpustakaan untuk menunjang pembelajaran para siswa dan mahasiswanya. Terlebih lagi perpustakaan di universitas, yang memilki banyak sekali pengguna karena besarnya universitas tersebut. Tentu dengan banyaknya, tingkat transaksi yang terjadi terbilang tinggi, meskipun mungkin belum mencapai potensi maksimalnya(Khoir, 2014).

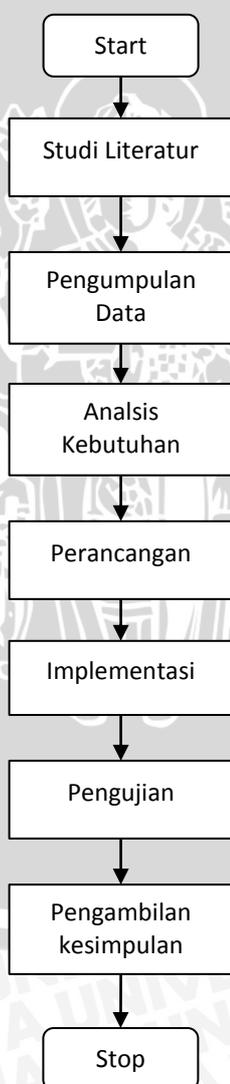
Perpustakaan Universitas Brawijaya menyediakan beberapa pelayanan untuk memudahkan mahasiswa dalam memanfaatkan perpustakaan diantaranya jasa pelayanan penelusuran informasi mengenai koleksi baik, majalah, buku, laporan skripsi, tesis dll. Informasi tersebut bisa dilakukan dengan memanfaatkan jasa muatan lokal Universitas Brawijaya yaitu Brawijaya *Knowledge Garden*(BKG). BKG ini menyediakan koleksi skripsi, tesis, disertasi, laporan penelitian, pidato pengukuhan, *prosiding* dan lainnya yang dihasilkan oleh civitas akademika Universitas Brawijaya. Perpustakaan UB juga memberikan jasa penelusuran informasi yang dinamakan OPAC(*Online Public Acces Catalogue*). OPAC adalah katalog komputer untuk mencari koleksi buku dan informasi BKG. Dengan informasi tersebut ketersediaan buku dapat tersampaikan. Jasa yang lain yang diberikan oleh Universitas Brawijaya adalah koleksi elektronik, koleksi *E-book*, *E-journal* seperti *Proquest*, *ScienceDirect*, EBSCO, IEEE, ASCE, ASME(Khoir, 2014).

BAB 3 METODOLOGI

Pada bab metodologi ini dijelaskan langkah-langkah yang akan dilakukan dalam pembuatan sistem rekomendasi pencarian buku perpustakaan menggunakan metode *Item-base Collaborative Filtering*. Tipe penelitian pada penelitian ini adalah tipe penelitian implementatif-perancangan, karena pada penelitian ini menitikberatkan pada aluran penelitian berupa produk baik *software* atau *hardware* sebagai solusi terhadap suatu permasalahan yang diangkat dalam penelitian.

3.1 Diagram Alir Penelitian

Diagram alir penelitian ini menjelaskan tahap-tahap yang dilakukan selama pembuatan penelitian dalam pembuatan aplikasi sistem rekomendasi.



Gambar 3. 1 Diagram Alir Penelitian

3.2 Studi Literatur

Studi literatur sebagai dasar teori yang digunakan selama penelitian, dan sebagai acuan dalam penulisan penelitian serta memberikan penjelasan terhadap penelitian dalam pengembangan sistem rekomendasi. Beberapa teori dan pustaka yang berkaitan dengan penulisan penelitian diantaranya:

1. Kajian Pustaka
2. Sistem Rekomendasi
3. *Collaborative Filtering*
 - a. *Item-based Collaborative Filtering*
 - b. *User-based Collaborative Filtering*
4. Evaluasi Sistem Rekomendasi
 - a. Penghitungan *error MAE*
 - b. Presisi (Precision)

3.3 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan di Perpustakaan Pusat Universitas Brawijaya Malang. Data yang dibutuhkan sesuai dengan surat permohonan data yang dilampirkan pada lampiran B.1. Pengumpulan data sebagai awal dari penelitian dalam mengembangkan aplikasi rekomendasi. Data yang diperlukan Data Mahasiswa, data buku, dan data peminjaman buku pada perpustakaan universitas Brawijaya. Caranya dengan melakukan observasi di perpustakaan Universitas Brawijaya.

Pengumpulan data untuk kegiatan penelitian ini dilakukan dengan cara observasi yaitu data yang didapatkan langsung dari objek penelitian. Metode pengumpulan data ini yang bersifat kuantitatif. Pada Tabel 3.1 dapat dilihat kebutuhan data pada penelitian ini.

Tabel 3. 1 Penentuan Kebutuhan Data Penelitian

No.	Kebutuhan Data	Sumber Data	Metode	Kegunaan Data
1.	Data Buku	Pepust Universitas Brawijaya	Observasi	Sebagai data buku dan rekomendasi
2.	Data Mahasiswa	Pepust Universitas Brawijaya	Observasi	Sebagai data <i>user</i> untuk melakukan penilaian
3.	Data transaksi peminjaman buku	Perpust Universitas Brawijaya	Observasi	Data yang didapat akan digunakan sebagai data latihan

3.4 Analisis Kebutuhan

Analisis kebutuhan ini meliputi analisis permasalahan, analisis kebutuhan pada perangkat lunak, dan perangkat keras yang akan digunakan pada sistem yang akan dirancang. Analisis kebutuhan diperlukan untuk menyesuaikan

kebutuhan sistem, kebutuhan fungsional. Berikut analisis kebutuhan dalam penelitian ini.

1. Kebutuhan perangkat keras, meliputi:
 - Laptop dengan Processor Intel® Core™ i3-2350M CPU @ 2.30GHz
2. Kebutuhan perangkat lunak, meliputi:
 - Sistem Operasi Windows 8
 - MySql DBMS
 - Visual Studio 2010 dengan bahasa pemrograman C#
3. Kebutuhan data, meliputi:
 - Data transaksi peminjaman buku oleh mahasiswa
 - Data buku sebagai *item* yang direkomendasi kepada mahasiswa
 - Data mahasiswa
4. Kebutuhan Fungsional, Meliputi:
 - Sistem dapat menerima *input* dari buku yang dipilih dari daftar pencarian yang dilakukan oleh *user*
 - Sistem membutuhkan NIM dari *user* untuk mengetahui jurusan dari *user* tersebut
 - Sistem membutuhkan data *user*, data transaksi yang pernah dilakukan dan data buku
 - Sistem dapat melakukan perhitungan dengan menggunakan metode *Item-Based Collaborative Filtering*
 - Sistem dapat mengeluarkan *output* berupa beberapa rekomendasi buku sesuai dengan buku yang dicari oleh pengunjung perpustakaan
5. Kebutuhan Non Fungsional, Meliputi:
 - Tampilan *interface* yang digunakan dikenal oleh *user*. Artinya *user* tidak merasa asing dengan penggunaan fitur-fitur tampilan seperti *button*, *label*, *textbox*.
 - Monitor, Mouse, Keyboard.

3.5 Perancangan

Perancangan merupakan tahap lanjutan dari analisis kebutuhan sistem, di mana pada tahap ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean pada bahasa pemrograman. Perancangan sistem dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Proses perancangan perangkat lunak ini diantaranya: Perancangan diagram blok sistem digunakan untuk menerangkan tahapan proses sistem, diagram alir perancangan sistem, perancangan perhitungan, perancangan *database* diantaranya perancangan tabel data, pemodelan *Entity Relationship Diagram* (ERD) yang digunakan untuk menjelaskan hubungan antar data dalam *database*, perancangan antarmuka bertujuan untuk menampilkan tampilan antarmuka dari sistem yang akan dibangun. Tahapan perancangan bertujuan untuk mempermudah implementasi dan pengujian dalam sistem rekomendasi. Langkah kerja dalam sistem akan disesuaikan dengan diagram blok sistem rekomendasi.

3.6 Implementasi

Implementasi aplikasi sistem rekomendasi yang menerapkan metode *Item-base Collaborative Filtering*. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman berorientasi object yaitu dengan menggunakan bahasa pemrograman C# dengan manajemen basis data menggunakan *MYSQL*. Implementasi aplikasi yaitu meliputi :

- Spesifikasi sistem
- Implementasi basis data dengan menggunakan DBMS *MySql* yang bertujuan untuk memudahkan melakukan memanipulasi dan menyimpan data.
- Implementasi *Algoritma Collaborative Filtering* ke dalam bahasa pemrograman C#, dengan melakukan perhitungan untuk menentukan rekomendasi yang sesuai, dengan menggunakan metode *Item-Based Collaborative Filtering*.
- Implementasi *interface* sistem.
- Output* yang diperoleh berupa hasil rekomendasi dari sistem dengan memberikan saran atau rekomendasi buku yang sesuai berdasarkan ketertarikan mahasiswa menggunakan metode *Item-base Collaborative Filtering*.

3.7 Pengujian

Pengujian ini merupakan pengujian terhadap sistem yang sudah diimplementasikan sebelumnya, pengujian ini menggunakan beberapa teknik-teknik pengujian, dengan tujuan untuk menilai seberapa baik kualitas rekomendasi yang dihasilkan. Pengujian ini dilakukan dengan menggunakan beberapa teknik pengujian yang pertama fungsi perhitungan *error* (MAE) pengujian ini bertujuan menguji tingkat akurasi rekomendasi, pengujian Relevansi presisi dengan DDC(*Dewey Decimal Clasification*), dan pengujian waktu eksekusi proses rekomendasi.

- Fungsi perhitungan *error* yang digunakan adalah *Mean Absolute Error* (MAE). Perhitungan MAE dengan mencari selisih antara *rating* asli dengan *rating* prediksi kemudian membaginya dengan jumlah *item* yang dihitung. Berdasarkan sumber-sumber yang ada, rata-rata untuk pengujian *sitem* rekomendasi dengan metode *Item-based collaborative filtering* menggunakan MAE.
- Pengujian Relevansi Presisi dengan DDC(*Dewey Decimal Clasification*), sistem diuji tingkat akurasi dengan membandingkan nomer DDC buku yang dicari/atau dipilih dengan buku yang direkomendasikan.
- Pengujian waktu eksekusi sistem rekomendasi, pengujian ini bertujuan untuk melihat seberapa besar waktu yang dibutuhkan untuk menghasilkan rekomendasi buku. Sistem akan diuji dengan memasukkan data -buku dalam jumlah tertentu kemudian dihitung waktu yang diperlukan untuk memperoleh hasil rekomendasi.

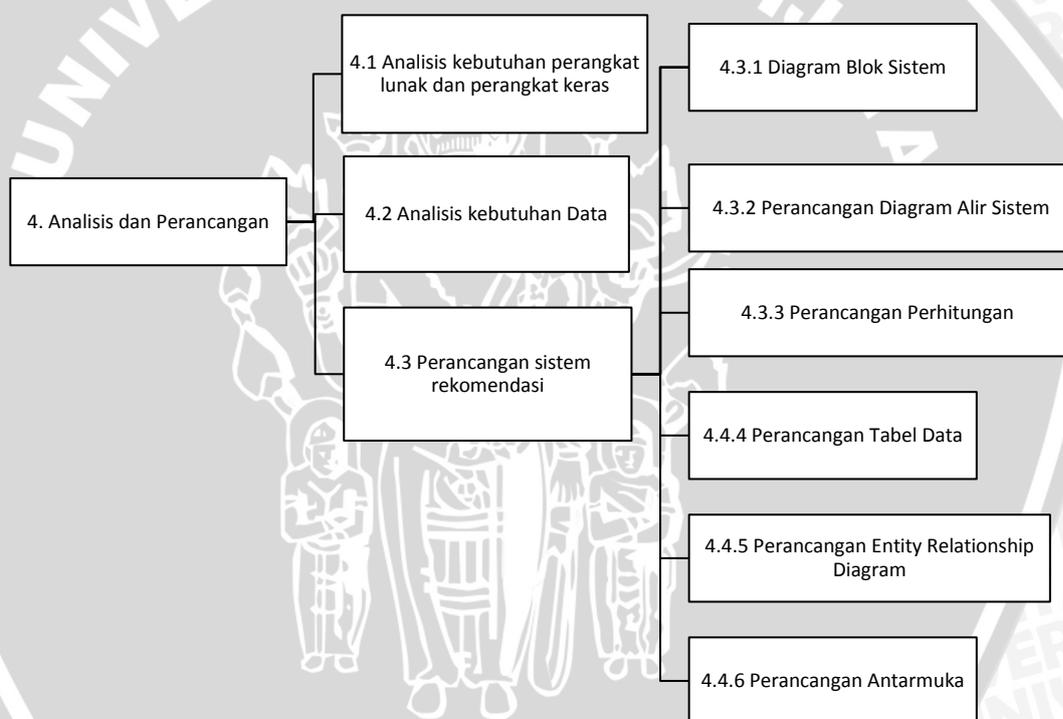
3.8 Pengambilan Kesimpulan

Kesimpulan dilakukan setelah semua tahapan dilakukan. Tahap perancangan, implementasi, dan pengujian yang diterapkan telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.



BAB 4 ANALISIS DAN PERANCANGAN

Pada bab ini membahas mengenai analisis dan perancangan aplikasi sistem rekomendasi pencarian buku di perpustakaan menggunakan metode *Item-based Collaborative Filtering*. Pada subbab Analisis dan perancangan membahas mengenai analisis kebutuhan perangkat lunak dan perangkat keras, dan analisis kebutuhan data. Kemudian untuk subbab perancangan ada perancangan sistem rekomendasi terdiri dari 6 bagian, yaitu perancangan diagram alir sistem, perancangan perhitungan, perancangan tabel data, Perancangan *Entity Relationship Diagram* dan perancangan antarmuka. Pohon perancangan sistem rekomendasi dapat dilihat pada Gambar 4.1 berikut ini.



Gambar 4. 1 Pohon Analisis dan Perancangan

4.1 Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras

Analisis kebutuhan ini bertujuan untuk menganalisis kebutuhan-kebutuhan perangkat lunak dan perangkat keras yang harus disediakan oleh sistem rekomendasi yang akan dibangun agar dapat memenuhi kebutuhan *user*.

Agar dapat bekerja dengan baik, sistem membutuhkan dukungan perangkat keras dan perangkat lunak untuk menjalankan sistem tersebut. Berikut analisis perangkat keras dan perangkat lunak yang dibutuhkan untuk membangun dan mengimplementasikan sistem rekomendasi buku.

1. Perangkat lunak
 - Sistem Operasi : Windows 8.1
 - Basis Data : *MySQL Database Management System* (DBMS)
 - Tools : Visual Studio 2010 dengan Bahasa Pemrograman C#
2. Kebutuhan Perangkat keras meliputi :
 - Komputer Laptop : *Processor core i3 2.40 GHz*
 - Memori : 4 GB

4.2 Analisis Kebutuhan Data

Data buku dan data mahasiswa yang digunakan pada penelitian ini merupakan data yang terdapat di perpustakaan Universitas Brawijaya namun untuk data peminjaman yang digunakan pada penelitian ini adalah data *dummy*. Dikarenakan data peminjaman buku oleh mahasiswa, dari data yang diberikan oleh pihak perpustakaan Universitas Brawijaya memiliki banyak ketidakcocokan. Banyak buku yang ada di data peminjaman tidak ada di data buku. Begitupun data mahasiswa. Karena itu, untuk data peminjaman digunakan data *dummy* dengan berbagai kondisi. Data yang digunakan adalah data buku, mahasiswa (angkatan 2013-2014), dan histori peminjaman. Data-data buku, mahasiswa dan peminjaman diperoleh dengan meng-*export* data excel ke *database* sistem.

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 4.1.

Tabel 4. 1 Analisis Kebutuhan Data

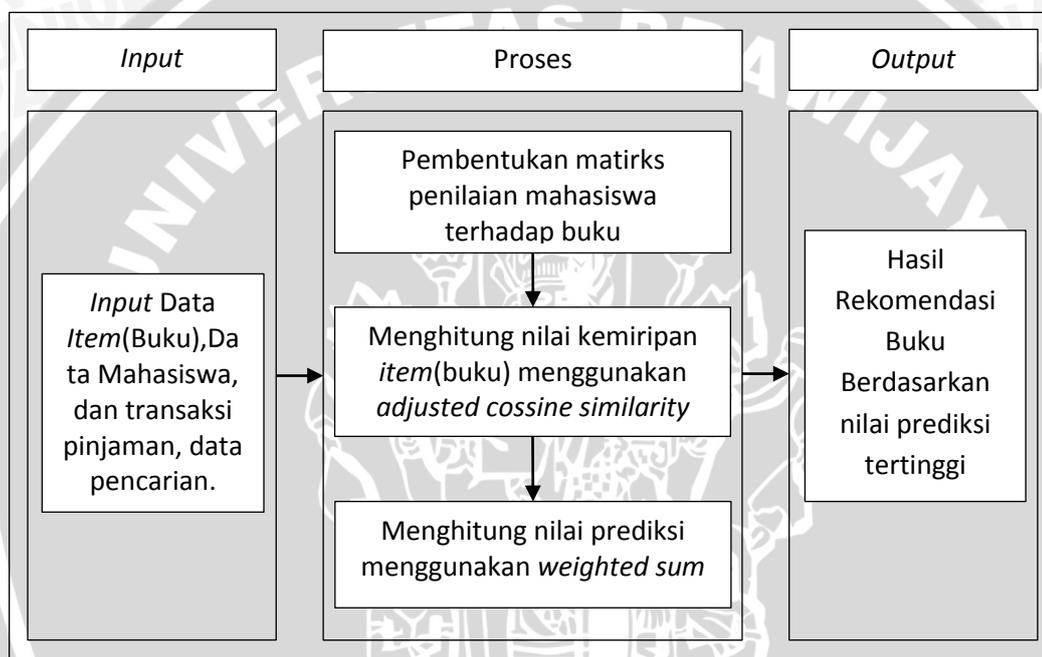
Data	Keterangan
Mahasiswa	Mahasiswa yang menggunakan sistem untuk memperoleh rekomendasi buku. Data ini mencakup nomor mahasiswa/NIM (Nomor Induk Mahasiswa, nama, jurusan, dan program studi).
Buku	Buku berisi data buku-buku yang akan direkomendasikan dan dipinjam oleh mahasiswa. Data ini mencakup id 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.

4.3 Perancangan Sistem Rekomendasi

Pada tahap ini akan dijelaskan perancangan sistem rekomendasi yang akan dibangun. Perancangan ini terdiri dari beberapa bagian, diantaranya perancangan diagram blok sistem, diagram alir sistem, dan perhitungan manual.

4.3.1 Diagram Blok Sistem Rekomendasi Pencarian Buku Perpustakaan

Diagram Blok sistem Rekomendasi adalah diagram yang berbentuk blok-blok yang menggambarkan aliran proses dari komponen-komponen sistem yang memuat fungsi matematis. Diagram blok sistem menjelaskan cara kerja sistem yang dimulai dari masukan sampai keluaran yang dihasilkan. Diagram blok sistem yang akan digunakan dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Diagram Blok Sistem Rekomendasi Pencarian Buku Perpustakaan

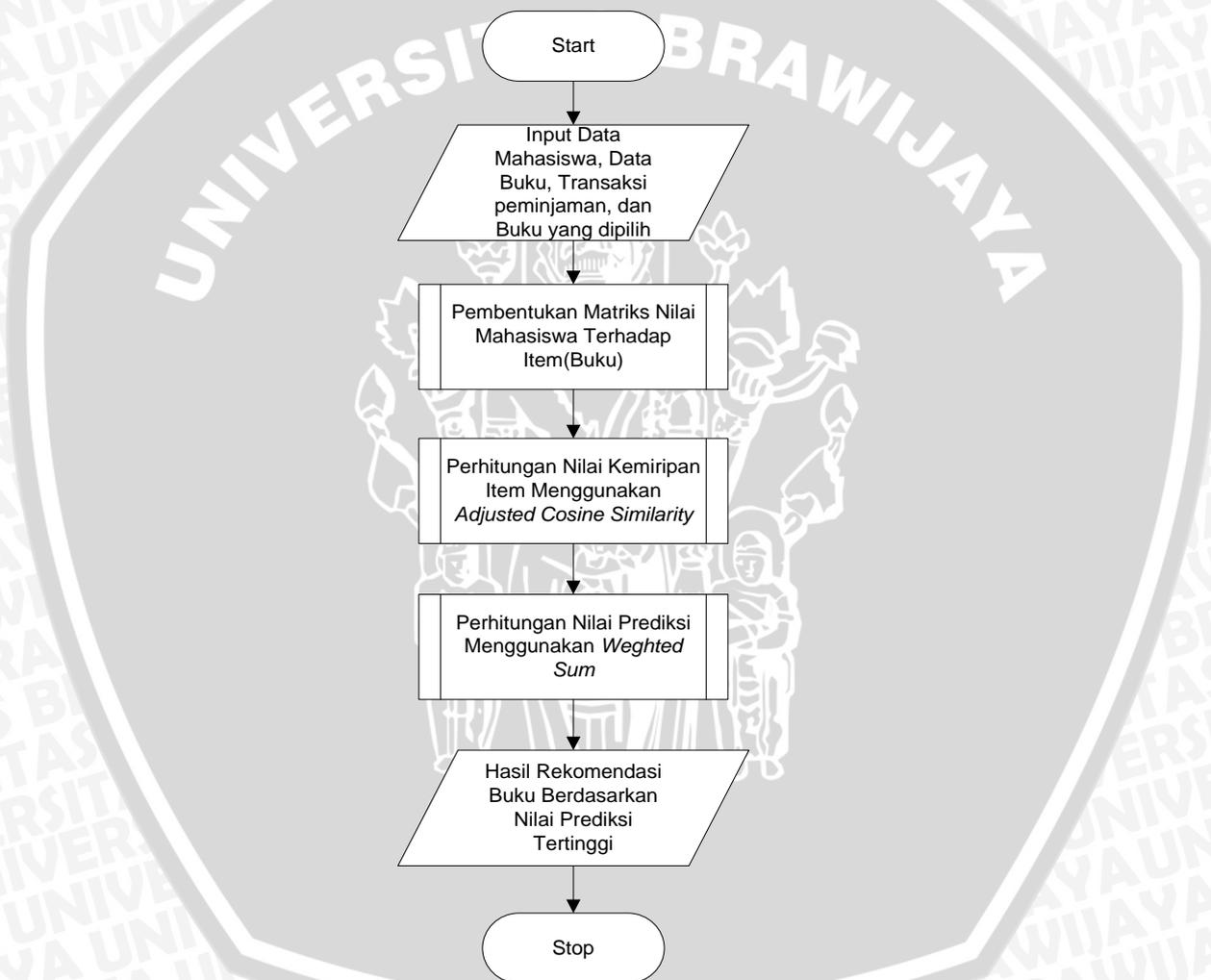
Pada Diagram di atas terdiri dari tiga proses utama, yaitu:

- **Input**
 Input pada sistem rekomendasi ini adalah Data *Item*(Buku), Mahasiswa, data Pinjaman Buku, dan data pencarian atau data buku pilihan mahasiswa.
- **Proses**
 Proses perhitungan pada penelitian ini menggunakan metode *Item-Based Collaborative Filtering* untuk menentukan rekomendasi buku. Tahap-tahap perhitungan menggunakan metode *Item-base Collaborative Filtering*, antara lain sebagai berikut.
 1. Pembentukan matriks penilaian mahasiswa terhadap buku.
 2. Menghitung nilai rata-rata penilaian mahasiswa terhadap buku.
 3. Menghitung nilai kemiripan buku menggunakan *adjusted cosine similarity*.

4. Menghitung nilai prediksi menggunakan *weighted sum*, nilai tertinggi akan dijadikan hasil rekomendasi.
- *Output*
Keluaran dari sistem rekomendasi ini adalah rekomendasi berupa buku yang sesuai berdasarkan perhitungan dengan metode *item-based collaborative filtering* dengan nilai terbesar dari nilai Prediksi.

4.3.2 Alir Perancangan Sistem

Sistem rekomendasi buku menggunakan metode *Item-Based Collaborative Filtering* ini sendiri memiliki alir perancangan sistem seperti yang akan ditunjukkan pada Gambar 4.2



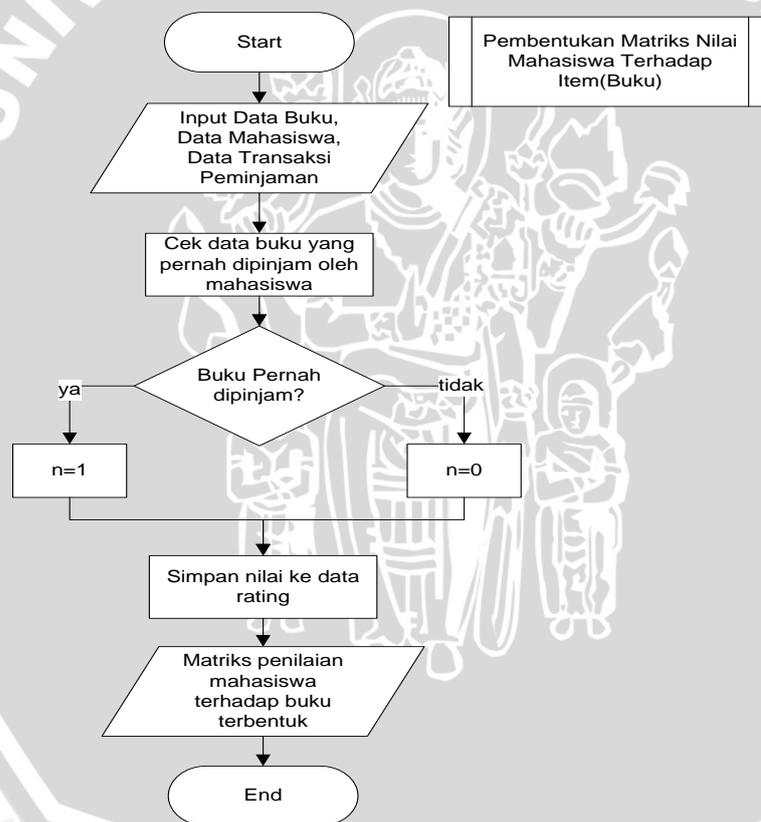
Gambar 4. 2 Diagram Alir Sistem Rekomendasi Buku Perpustakaan Dengan Metode *Item-Based Collaborative Filtering*

Tahapan-tahapan dalam sistem rekomendasi buku dengan menggunakan *item-base collaborative filtering* dimulai dengan menginputkan buku, data mahasiswa, data peminjaman, dan data buku yang dipilih. 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 *item*. 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 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 persamaan (2.2) sehingga menghasilkan nilai prediksi yang paling tinggi. Sistem selesai ketika sistem mengeluarkan *output* berupa hasil rekomendasi yang akan terletak di bawah detail buku yang dicari.

4.3.2.1 Alir Perancangan Pembentukan Matriks

Proses yang dilakukan pertama adalah pembentukan matriks dari *input* tersebut. Diagram alir dari pembentukan matriks tersebut akan ditunjukkan pada gambar 4.3.

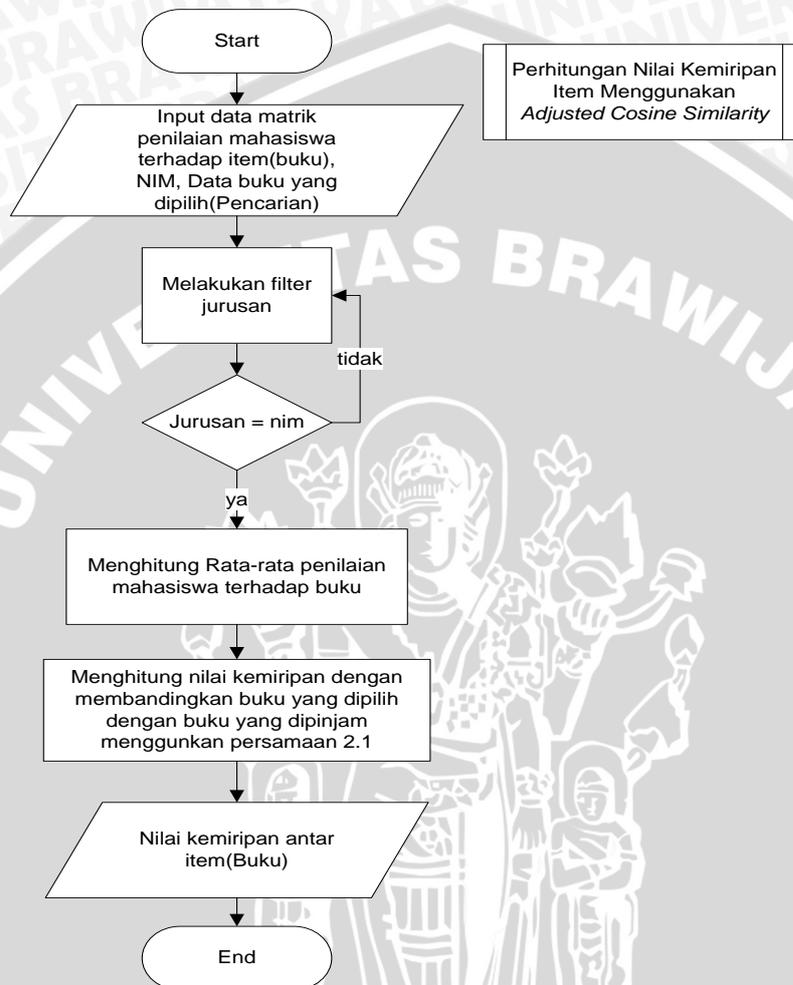


Gambar 4. 3 Diagram Alir Perancangan Pembentukan Matrik

Penyusunan matriks membutuhkan dua atribut, yakni mahasiswa dan buku. Nantinya pada kolom-kolom matriks tersebut akan terisi oleh angka 1 atau 0. Angka 1 menunjukkan bahwa buku tersebut pernah dipinjam oleh mahasiswa, dan angka 0 menunjukkan buku tidak pernah dipinjam oleh mahasiswa tersebut. Pengisian data dengan cara mengkonversi data peminjaman sebagai data penilaian mahasiswa terhadap buku berupa matriks penilaian mahasiswa terhadap buku.

4.3.2.2 Alir Perancangan Perhitungan Nilai Kemiripan(*Similarity*)

Setelah matriks terbentuk, langkah yang dilakukan selanjutnya adalah menghitung nilai kemiripan antar buku menggunakan algoritma *Adjusted Cosine Similarity*. Diagram alir dari proses perhitungan nilai kemiripan ditunjukkan pada gambar 4.4 berikut.

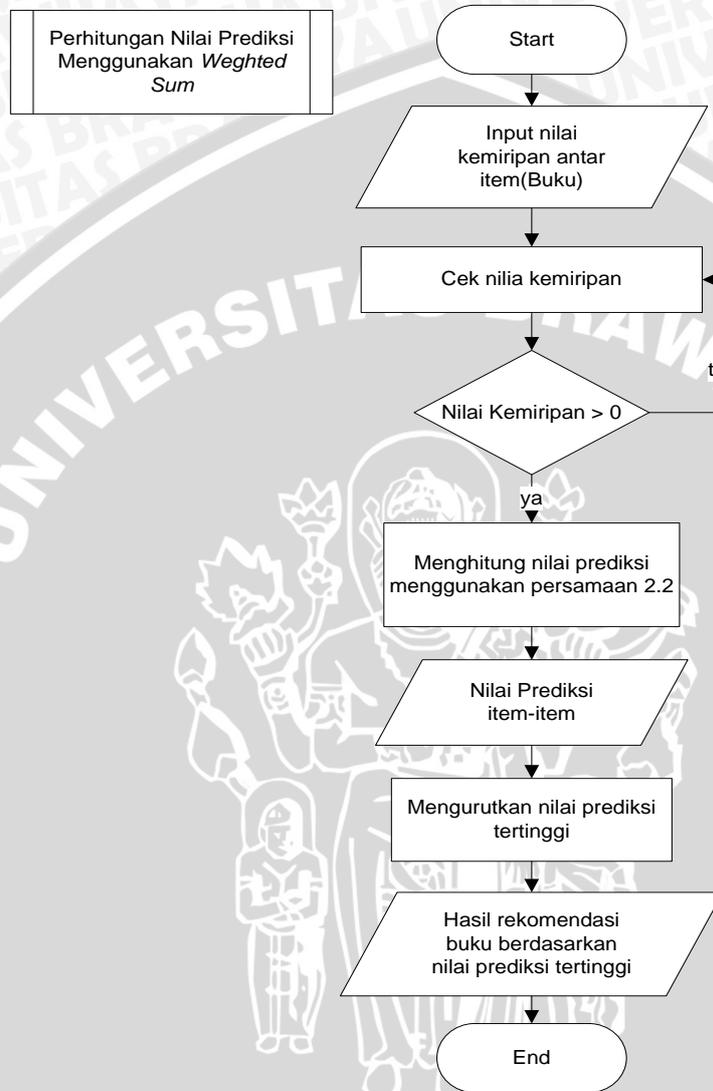


Gambar 4. 4 Diagram alir Perhitungan Nilai Kemiripan

Terdapat 3 *input* yang dibutuhkan dalam tahap perhitungan kemiripan, yakni matrik penilaian mahasiswa terhadap *item*, Nim, dan buku yang dipilih. Proses awal yang dilakukan adalah melakukan *filtering* jurusan terhadap mahasiswa, sehingga perhitungan hanya akan dilakukan terhadap mahasiswa yang satu jurusan dengan *user*. Setelah itu barulah dihitung nilai kemiripannya dengan menggunakan persamaan (2.1) untuk menghitung kemiripan karena menurut dari penelitian yang dilakukan (Sarwar,2001), algoritma yang dapat menghasilkan nilai error (MAE) terkecil adalah algoritma *adjusted cosine similarity*.

4.3.2.3 Alir Perancangan Perhitungan Nilai Prediksi

Setelah menghasilkan nilai kemiripan antar buku, langkah selanjutnya adalah menghitung nilai prediksi menggunakan persamaan 2.2. Diagram alir dari perhitungan nilai prediksi akan ditunjukkan pada gambar 4.6 sebagai berikut.



Gambar 4. 5 Diagram Alir Perhitungan Nilai Prediksi

Perhitungan nilai prediksi menghasilkan *output* nilai prediksi yang nantinya akan direkomendasikan kepada mahasiswa dengan memberikan rekomendasi dengan nilai prediksi *item*(buku) tertinggi.

4.3.3 Perancangan Perhitungan

Perancangan perhitungan akan menjelaskan mengenai manualisasi sistem rekomendasi buku menggunakan metode *Item-Based Collaborative Filtering*. Disini akan dijelaskan secara bertahap bagaimana metode tersebut akan membuat rekomendasi. Tahapan-tahapan metode *Item-Based Collaborative Filtering* sendiri adalah pembentukan matriks, perhitungan nilai *similarity* dan perhitungan nilai prediksi.

4.3.3.1 Pembentukan Matriks Penilaian Mahasiswa Terhadap Buku

Matriks disini digunakan untuk mengetahui apakah suatu buku pernah dipinjam oleh mahasiswa atau tidak. Buku yang pernah dipinjam bernilai 1 dan buku yang tidak pernah dipinjam bernilai 0. Angka 1 dan 0 ini berfungsi sebagai *rating* atau penilaian dari buku tersebut. skala penilaian ini dinamakan skala penilaian *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 (Babbie, 2012).

Sebagai contoh terdapat tabel matriks antara mahasiswa dengan buku yang pernah dipinjam atau tidak. Tabel tersebut bersis 6 buah buku/*item* A,B,C,D,E,F dan 6 buah *user*/mahasiswa 1,2,3,4,5,6. Buku A merupakan perwakilan buku Sistem Operasi, Buku B adalah perwakilan buku ‘Algoritma Pemrograman C++’, Buku C merupakan perwakilan dari buku ‘Just XML’, Buku D merupakan perwakilan buku ‘Membuat Sendiri Aplikasi Android Untuk Pemula’, Buku E merupakan perwakilan buku ‘Bahasa Pemrograman C#’ dan Buku F merupakan perwakilan buku ‘Belajar PHP’. Dengan membuat matriks penilaian mahasiswa terhadap buku sehingga dapat dilakukan perhitungan kemiripannya. Pembentukan matriks penilaian terhadap buku dapat dilihat pada tabel 4.2.

Tabel 4. 2 Matriks Hubungan Mahasiswa Terhadap Buku

	Buku A	Buku B	Buku C	Buku D	Buku E	Buku F
Mahasiswa 1	1	0	1	0	1	0
Mahasiswa 2	1	1	1	1	0	1
Mahasiswa 3	1	1	1	1	1	0
Mahasiswa 4	1	1	1	0	1	0
Mahasiswa 5	1	1	1	1	1	0
Mahasiswa 6	1	1	1	1	1	0
Mahasiswa X	1	?	1	1	1	?

4.3.3.2 Perhitungan Nilai Kemiripan

Setelah pembentukan matriks mahasiswa terhadap buku Kemudian dilakukan proses menghitung nilai kemiripan menggunakan persamaan *adjusted cosine similarity* (2.1). sebelum melakukan perbandingan antar *item*/buku untuk mencari nilai kemiripan langkah pertama mencari rata-rata penilaian mahasiswa terhadap buku. Jika semua telah dihitung, maka hasilnya akan tampak pada tabel 4.3.

Tabel 4. 3 Tabel Rata-rata penilaian Mahasiswa terhadap Buku

	Buku A	Buku B	Buku C	Buku D	Buku E	Buku F	Rata-Rata Penilaian
Mahasiswa 1	1	0	1	0	1	0	0,50
Mahasiswa 2	1	1	1	1	0	1	0,83
Mahasiswa 3	1	1	1	1	1	0	0,83
Mahasiswa 4	1	1	1	0	1	0	0,67
Mahasiswa 5	1	1	1	1	1	0	0,83
Mahasiswa 6	1	1	1	1	1	0	0,83
Mahasiswa X	1	?	1	1	1	?	0,67

Dalam menghitung nilai kemiripan, nilai yang akan dihasilkan oleh persamaan *adjusted-cosine similarity* adalah berkisar antara +1.0 dengan -1.0, *Item* dianggap saling berkolerasi jika nilai *similarity* antara kedua *item* tersebut mendekati +1, begitu juga sebaliknya *item* dianggap tidak berkolerasi apabila nilai *similarity*-nya mendekati -1

Sebagai contoh akan dihitung nilai kemiripan antara Buku B (Algoritma Pemrograman C++) dan Buku C (Just XML) dapat dilihat pada perhitungan dibawah ini perhitungan kemiripan menggunakan persamaan 2.1.

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

$$sim(B, C) = \frac{(0-0,50)*(1-0,50)+(1-0,83)*(1-0,83)+(1-0,83)*(1-0,83)+(1-0,67)*(1-0,67)+(1-0,83)*(1-0,83)+(1-0,83)*(1-0,83)}{\sqrt{(0-0,50)^2+(1-0,83)^2+(1-0,83)^2+(1-0,67)^2+(1-0,83)^2+(1-0,83)^2} \sqrt{(1-0,50)^2+(1-0,83)^2+(1-0,83)^2+(1-0,67)^2+(1-0,83)^2+(1-0,83)^2}}$$

$$sim(B, C) = \frac{-0,027}{\sqrt{0,687184271} \sqrt{0,687184271}}$$

$$sim(B, C) = -0,058823529$$

Setelah menghitung semua kemiripan nilai yang lainnya menggunakan rumus yang sama seperti diatas, maka diperoleh tabel kemiripan nilai antar buku sebagai berikut:

Tabel 4. 4 Tabel Hasil Perhitungan Kemiripan Antar Buku

BUKU 1	BUKU 2	Nilai Kemiripan
Buku B	Buku A	-0,058823529
Buku B	Buku C	-0,058823529
Buku B	Buku D	0,225188675
Buku B	Buku E	-0,265143907
Buku F	Buku A	-0,748129097



Buku F	Buku C	-0,748129097
Buku F	Buku D	0,203251144
Buku F	Buku E	-0,574975194

Setelah memperoleh nilai kesamaan antar buku, nilai yang lebih besar dari 0 akan digunakan untuk mendapatkan nilai prediksi, karena nilai tersebut dianggap sebagai batas bawah keterhubungan antar *item*. Untuk memperoleh nilai prediksi digunakan persamaan *weighted sum* (2.2).

4.3.3.3 Perhitungan Nilai Prediksi

Tahap selanjutnya setelah didapatkan nilai kemiripan (*Similarity*) adalah menghitung prediksi menggunakan persamaan *Weighted Sum* (2.2). Pertama sekali yang dilakukan dalam menghitung prediksi adalah membaca nilai *similarity* yang lebih dari 0, di mana nilai yang lebih besar dari 0 dianggap berkorelasi. Setelah itu menghitung prediksi untuk masing-masing buku/*item* dengan persamaan (2.2).

Berikut contoh perhitungan prediksi dari tabel *similarity* diatas yang sebelumnya sudah dilakukan perhitungan kemiripan. Sebagai contoh untuk memprediksi kecenderungan Buku B untuk Mahasiswa X dengan melakukan perhitungan menggunakan persamaan *weighted sum* (2.2) maka dilakukan perhitungan sebagai berikut:

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

$$P(\text{Mahasiswa X, Buku B}) = \frac{(1 * 0,225)}{|1|}$$

$$P(\text{Mahasiswa X, Buku B}) = 0,225188675$$

Setelah menghitung nilai prediksi menggunakan persamaan (2.2) maka diperoleh hasil perhitungan prediksi yang ditunjukkan pada tabel 4.5

Tabel 4. 5 Tabel Hasil Perhitungan Nilai Prediksi

MAHASISWA	BUKU	Nilai Prediksi
Mahasiswa X	B	0,225188675
Mahasiswa X	E	0,203251144

Dari Tabel hasil perhitungan nilai prediksi pada tabel 4.5 diatas dapat kita lihat bahwa prediksi Mahasiswa X mendapatkan rekomendasi Buku B, karena Buku B lebih tinggi dari pada nilai prediksi Buku E. Dapat dilihat tabel top N prediksi pada tabel 4.6.



Tabel 4. 6 Tabel Nilai Top N Prediksi

MAHASISWA	Nilai top N prediksi	BUKU Yang direkomendasikan	
Mahasiswa X	0,225188675	B	Algoritma Premrograman C++
Mahasiswa X	0,203251144	E	BELAJAR PHP

4.3.4 Perancangan Tabel Data

Pada perancangan tabel data ini menjelaskan pengelolaan data dengan menampilkan rancangan tabel data yang menampung data pada sistem rekomendasi buku.

4.3.4.1 Tabel Mahasiswa

Tabel mahasiswa digunakan untuk menyimpan data mahasiswa sebagai anggota perpustakaan yang dapat meminjam buku dan mendapatkan rekomendasi buku.

Tabel 4. 7 Tabel Mahasiswa

Nama Field	Tipe dan lebar	Keterangan
id_mahasiswa	INT(11)	Primary key
nama_mahasiswa	Varchar(20)	-
Jurusan	Varchar(20)	-
Fakultas	Varchar(20)	-

4.3.4.2 Tabel Buku

Tabel Buku berfungsi untuk menyimpan data-data buku yang ada di perpustakaan Universitas Brawijaya dan juga sebagai rekomendasi.

Tabel 4. 8 Tabel Buku

Nama Field	Tipe dan lebar	Keterangan
id_buku	Int(20)	Primary key
Judul	Varchar(250)	-
Printing	Varchar(10)	-
Edisi	Varchar(10)	-
Volume	Varchar(10)	-
dewey_no	Varchar(10)	-
kota_pub	Varchar(20)	-
Publikasi	Varchar(20)	-

tahun_pub	Date	-
Copies	Varchar(10)	-
Creator	Varchar(20)	-
Subject	Varchar(20)	-
ISBN	Varchar(10)	-
phys_description	Varchar(50)	-
Index	Varchar(20)	-
Bibliography	Varchar(20)	-
Language	Varchar(20)	-
Cover	Binary	-

4.3.4.3 Tabel Peminjaman

Tabel pinjaman berguna untuk menyimpan data setiap transaksi peminjaman yang data ini nantinya dikonversi menjadi data penilaian (*rating*) terhadap suatu *item*(buku).

Tabel 4. 9 Tabel Peminjaman

Nama Field	Tipe dan lebar	Keterangan
id_peminjaman	INT(11)	Primary key
id_mahasiswa	INT(11)	-
tgl_pinjam	Date and time	-
tgl_kembali	Date and time	-

4.3.4.4 Tabel Detail Peminjaman

Tabel Detail pinjaman berguna untuk menyimpan data detail setiap transaksi peminjaman buku yang data ini berfungsi sebagai data penilaian mahasiswa terhadap suatu *item*.

Tabel 4. 10 Tabel Detail Peminjaman

Nama Field	Tipe dan lebar	Keterangan
id_detail	INT(11)	Primary key
id_peminjaman	INT(11)	-
id_buku	INT(11)	-

4.3.4.5 Tabel Rating

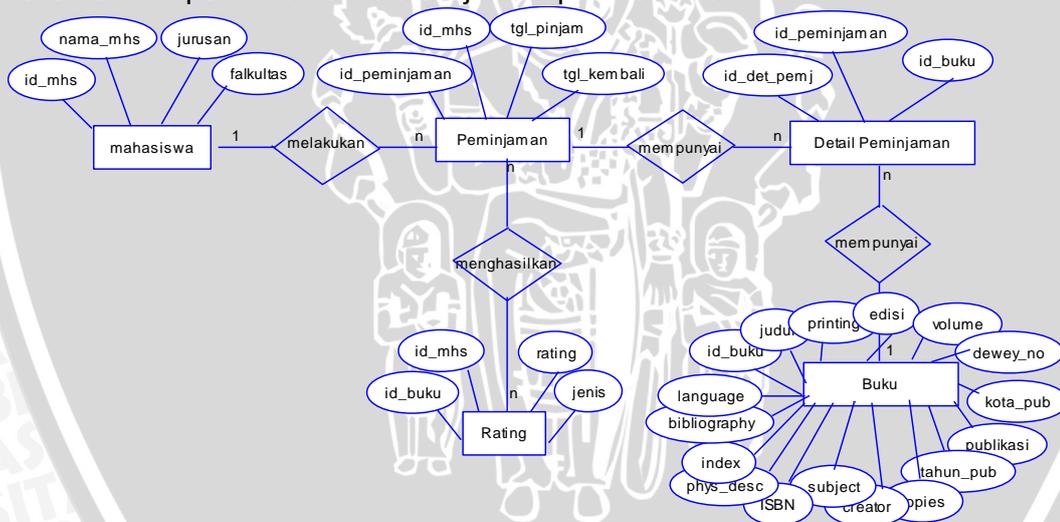
Tabel *rating*(penilaian) digunakan untuk menyimpan data penilaian pada setiap *item*(buku), tabel ini menyimpan data id_buku, id_mahasiswa, *rating* dan jenis penilaian.

Tabel 4. 11 Tabel *Rating*

Nama Field	Tipe dan lebar	Keterangan
id_buku	INT (20)	Primary key
id_mahasiswa	INT (11)	Primary key
Rating	INT(10)	-
Jenis	INT(10)	-

4.3.5 Perancangan *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang dipakai untuk mendokumentasikan data dengan mengidentifikasi jenis entitas dan hubungannya. ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang merepresentasikan seluruh fakta yang ditinjau dari keadaan nyata. Pada ERD dapat dilihat terdapat 5 tabel di mana tabel buku berelasi dengan tabel mahasiswa yang akan menghasilkan tabel peminjaman. Tabel peminjaman ini akan di konversikan menjadi nilai *rating* yang disimpan kedalam tabel *rating*. Rancangan ERD sistem rekomendasi pencarian buku ditunjukkan pada Gambar 4.6



Gambar 4. 6 Rancangan ERD Sistem Rekomendasi Pencarian Buku

Dalam rancangan ERD di atas terdapat 5 entitas, yaitu mahasiswa, buku, peminjaman, detail peminjaman, dan *rating*. Entitas mahasiswa memiliki atribut id_mahasiswa (*primary key*), nama_mahasiswa, jurusan, dan fakultas. Entitas buku, memiliki atribut id_buku (*primary key*), judul, dan seterusnya. Entitas peminjaman memiliki atribut id_peminjaman (*primary key*), id_mahasiswa, tgl_pinjam, dan tgl_kembali. Entitas detail peminjaman memiliki atribut id_de_peminjaman (*primary key*), id_peminjaman, id_buku. Entitas *rating* memiliki atribut id_rating (*primary key*), id_buku(*primary key*), id_mahasiswa(*primary key*), nilai_rating dan jenis. Entitas mahasiswa dapat melakukan peminjaman sehingga entitas mahasiswa berelasi dengan entitas

peminjaman. Entitas *det_pinjam* berelasi dengan entitas buku. Entitas peminjaman berelasi dengan entitas *rating* yang berfungsi sebagai data *rating* yang bersal dari entitasi peminjaman.

4.3.6 Perancangan Antarmuka

Pada sub bab ini, hanya menjelaskan rancangan antarmuka pada proses-proses utama dari sistem rekomendasi buku perpustakaan seperti tampilan pencarian buku, tampilan detail dan rekomendasi buku dan tampilan analisis perhitungan. Antarmuka yang dibangun adalah sebagai berikut:

4.3.6.1 Tampilan Utama Pencarian Buku

Tampilan pada halaman pencarian buku ini adalah tampilan dari sistem saat mahasiswa melakukan pencarian buku. Pada tampilan ini terdapat sebuah tombol pencarian yang akan digunakan oleh mahasiswa untuk mencari buku yang diinginkan di mana hasil pencarian ini akan menjadi nilai *rating* mahasiswa. Tampilan pencarian dapat dilihat pada Gambar 4.7.

The image shows a web interface for a library catalog. At the top, it says "CENTRAL LIBRARY CATALOG". To the right, there is a button labeled "BUKU REKOMENDASI". Below this, there is a search form. The form has a text input field labeled "Masukan NIM". Below that, there is a "SEARCH by" section with two dropdown menus: "Judul" and "Tahun". Below the dropdowns, there is a text input field labeled "Kata kunci" and a "SEARCH" button. At the bottom of the form, there is a large empty box labeled "HASIL PENCARIAN".

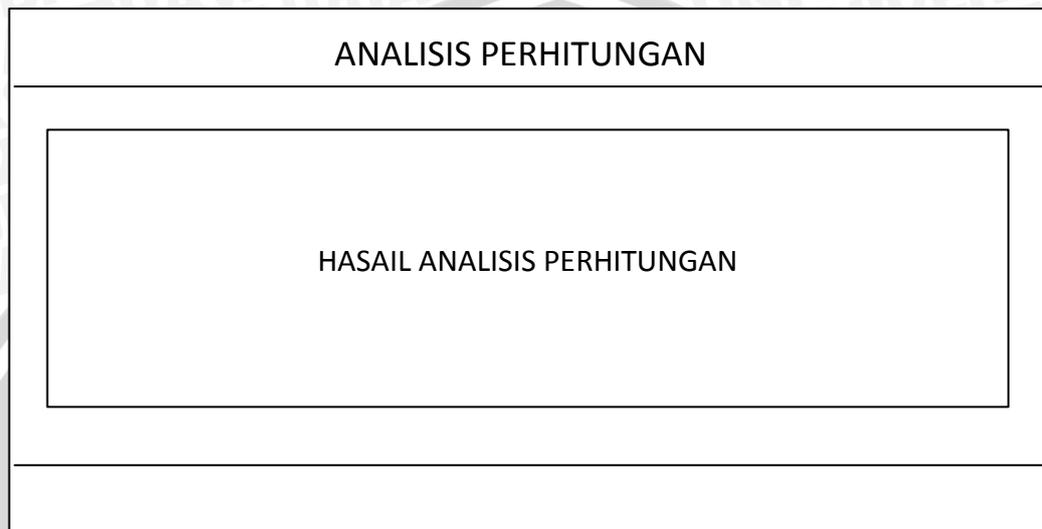
Gambar 4. 7 Halaman Pencarian Buku

4.3.6.2 Tampilan Detail Buku dan Hasil Rekomendasi

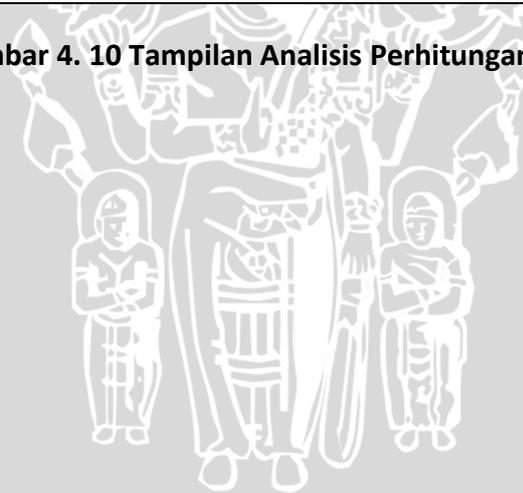
Tampilan pada halaman detail buku ini berfungsi untuk menampilkan detail buku dari buku yang dicari oleh mahasiswa dan hasil rekomendasi buku terhadap mahasiswa yang melakukan pencarian buku. Dengan menekan *list* buku pada tampilan menu utama pencarian buku, akan muncul detail buku. Tampilan detail buku dapat dilihat pada Gambar 4.8.

4.3.6.4 Tampilan Analisis perhitungan

Antarmuka Analisis perhitungan pada gambar di bawah ini merupakan antarmuka untuk menampilkan hasil perhitungan dari proses penentuan kemiripan hingga nilai prediksi. Tampilan ini berfungsi untuk mengetahui proses perhitungan dari sistem rekomendasi pencarian buku di perpustakaan. Analisis perhitungan dapat dilihat pada Gambar 4.10.

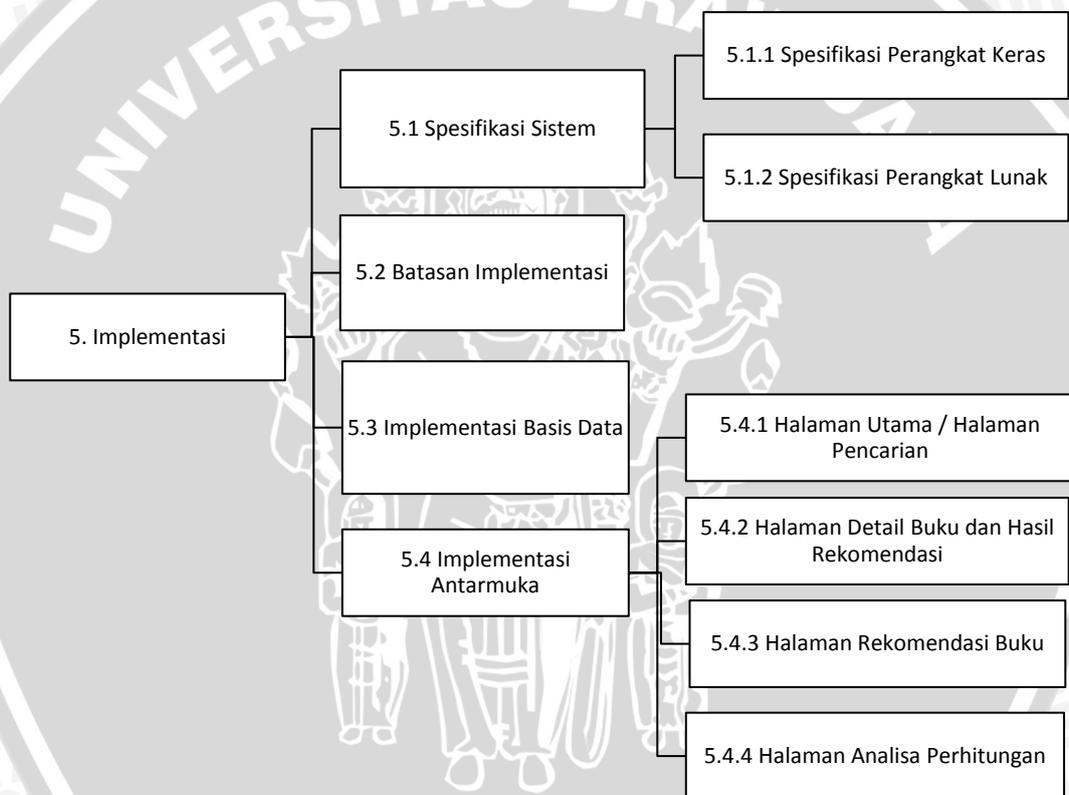


Gambar 4. 10 Tampilan Analisis Perhitungan



BAB 5 IMPLEMENTASI

Bab ini akan membahas tentang implementasi sistem rekomendasi yang dibangun untuk merekomendasikan buku kepada mahasiswa menggunakan metode *Item-based Collaborative Filtering*. Bab ini juga akan membahas tentang spesifikasi dan lingkungan sistem, implementasi algoritma dalam sistem serta implementasi antarmuka sistem berdasarkan perancangan yang telah dilakukan sebelumnya. Berikut pohon implementasi ditunjukkan pada Gambar 5.1 sebagai berikut.



Gambar 5. 1 Pohon Implementasi

5.1 Spesifikasi sistem

Sub bab ini akan menjelaskan tentang spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam mengimplementasikan sistem yang akan dibuat pada penelitian ini.

5.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem rekomendasi pencarian buku menggunakan metode *Item-based Collaborative Filtering* ditunjukkan pada Tabel 5.1.

Tabel 5. 1 Spesifikasi Perangkat Keras

Perangkat Keras	Keterangan
Processor	Prosesor Intel(R) Core i3 CPU M350 @2.27GHz(4 CPUs)
Memory RAM	4096MB
Harddisk	265 GB
Monitor	14"

5.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem rekomendasi peminjaman buku menggunakan metode *Item-based Collaborative Filtering* ditunjukkan ditunjukkan pada Tabel 5.2.

Tabel 5. 2 Spesifikasi Perangkat Lunak

Perangkat Lunak	Keterangan
Sistem Operasi	Windows 8.1 64-bit
Bahasa Pemrograman	C#
Tools	Visual Studio 2010
DBMS	MySQL

5.2 Batasan Implementasi

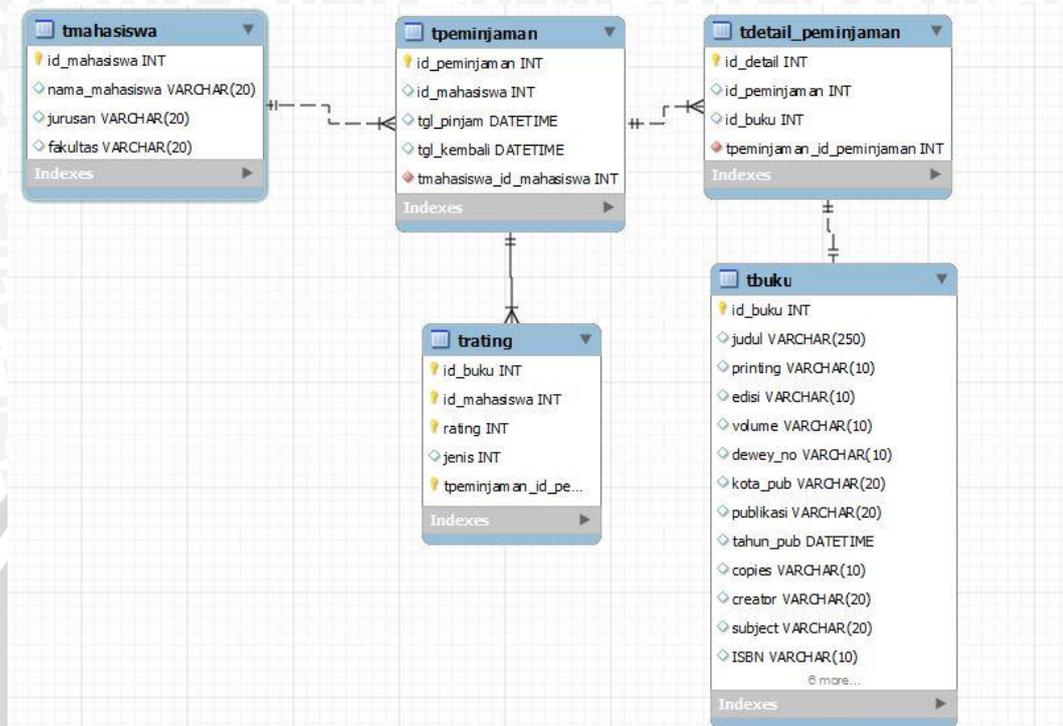
Berikut ini merupakan batasan implementasi pada pembangunan sistem rekomendasi pencarian buku.

1. Sistem yang dibangun berbasis *desktop* dengan menggunakan bahasa pemrograman C# dan MySQL DBMS.
2. Data yang digunakan adalah data buku yang terdapat di perpustakaan Universitas Brawijaya, tidak termasuk laporan kerja praktek (KP), laporan skripsi, laporan thesis dan disertasi.
3. Data peminjaman yang digunakan adalah data *dummy* dengan 31 data transaksi.
4. *Filtering* yang digunakan hanya *filtering* jurusan.
5. Menerapkan metode *Item-based collaborative filtering* pada proses perhitungan penentuan rekomendasi buku.
6. Keluaran yang dihasilkan oleh sistem yaitu berupa rekomendasi buku kepada mahasiswa.
7. *User* utama pada sistem ini adalah mahasiswa.

5.3 Implementasi Basis Data

Implementasi basis data merupakan proses penyimpanan data-data dari sistem ke dalam sebuah *database*. Basis data yang digunakan dalam implementasi ini adalah DBMS MySQL. Penyimpanan data dikelompokkan

kedalam tabel-tabel kemudian yang digambarkan dalam diagram model konseptual *Entity Relationship* yang ditunjukkan pada Gambar 5.2.



Gambar 5. 2 Implementasi Diagram ER Sistem

5.4 Implementasi Algoritma

Dari perancangan implementasi yang telah dibahas pada bab 4, maka disini akan dijelaskan mengenai implementasi algoritma dengan metode *Item-Based Collaborative Filtering* sesuai dengan yang ada pada perancangan tersebut. Implementasi algoritma menggunakan bahasa pemrograman C# dan berbasis desktop.

5.4.1 Implementasi Algoritma Pembentukan Matriks

Tahap awal dari metode ini adalah penyusunan matriks yang berisikan data mahasiswa, data buku, dan transaksi peminjaman. Relasi antara mahasiswa dengan buku yang pernah dipinjam digambarkan dengan angka 0 dan 1, di mana 0 artinya tidak pernah dipinjam dan 1 artinya pernah dipinjam. Pengisian data dengan cara mengkonversi data peminjaman sebagai data penilaian mahasiswa terhadap buku berupa matriks penilaian mahasiswa terhadap buku. Kode program pada tahapan ini dapat dilihat pada Kode Program 5.1.

```

1. void matriks()
2. {
3.     workTable.Clear();
4.     dvdiketahui = null;
5.     dvdicari = null;
6.     dvhasil = null;
7.     dtdicaridomain = null;
8.     dtdiketahuidomain = null;

```

```

9.         dtdicari = null;
10.        dtdiketahui = null;
11.        dtrata = null;
12.        var koneksi = new ClassKoneksi();
13.        jurusan = "";
14.        /////// Pembentukan Matrik dan filtering jurusan////////////////////
15.        DataSet dus = new DataSet();
16.        dtdiketauidomain = koneksi.Ctable("Select
17. id_buku,id_mahasiswa,rating from trating where
18. trating.id_mahasiswa = '" + FormMain.idmhscatalog + "' AND rating
19. = 1");
20.        var rowAsString = string.Join(", ",
21. dtdiketauidomain.Rows.Cast<DataRow>().Select(r =>
22. r.ItemArray[0].ToString()));
23.        dtdicaridomain = koneksi.Ctable("Select
24. trating.id_buku from trating where trating.id_buku NOT IN (" +
25. rowAsString + ") Group by trating.id_buku Order by
26. trating.id_buku");
27.        //// jurusan = nim
28.        dus = koneksi.getRows("select jurusan from tmahasiswa
29. where id_mahasiswa = '" + FormMain.idmhscatalog + "'");
30.        jurusan =
31. Convert.ToString(dus.Tables["hasil"].Rows[0][0]);
32.        dtdiketahui = koneksi.Ctable("Select
33. trating.id_buku,trating.id_mahasiswa,trating.rating from trating
34. INNER JOIN tmahasiswa ON trating.id_mahasiswa =
35. tmahasiswa.id_mahasiswa where trating.id_mahasiswa <> '" +
36. FormMain.idmhscatalog + "' AND tmahasiswa.jurusan = '" + jurusan +
37. "' Order by trating.id_buku, trating.id_mahasiswa");
38.        dtdicari = dtdiketahui;
39.        //// nilai rata rata
40.        dtrata = koneksi.Ctable("Select trating.id_mahasiswa,
41. SUM(IFNULL(trating.rating,0))/count(trating.rating) as rata2 from
42. trating INNER JOIN tmahasiswa ON trating.id_mahasiswa =
43. tmahasiswa.id_mahasiswa where trating.id_mahasiswa <> '" +
44. FormMain.idmhscatalog + "' AND tmahasiswa.jurusan = '" + jurusan +
45. "' Group By id_mahasiswa");
        /////////////// Akhir Pembentukan Matrik
    }

```

Kode Program 5. 1 Implementasi Algoritma Pembentukan Matriks

5.4.2 Implementasi Algoritma Perhitungan Nilai Kemiripan(*Similarity*)

Tahap selanjutnya adalah perhitungan nilai *similarity* antara *item* dengan *item* lainnya. Disinilah inti dari metode *Item-Based Collaborative Filtering*, di mana perhitungan kemiripan dilihat dari perbandingan kemiripan penilaian antar *item* yang diberikan *user*. Penilaian disini adalah pernah tidaknya *user* meminjam suatu buku. Nilai kemiripan akan dicari melalui berapa nilai buku yang pernah dipinjam pada masa lalu, dan juga untuk proses *filtering*nya dilakukan *filtering* jurusan untuk memaksimalkan efisiensi waktu proses rekomendasi. Kode program perhitungan nilai *similarity* dapat dilihat pada Kode Program 5.2.

```

1.         for (int i = 0; i <= dtdicaridomain.Rows.Count - 1;
2.     i++)
3.         {
4.             var idbuku2 =
5.     Convert.ToString(dtdicaridomain.Rows[i][0]);

```

```

6.         idbk2 = idbuku2;
7.         dvdicari.RowFilter = "id_buku = " + idbuku2;
8.         for (int j = 0; j <= dtdiketahuidomain.Rows.Count
9. - 1; j++)
10.        {
11.            var idbuku =
12. Convert.ToString(dtdiketahuidomain.Rows[j][0]);
13.            idbk = idbuku;
14.            dvdiketahui.RowFilter = "id_buku = " + idbuku;
15.            double hasil = 0;
16.            double hasilpangkat = 0;
17.            double hasilpangkat2 = 0;
18.            double hasilakar = 0;
19.            double hasilakar2 = 0;
20.            double hasilakhir = 0;
21.            for (int k = 0; k <= dvdiketahui.Count - 1;
22. k++)
23.            {
24.                var rt = dvdiketahui[k][2];
25.                var rt2 = dvdicari[k][2];
26.                //// nilai rata2
27.                var rata = dtrata.Rows[k][1];
28.                // Menghitung kemiripan
29.                hasil = hasil +
30. ((Math.Round(Convert.ToDouble(rt2) - Convert.ToDouble(rata), 3)) *
31. Math.Round((Convert.ToDouble(rt) - Convert.ToDouble(rata)), 3));
32.                hasilpangkat2 = hasilpangkat2 +
33. (Math.Pow(Convert.ToDouble(rt2) - Convert.ToDouble(rata), 2));
34.                hasilpangkat = hasilpangkat +
35. (Math.Pow(Convert.ToDouble(rt) - Convert.ToDouble(rata), 2));
36.            }
37.            hasilakar =
38. Math.Round(Math.Sqrt(hasilpangkat), 3);
39.            hasilakar2 =
40. Math.Round(Math.Sqrt(hasilpangkat2), 3);
41.            hasilakhir = Math.Round(hasil, 3) /
42. Math.Round((hasilakar * hasilakar2), 3);
43.            //akhir perhitungan similarity

```

Kode Program 5. 2 Implementasi Algoritma Perhitungan Nilai Kemiripan

5.4.3 Implementasi Algoritma Perhitungan Nilai Prediksi

Langkah terakhir dari metode *Item-Based Collaborative Filtering* adalah perhitungan nilai prediksi rekomendasi buku yang akan diberikan kepada *user*. Tahap tersebut menggunakan perhitungan *Weighted Sum*. Buku yang akan direkomendasikan adalah buku yang tidak pernah dipinjam oleh *user*, tapi pernah dipinjam oleh *user* lain yang nilai *similarity*-nya paling tinggi. Tahapan ini akan dijelaskan pada Kode Program 5.3.

```

1. //mengisi data
2.         DataRow workRow = workTable.NewRow();
3.         workRow[0] = idbk2;
4.         workRow[1] = idbk;
5.         workRow[2] = Math.Round(hasilakhir, 3);
6.         workRow[3] = dtdiketahuidomain.Rows[j][2];
7. // menghitung nilai prediksi
8.         workRow[4] =

```

```

9.   Math.Round(Math.Round(Math.Round(hasilakhir, 3) *
10.  Convert.ToDouble(dtdiketahuidomain.Rows[j][2]), 3) /
11.  Math.Abs(Convert.ToDouble(dtdiketahuidomain.Rows[j][2])), 3);
12.  workTable.Rows.Add(workRow);
    
```

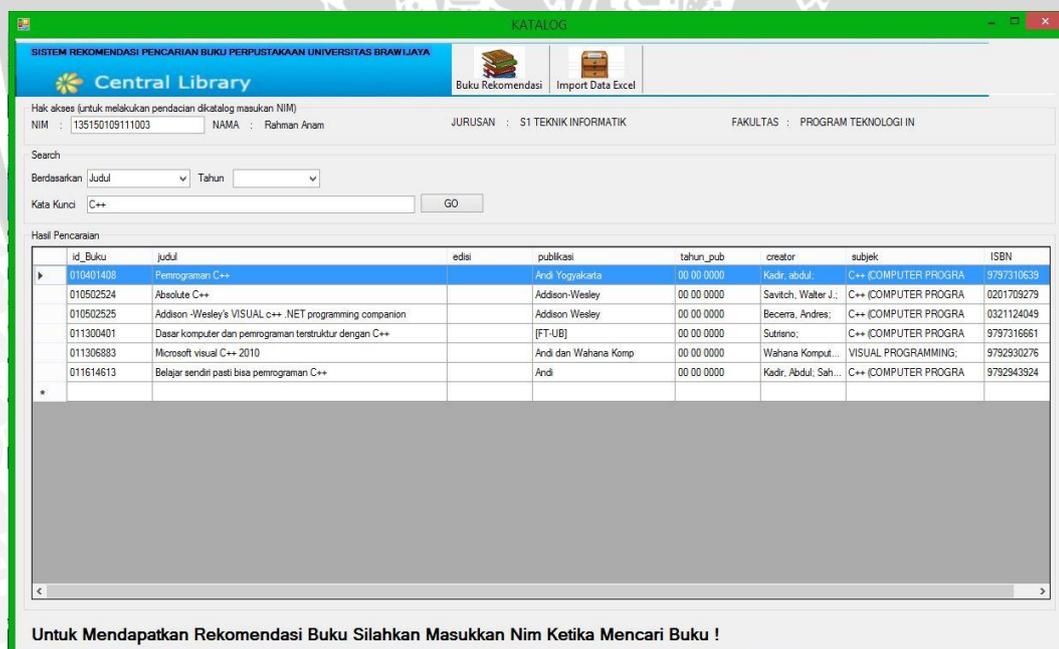
Kode Program 5. 3 Implementasi Algoritma Perhitungan Nilai Pediksi (Weighted Sum)

5.5 Implementasi Antarmuka

Antarmuka aplikasi sistem rekomendasi buku ini digunakan *user* untuk berinteraksi dengan sistem perangkat lunak yang dibangun. Implementasi antarmuka terdiri dari halaman-halaman pada sistem yang telah dibangun berdasarkan bab perancangan antarmuka. Berikut ini merupakan tampilan halaman dari implementasi antarmuka sistem rekomendasi buku Menggunakan metode *Item-Based Collaborative Filtering*.

5.5.1 Halaman Utama Pencarian Buku

Halaman utama pencarian pada sistem rekomendasi ini terdapat menu buku rekomendasi untuk melihat rekomendasi buku. Halaman pencarian buku ini dapat diakses oleh mahasiswa tanpa melakukan proses login hanya dengan memasukan nim untuk mendapatkan hasil rekomendasi. Pada halaman ini juga terdapat hasil rekomendasi buku terhadap *user*/mahasiswa aktif. Berikut tampilan halaman utama pencarian dan rekomendasi buku ditunjukkan pada Gambar 5.3.

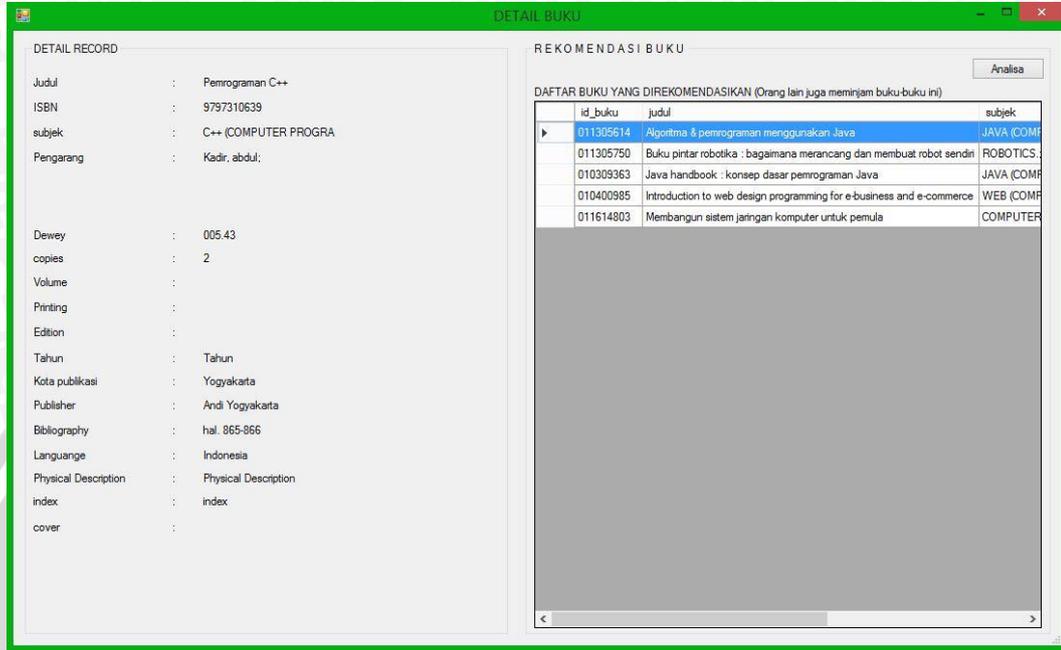


Gambar 5. 3 Halaman Utama Pencarian dan Hasil Rekomendasi Buku

5.5.2 Tampilan Detail Buku Dan Hasil Rekomendasi Buku

Tampilan pada halaman detail buku dan hasil rekomendasi buku ini berfungsi untuk menampilkan detail buku dan hasil rekomendasi buku terhadap mahasiswa yang melakukan pencarian buku. Dengan melihat detail buku bahwa

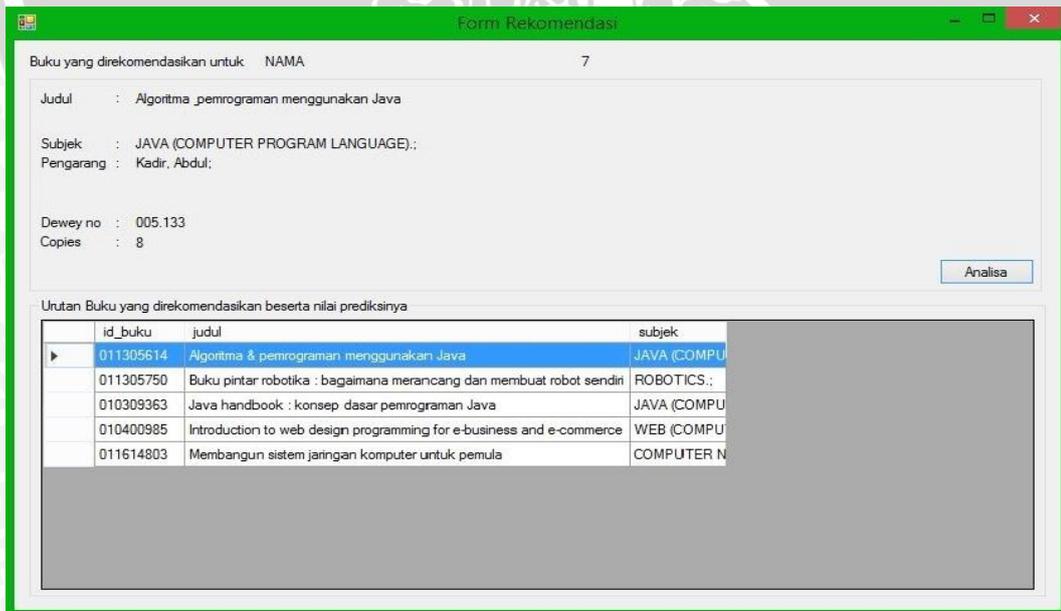
diasumsikan mahasiswa suka dengan buku tersebut sehingga didata sebagai penilaian terhadap buku. Dengan menekan list buku pada tampilan menu utama pencarian buku, akan muncul detail buku. Tampilan detail buku dapat dilihat pada Gambar 5.4.



Gambar 5. 4 Halaman Detail Buku

5.5.3 Halaman Hasil Rekomendasi

Pada tampilan ini akan ditampilkan buku-buku yang direkomendasi dengan diurutkan dari nilai prediksi tertinggi kepada mahasiswa (*user active*). Tampilan hasil rekomendasi dapat dilihat pada Gambar 5.5.



Gambar 5. 5 Halaman Hasil Rekomendasi

5.5.4 Halaman Analisis Perhitungan

Halaman analisis perhitungan berfungsi untuk menampilkan hasil perhitungan dari tahapan metode *item-based collaborative filtering* yang diantaranya perhitungan nilai kemiripan dengan menggunakan *adjusted cosine similarity*, dan untuk perhitungan prediksi menggunakan metode *weighted sum*. Tampilan analisis perhitungan dapat dilihat pada Gambar 5.6.

HASIL PERHITUNGAN MEAN ADJUSTED SCORE
RATA-RATA RATING MAHASISWA TERHADAP BUKU (ITEM)

id_mahasiswa	rata2
1	0.5000
2	0.8333
3	0.8333
4	0.6667
5	0.8333
6	0.8333

ID BUKU BELUM DIPINJAM (DICARI)

id_buku
2
6

ID BUKU YANG SUDAH DIPINJAM (DIKETAHUI)

id_buku	id_mahasiswa	rating
1	7	2
3	7	1
4	7	1
5	7	1

ID MAHASISWA
7

HASIL PERHITUNGAN SIMILARITY ITEM

bukuA	bukuB	mirip	n_rating	prediksi
2	1	-0.059	2	-0.059
2	3	-0.059	1	-0.059
2	4	0.225	1	0.225
2	5	-0.266	1	-0.266
6	1	-0.749	2	-0.749
6	3	-0.749	1	-0.749
6	4	0.203	1	0.203
6	5	-0.576	1	-0.576

RATING DARI BUKU YANG PERNAH BELUM PERNAH DIPINJAM

id_buku	id_mahasiswa	rating
6	1	0
6	2	1
6	3	0
6	4	0
6	5	0
6	6	0

RATING DARI BUKU YANG PERNAH DIPINJAM

id_buku	id_mahasiswa	rating
5	1	1
5	2	0
5	3	1
5	4	1
5	5	1
5	6	1

HASIL PERHITUNGAN PREDIKSI TOP N ITEM

bukuA	bukuB	mirip	n_rating	prediksi
2	4	0.225	1	0.225
6	4	0.203	1	0.203

ANALISA

Gambar 5. 6 Halaman Analisis Perhitungan

BAB 6 PENGUJIAN

Setelah tahap implementasi dilakukan maka tahap selanjutnya adalah melakukan pengujian. Pengujian sistem dilakukan untuk mengetahui sistem yang dibangun apakah sesuai dengan hasil analisis dan perancangan. Sehingga diharapkan dapat memberikan kesimpulan akhir yang dapat menjawab rumusan masalah yang sudah ditentukan di awal. Pengujian pertama yaitu Pengujian akurasi prediksi sistem dilakukan dengan melihat MAE (*Mean Absolute Error*) pengujian ini berfungsi untuk mengetahui tingkat kesalahan prediksi dari sistem rekomendasi, selanjutnya pengujian relevansi presisi dengan DCC (*Dewey Decimal Clasification*) pengujian ini berfungsi mengetahui akurasi relevansi dengan nomor DDC dan pengujian waktu eksekusi sistem dalam memproses rekomendasi pencarian buku, pengujian ini bertujuan untuk mengetahui waktu tempuh dalam melakukan proses perhitungan sistem dalam melakukan rekomendasi.

6.1 Pengujian MAE (Mean Absolute Error)

Pengujian MAE ini bertujuan yaitu menghitung tingkat *error* hasil rekomendasi yang diberikan sistem dengan menggunakan persamaan 2.4. Data yang diuji adalah data uji yang nilai *rating*-nya dikosongkan dengan tingkat *Sparsity* tertentu. sehingga dapat diketahui tingkat akurasi berdasarkan tingkat *Sparsity* data. Data akan diujikan pada 5 *user* yang diambil secara acak. Nilai MAE diperoleh dengan membandingkan *rating* yang diberikan *user* dengan *rating* yang diberikan sistem. Nilai MAE diperoleh dengan membandingkan *rating* yang diberikan *user* dengan *rating* yang diberikan sistem. Berikut ini contoh perhitungan nilai MAE untuk hasil rekomendasi buku yang diberikan kepada *user* Mahasiswa 1. Untuk detail hasil perhitungan tiap mahasiswa dapat dilihat pada lampiran C.3.

Pada saat tingkat *Sparsity* 30% dari 31 transaksi peminjaman, pengujian pada mahasiswa 1 dengan NIM 125150200111116, untuk buku pertama diperoleh nilai *rating* prediksi 0,704 dan *rating* asli 1. Buku ke 2 nilai *rating* prediksi 0,704 dan *rating* asli 1. Buku ke 3 nilai *rating* prediksi 0,495 dan *rating* asli 1. Buku ke 4 nilai *rating* prediksi 0,495 dan *rating* asli 1. Buku ke 5 nilai *rating* prediksi 0,403 dan *rating* asli 1. Berikut contoh perhitungan MAE menggunakan persamaan (2.4).

$$\begin{aligned}
 MAE &= \frac{\sum_{i=1}^n |p_i - q_i|}{N} \\
 &= \frac{|0,704 - 1| + |0,704 - 1| + |0,495 - 1| + |0,495 - 1| + |0,403 - 1|}{5} \\
 &= \frac{2,199}{5} \\
 &= 0,439
 \end{aligned}$$

Pada saat tingkat *Sparsity* 50% dari 31 transaksi peminjaman, pengujian pada mahasiswa 1 dengan NIM 125150200111116, untuk buku pertama diperoleh nilai *rating* prediksi 0,575 dan *rating* asli 1. Buku ke 2 nilai *rating* prediksi 0,574. Berikut contoh perhitungan MAE menggunakan persamaan (2.4)

$$\begin{aligned}
 MAE &= \frac{\sum_{i=1}^n |p_i - q_i|}{N} \\
 &= \frac{|0,575 - 1| + |0,574 - 1|}{2} \\
 &= \frac{0,851}{2} \\
 &= 0,4255
 \end{aligned}$$

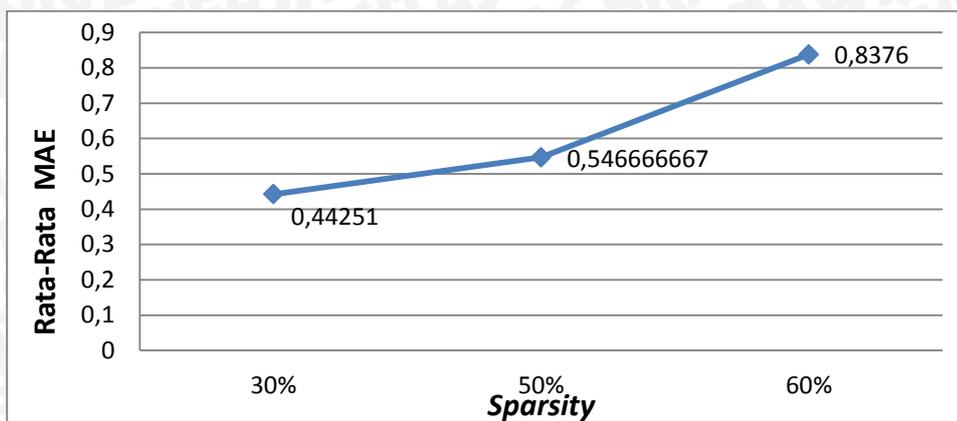
Pada saat tingkat *Sparsity* 60% dari 31 transaksi peminjaman, pengujian pada mahasiswa 1 dengan NIM 125150200111116, tidak mendapatkan rekomendasi buku dari sistem, sehingga mendapat nilai *rating* prediksi 0 dan *rating* asli 1. Berikut contoh perhitungan MAE menggunakan persamaan (2.4)

$$\begin{aligned}
 MAE &= \frac{\sum_{i=1}^n |p_i - q_i|}{N} \\
 &= \frac{|0 - 1|}{1} \\
 &= \frac{1}{1} \\
 &= 1
 \end{aligned}$$

Setelah dilakukan pengujian terhadap kelima *user* dengan dengan tingkat *Sparsity* yang berbeda, maka didapatkan hasil tingkat akurasi prediksi untuk setiap tingkatan *Sparsity*. nilai MAE keseluruhan pengujian dapat dilihat pada tabel 6.1 dan Gambar 6.1 sebagai berikut :

Tabel 6. 1 Tabel Hasil Pengujian MAE(Mean Absolute Error)

Tingkat <i>Sparsity</i> Data % dari 31 Data Transaksi	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 6. 1 Grafik Hasil Pengujian MAE (Mean Absolute Error)

Dari hasil pengujian pada tabel 6.1 dan gambar 6.1 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* di mana 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 Classifications*)

Pada pengujian relevansi presisi dengan DDC, Data latih peminjaman yang digunakan data yang memiliki tingkat *Sparsity* data 30% dari 31 Data Transaksi Peminjaman. Untuk detail hasil pengujian dapat dilihat pada lampiran C.2. Pengujian akurasi ini akan melakukan perbandingan relevansi antara nomor DDC buku yang direkomendasikan dengan nomor DDC buku yang dipilih. Pengujian ini bertujuan untuk menguji tingkat akurasi. Hasil pengujian akurasi terdapat pada Tabel 6.2 dan Gambar 6.2.

Tabel 6. 2 Tabel Hasil Pengujian dengan Nomor 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	125150200111116	5	1	4	0,2
2	125150200111119	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*.

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.

6.3.1 Hasil Pengujian Waktu Eksekusi Sistem Berdasarkan Jumlah *Item*

Setelah pengujian waktu berdasarkan tingkat *item* dilakukan dengan beberapa parameter yang sudah ditentukan, maka diketahui hasil pengujian untuk setiap parameter diperoleh durasi waktu dengan dinyatakan dalam satuan detik, hasil pengujian berdasarkan jumlah *item* dapat dilihat pada tabel 6.3 dan Gambar 6.2.

Tabel 6. 3 Tabel Hasil Pengujian Waktu Eksekusi Berdasarkan Jumlah *Item*

Jumlah <i>Item</i> /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 6. 2 Grafik Hasil Pengujian Waktu Eksekusi Berdasarkan Jumlah Item

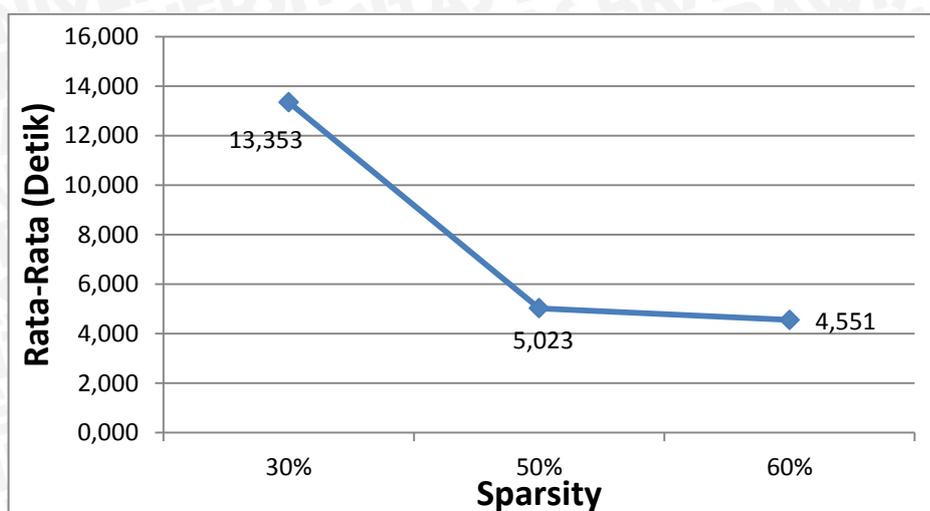
Pada Tabel 6.3 dan Gambar 6.2 menjelaskan pengujian waktu eksekusi yang dibutuhkan untuk setiap pengujian yang dilakukan. Dari Hasil pengujian secara keseluruhan, waktu eksekusi meningkat jika jumlah *item* yang diolah bertambah banyak. Sehingga pengujian ini menghasilkan waktu tercepat yaitu ketika 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.

6.3.2 Hasil Pengujian Waktu Eksekusi Sistem Berdasarkan Tingkat Sparsity

Pengujian waktu berdasarkan tingkat *Sparsity* data transaksi dilakukan dengan beberapa parameter yang sudah ditentukan, maka diketahui hasil pengujian untuk setiap parameter diperoleh durasi waktu dengan dinyatakan dalam detik, hasil pengujian berdasarkan tingkat *Sparsity* dapat dilihat pada tabel 6.4 dan Gambar 6.3.

Tabel 6. 4 Tabel Hasil Pengujian Waktu Eksekusi Berdasarkan Tingkat Sparsity

Tingkat Sparsity % dari 31 Data Transaksi	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 6. 3 Hasil Pengujian Waktu Eksekusi Berdasarkan Tingkat *Sparsity*

Tabel 6.4 dan Gambar 6.3 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 berdasarkan jumlah *item*. Di mana 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 Analisis Hasil Pengujian

Pengujian tingkat akurasi pada sistem rekomendasi buku perpustakaan dengan metode *item-base collaborative filtering* yang telah dibangun mampu mengatasi kekosongan data dengan tingkat *Sparsity* sebanyak 60%, di mana nilai MAE yang dihasilkan rata-rata sebesar 0,8376. Sedangkan hasil terbaik diperoleh ketika tingkat *Sparsity* sebanyak 30%, di mana nilai MAE yang dihasilkan rata-rata sebesar 0,4425

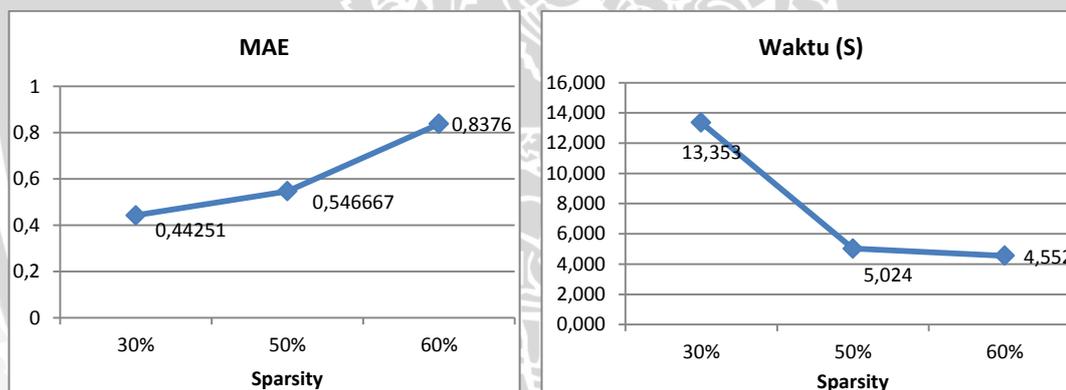
Dari hasil pengujian akurasi presisi dengan membandingkan nomor DDC buku yang dihasilkan oleh sistem dengan nomor DDC buku yang dicari menghasilkan kesimpulan bahwa nilai rata-rata precision untuk buku yang direkomendasikan sistem kepada mahasiswa rata-rata 0,579286. Dari hasil nilai precision tersebut dapat dikatakan bahwa sistem rekomendasi buku perpustakaan dengan metode *collaborative filtering* memberikan hasil rekomendasi buku yang relevan dengan *user* sebanyak 57%.

Dari pengujian waktu eksekusi terlihat bahwa jumlah *item* mempengaruhi waktu eksekusi sistem. Meningkatnya jumlah *item* akan menghasilkan perbedaan waktu yang signifikan. Terlihat pada data uji yang berjumlah 700 *item* rata-rata waktu eksekusi sistem rata-rata 4,451 detik dan ketika jumlah *item* ditambah hingga berjumlah ribuan waktu eksekusi meningkat seperti pada data yang berjumlah 1000 *item* waktu eksekusi yang diperlukan rata-rata 18,536 detik.

Waktu pemrosesan yang paling lama yaitu ketika sistem mengolah data sebanyak 4000 *item* di mana waktu eksekusi yang dibutuhkan rata-rata 69,894 detik. tingkat *Sparsity* juga mempengaruhi waktu eksekusi sistem. Semakin tinggi tingkat *Sparsity* maka waktu eksekusi yang dibutuhkan semakin kecil dan sebaliknya semakin rendah tingkat *Sparsity* maka waktu eksekusi yang dibutuhkan akan semakin meningkat. 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. Perbandingan antara nilai MAE dan waktu eksekusi dapat dilihat pada Tabel 6.5 dan Gambar 6.4

Tabel 6. 5 Perbandingan Rata-Rata 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 6. 4 Perbandingan Rata-Rata MAE Dengan Waktu Eksekusi Sistem

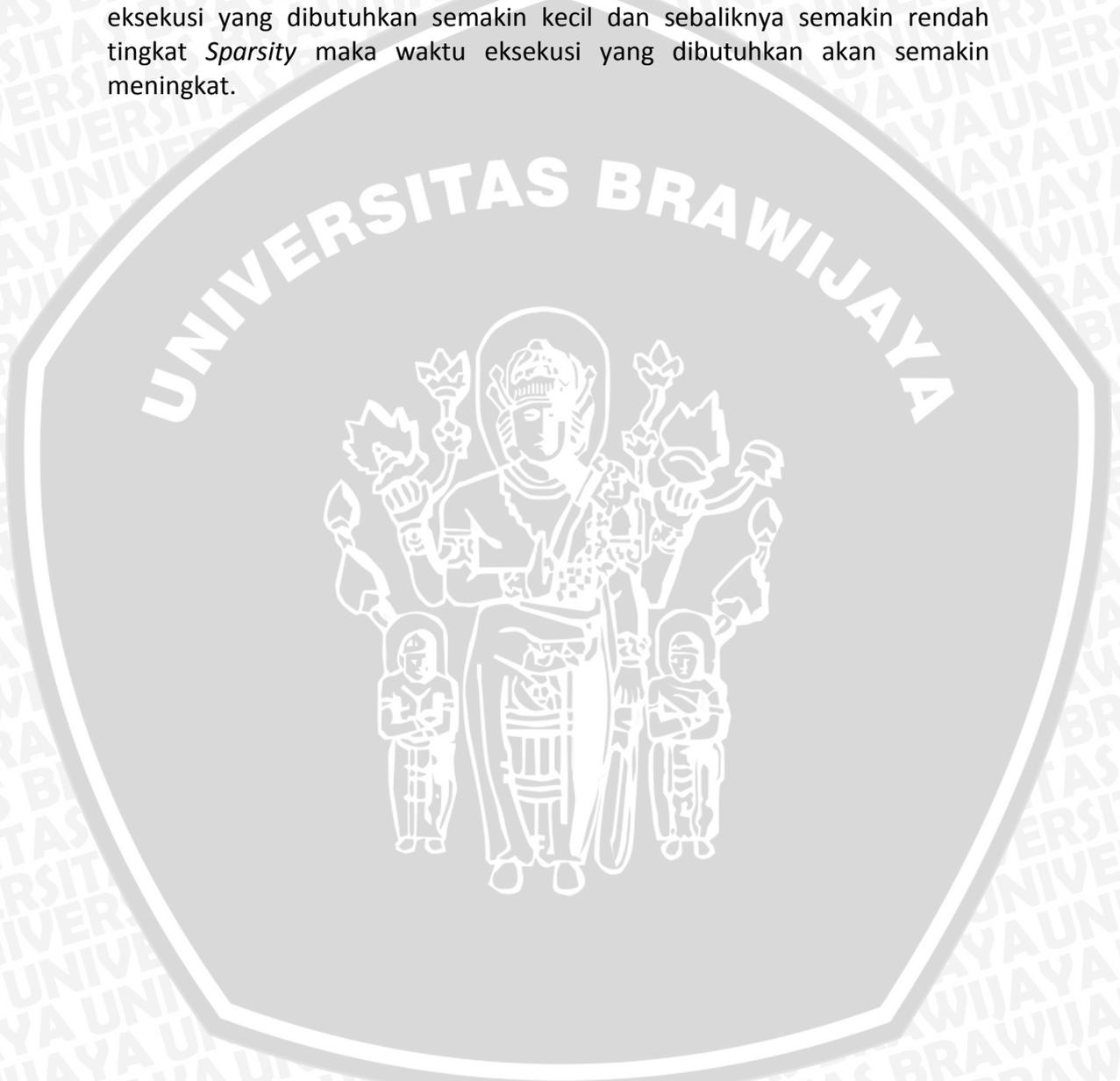
6.5 Kesimpulan Hasil Pengujian

Dari hasil pengujian-pengujian yang sudah dilakukan, dapat diambil kesimpulan hasil penelitain sebagai berikut :

1. Dari segi keakuratan hasil rekomendasi, nilai MAE dipengaruhi oleh tingkat *Sparsity* data. Semakin tinggi *Sparsity* data maka nilai MAE akan semakin tinggi. Pada sistem ini dapat mengatasi *Sparsity* data hingga 60% dengan nilai MAE 0,8376.
2. Dari hasil pengujian akurasi presisi dengan membandingkan nomor DDC yang dihasilkan oleh sistem dengan buku yang dicari menghasilkan kesimpulan bahwa nilai rata-rata precision untuk buku yang direkomendasikan sistem kepada mahasiswa rata-rata 0,579286. Dari hasil nilai precision tersebut dapat dikatakan bahwa sistem rekomendasi buku

perpustakaan dengan metode *item-base collaborative filtering* memberikan hasil rekomendasi buku yang relevan dengan *user* sebanyak 57% dari data dengan tingkat *Sparsity* 30% dari 31 transaksi peminjaman.

3. Dari segi waktu eksekusi, banyaknya waktu yang dibutuhkan untuk menghasilkan rekomendasi dipengaruhi oleh jumlah data yang diproses serta tingkat *Sparsity* data. Semakin tinggi tingkat *Sparsity* maka waktu eksekusi yang dibutuhkan semakin kecil dan sebaliknya semakin rendah tingkat *Sparsity* maka waktu eksekusi yang dibutuhkan akan semakin meningkat.



BAB 7 KESIMPULAN

7.1 Kesimpulan

Berdasarkan analisis, perancangan, implementasi dan pengujian sistem rekomendasi buku perpustakaan dengan menggunakan metode *item-based collaborative filtering*, maka dapat diambil beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Hasil pengujian yang ditunjukkan pada Tabel 6.1 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 ditunjukkan pada Tabel 6.2 menggambarkan akurasi apabila dibandingkan dengan nomor DDC menghasilkan tingkat akurasi 57% dari buku yang direkomendasikan dengan data *Sparsity* 30% dari 31 transaksi peminjaman.
3. Waktu eksekusi yang dibutuhkan dipengaruhi oleh jumlah data dan tingkat *Sparsity*. Pada Tabel 6.3 dapat disimpulkan semakin besar jumlah data yang digunakan maka semakin besar juga waktu eksekusi yang dibutuhkan. Sedangkan untuk tingkat *Sparsity* pada Tabel 6.4 dapat disimpulkan semakin tinggi *Sparsity* maka waktu eksekusi yang dibutuhkan semakin kecil.

7.2 Saran

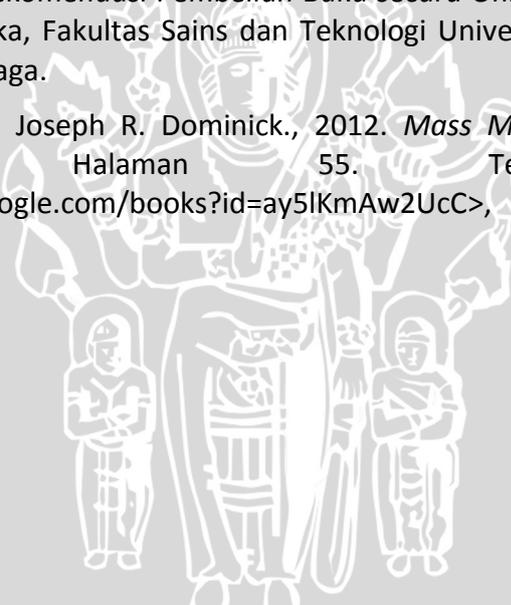
Adapun saran-saran yang diajukan oleh penulis untuk sistem rekomendasi ini adalah:

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.

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*. Tersedia di: <<http://books.google.co.id/books?id=kaza3qSULoC&dq=Likert+and+Guttm+an+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.
- Khoir, Daimul., 2014. *Informasi Perpustakaan Universitas Brawijaya*. Tersedia di: <lib.ub.ac.id/jaminan-mutu/informasi-perpustakaan-universitas-brawijaya/>, [Diakses 04 Januari 2017].
- Fitri, Aditya., 2015. *Sistem rekomendasi mata kuliah pilihan mahasiswa dengan content-based filtering dan collaborative filtering* (studi kasus: universitas brawijaya). Program Studi Informatika/Ilmu 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.
- Kriswijayanti, Ratna., 2009. *Menumbuhkan Minat Baca Pada Mahasiswa*. Tersedia di: <nadzkuraka.blogspot.co.id/2010/01/menumbuhkan-minat-baca-pada-mahasiswa.html?m=1/>, [Diakses 04 Januari 2017].
- Kwon, H. J. (2009). Improved Memory-Based Collaborative Filtering Using Entropi-Based Similiarity Measures. *2009 International Symposium on Web Information Systems and Applications*.
- 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*. Departement 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. Tersedia di: <http://books.google.com/books?id=ay5IKmAw2UcC>, [Diakses 14 Februari 2016].



LAMPIRAN A SKRIP PROGRAM

A.1 Skrip Program Perhitungan pada Sistem Rekomendasi

```
Using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;

namespace SistemRekomendasi
{
    public partial class FormDetailBuku : Form
    {
        string jurusan;
        DataView dvdiketahui;
        DataView dvdicari;
        DataView dvhasil;
        DataTable dtdicaridomain;
        DataTable dtdiketauidomain;
        DataTable dtdicari;
        DataTable dtdiketahui;
        DataTable dtrata;
        DataTable workTable = new DataTable();
        public static string idpeminjam = "";
        public static string rekombuku = "";
        public static string rowAsString = "";
        Stopwatch stopWatch = new Stopwatch();
        string idbk;
        string idbk2;
        ClassKoneksi koneksi = new ClassKoneksi();
        FormTunggu ft = null;

        public FormDetailBuku()
        {
            System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls
= false;
            InitializeComponent();
        }

        void rekomendasibuku()
        {
            workTable.Clear();
            dvdiketahui = null;
            dvdicari = null;
            dvhasil = null;
            dtdicaridomain = null;
            dtdiketauidomain = null;
            dtdicari = null;
            dtdiketahui = null;
            dtrata = null;
            var koneksi = new ClassKoneksi();
            jurusan = "";
        }
    }
}
```

```

////////// Pembentukan Matrik dan filtering
jurusan//////////
    DataSet dus = new DataSet();
    //// buku yang pernah di pinjam, sebagai domain buku yang diketahui
    dtdiketahuiDomain = koneksi.Ctable("Select
id_buku,id_mahasiswa,rating from trating where trating.id_mahasiswa = '"
+ FormMain.idmhscatalog + "' AND rating = 1");
    var rowAsString = string.Join(", ",
dtdiketahuiDomain.Rows.Cast<DataRow>().Select(r =>
r.ItemArray[0].ToString()));
    //// buku yang akan di cari, belum pernah di pinjam
    dtdicariDomain = koneksi.Ctable("Select trating.id_buku from
trating where trating.id_buku NOT IN (" + rowAsString + ") Group by
trating.id_buku Order by trating.id_buku");
    //// jurusan = nim
    dus = koneksi.getRows("select jurusan from tmahasiswa where
id_mahasiswa = '" + FormMain.idmhscatalog + "'");
    jurusan = Convert.ToString(dus.Tables["hasil"].Rows[0][0]);
    //// ini detail buku yang diketahui, memuat nilai rating nya yang
    digunakan untuk penghitungan
    dtdiketahui = koneksi.Ctable("Select
trating.id_buku,trating.id_mahasiswa,trating.rating from trating INNER
JOIN tmahasiswa ON trating.id_mahasiswa = tmahasiswa.id_mahasiswa where
trating.id_mahasiswa <> '" + FormMain.idmhscatalog + "' AND
tmahasiswa.jurusan = '" + jurusan + "' Order by trating.id_buku,
trating.id_mahasiswa");
    //// ini detail buku yang dicari, memuat nilai rating nya yang digunakan
    untuk penghitungan (nilainya sama, yaitu memuat semua buku, nanti akan
    difilter perdomain dalam menampilkan
    //// data dtdicari dan dtdiketahui sama, yaitu semua data pada tabel
    rating yang telah difilter per jurusan
    dtdicari = dtdiketahui;
    //// nilai rata rata
    dtrata = koneksi.Ctable("Select trating.id_mahasiswa,
SUM(IFNULL(trating.rating,0))/count(trating.rating) as rata2 from trating
INNER JOIN tmahasiswa ON trating.id_mahasiswa = tmahasiswa.id_mahasiswa
where trating.id_mahasiswa <> '" + FormMain.idmhscatalog + "' AND
tmahasiswa.jurusan = '" + jurusan + "' Group By id_mahasiswa");
    ////// dibuat dataview pada dtdiketahui dan dtdicari untuk melakukan
    filtering
    dvdiketahui = new DataView(dtdiketahui);
    dvdicari = new DataView(dtdicari);
    //////////// Akhir Pembentukan Matrik
    //////////// Membuat wadah untuk menampung hasil perhitungan dan prediksi
    if (workTable.Columns.Count == 0)
    {
        workTable.Columns.Add("bukuA", typeof(String));
        workTable.Columns.Add("bukuB", typeof(String));
        workTable.Columns.Add("mirip", typeof(Double));
        workTable.Columns.Add("n_rating", typeof(Double));
        workTable.Columns.Add("prediksi", typeof(Double));
    }
    //////////// dilakukan looping terhadap buku yang akan di cari
    for (int i = 0; i <= dtdicariDomain.Rows.Count - 1; i++)
    {
        //// inisialisasi idbuku pada dtdicariDomain saat dilakukan looping pada
        row ke i
        var idbuku2 =
Convert.ToString(dtdicariDomain.Rows[i][0]);

```

```

        idbk2 = idbuku2;
    /// ini untuk memfilter buku yang ada pada dvdicari untuk menampilkan
    data buku yang memuat idbuku2 (idbuku2 adalah idbuku yang belum dipinjam)
        dvdicari.RowFilter = "id_buku = " + idbuku2;
    /// Dilakukan lagi looping terhadap buku yang telah diketahui
        for (int j = 0; j <= dtdiketahuidomain.Rows.Count - 1;
j++)
        {
    /// inisialisasi idbuku pada dtdiketahuidomain saat looping row ke j
            var idbuku =
Convert.ToString(dtdiketahuidomain.Rows[j][0]);
            idbk = idbuku;
    /// ini untuk memfilter buku yang ada pada dvdiketahui untuk
    menampilkan data buku yang memuat idbuku (idbuku adalah idbuku yang
    pernah dipinjam)
                dvdiketahui.RowFilter = "id_buku = " + idbuku;
                double hasil = 0;
                double hasilpangkat = 0;
                double hasilpangkat2 = 0;
                double hasilakar = 0;
                double hasilakar2 = 0;
                double hasilakhir = 0;
    /// dilakukan looping pada data dvdiketahui yang telah difilter
                for (int k = 0; k <= dvdiketahui.Count - 1; k++)
                {
    /// nilai rating data buku yang diketahui pada row ke k
                    var rt = dvdiketahui[k][2];
    /// nilai rating data buku yang dicari pada row ke k
                    var rt2 = dvdicari[k][2];
    /// nilai rata2
                    var rata = dtrata.Rows[k][1];
    // Menghitung kemiripan
                    hasil = hasil +
((Math.Round(Convert.ToDouble(rt2) - Convert.ToDouble(rata), 3)) *
Math.Round((Convert.ToDouble(rt) - Convert.ToDouble(rata)), 3));
                    hasilpangkat2 = hasilpangkat2 +
(Math.Pow(Convert.ToDouble(rt2) - Convert.ToDouble(rata), 2));
                    hasilpangkat = hasilpangkat +
(Math.Pow(Convert.ToDouble(rt) - Convert.ToDouble(rata), 2));
                }
                    hasilakar = Math.Round(Math.Sqrt(hasilpangkat), 3);
                    hasilakar2 = Math.Round(Math.Sqrt(hasilpangkat2), 3);
                    hasilakhir = Math.Round(hasil, 3) /
Math.Round((hasilakar * hasilakar2), 3);
    //akhir perhitungan similarity
    //mengisi data
                    DataRow workRow = workTable.NewRow();
                    workRow[0] = idbk2;
                    workRow[1] = idbk;
                    workRow[2] = Math.Round(hasilakhir, 3);
                    workRow[3] = dtdiketahuidomain.Rows[j][2];
    // menghitung nilai prediksi
                    workRow[4] =
Math.Round(Math.Round(Math.Round(Math.Round(hasilakhir, 3) *
Convert.ToDouble(dtdiketahuidomain.Rows[j][2]), 3) /
Math.Abs(Convert.ToDouble(dtdiketahuidomain.Rows[j][2])), 3);
                    workTable.Rows.Add(workRow);
                }
        }
    }

```

```

        dvhasil = new DataView(workTable);
        //// dilakukan filter untuk menampilkan yang nilai prediksinya > 0 saja
        dvhasil.RowFilter = "prediksi > 0";
        ////di urutkan dari yang nilai prediksi tertinggi
        dvhasil.Sort = "prediksi Desc";
    }
    private void FormDetailBuku_Load(object sender, EventArgs e)
    {
    }
    private void backgroundWorker1_DoWork(object sender,
DoWorkEventArgs e)
    {
        DataSet ds = new DataSet();
        ds = koneksi.getRows("select
judul,ISBN,subjek,creator,dewey_no,copies,volume,printing,edisi,kota_pub,
publikasi,bibliography,language,cover from tbuku where id_buku = '" +
FormMain.idbukucatalog + "'");
        if (ds.Tables["hasil"].Rows.Count > 0)
        {
            txtjudul.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["judul"]);
            txtISBN.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["ISBN"]);
            txtsubjek.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["subjek"]);
            txtpengarang.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["creator"]);
            txtDewey.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["dewey_no"]);
            txtcopi.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["copies"]);
            txtvolume.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["volume"]);
            txtprinting.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["printing"]);
            txtedisi.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["edisi"]);
            txtkotapublikasi.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["kota_pub"]);
            txtpublikasi.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["publikasi"]);
            txtbibliography.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["bibliography"]);
            txtbahasa.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["language"]);
            pictureBoxcover.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["cover"]);
            rekomendasibuku();
        }
    }
    private void backgroundWorker1_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)
    {
        DataTable dt = new DataTable();
        dt = dvhasil.ToTable();
        FormMain fm = new FormMain();
        var rowAsString = string.Join(", ",
dt.Rows.Cast<DataRow>().Select(r => r.ItemArray[0].ToString()));

```

```

        if (rowAsString != "")
        {
            koneksi.showToGrid("select id_buku, judul, subjek,
creator, dewey_no, copies from tbuku where id_buku IN (" + rowAsString +
") order by field(id_buku," + rowAsString + ")", dataGridView1);
            dataGridView1.Focus();
        }
    }
    private void FormDetailBuku_FormClosing(object sender,
FormClosingEventArgs e)
    {
        if (backgroundWorker1.IsBusy == true)
        {
            backgroundWorker1.CancelAsync();
        }
    }
    private void button1_Click(object sender, EventArgs e)
    {
        FormAnalisaPerhitungan_Copy fac = new
FormAnalisaPerhitungan_Copy();
        fac.Show();
        this.Close();
    }
    private void FormDetailBuku_Shown(object sender, EventArgs e)
    {
        labelnim.Text = FormMain.idmhscatalog;
        labelnama.Text = FormMain.namamhscatalog;
        backgroundWorker1.RunWorkerAsync();
    }
    private void dataGridView1_SelectionChanged(object sender,
EventArgs e)
    {
    }
}
}

```

A.2 Skrip Program Pencarian Katalog di Menu Utama

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace SistemRekomendasi
{
    public partial class FormMain : Form
    {
        public static String userlogin = "";
        public static FormMain obj;
        public string login;
        public static string idbukucatalog = "";
        public static string idmhscatalog = "";
    }
}

```

```

public static string namamhscatalog = "";
ClassKoneksi koneksi = new ClassKoneksi();
public FormMain()
{
    obj = this;
    InitializeComponent();
}
public void fillrekomen(DataTable dt)
{
}
private void FormMain_Load(object sender, EventArgs e)
{
}
private void toolStripEntriDataPeminjaman_Click_1(object sender,
EventArgs e)
{
    FormPeminjaman fp = new FormPeminjaman();
    fp.Show();
}

private void toolStripImportData_Click(object sender, EventArgs
e)
{
    FormImportData im = new FormImportData();
    im.Show();
}
private void toolStripDataMahasiswa_Click_1(object sender,
EventArgs e)
{
    FormDataMahasiswa mhs = new FormDataMahasiswa();
    mhs.Show();
}

private void toolStripDataBuku_Click_1(object sender, EventArgs
e)
{
    FormDataBuku fb = new FormDataBuku();
    fb.Show();
}

private void toolStripButton2_Click(object sender, EventArgs e)
{
    UseWaitCursor = true;
    DataSet ds = new DataSet();
    ds = koneksi.getRows("select id_mahasiswa from tmahasiswa
where id_mahasiswa = '" + textBox1.Text + "'");
    if (ds.Tables["hasil"].Rows.Count > 0)
    {
        ds = koneksi.getRows("select count(id_mahasiswa) from
trating where id_mahasiswa = '" + textBox1.Text + "'");
        if (Convert.ToString(ds.Tables["hasil"].Rows[0][0]) !=
"0")
        {
            idmhscatalog = textBox1.Text;
            namamhscatalog = textnama.Text;
            FormRekomendasi fr = new FormRekomendasi();
            fr.Show();
        }
    }
    else

```

```

        {
            MessageBox.Show("Anda belum melakukan pencarian buku
silahkan melihat katalog untuk mencari buku");
        }
    }
    else
    {
        MessageBox.Show("Apabila anda ingin melihat rekomendasi
buku, Masukan NIM Terlebih dahulu");
    }
    UseWaitCursor = false;
}

private void toolStripLogOut_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Apakah Anda Yakin Ingin Keluar ???",
"Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
        this.Close();
    FormPeminjaman fp = new FormPeminjaman();
    fp.Close();
    FormMain m = new FormMain();
    m.Close();
    FormDataBuku fb = new FormDataBuku();
    fb.Close();
    FormDataMahasiswa mhs = new FormDataMahasiswa();
    mhs.Close();
    FormImportData imp = new FormImportData();
    imp.Close();
}

private void buttoncari_Click(object sender, EventArgs e)
{
    if (textBox1.Text != "")
    {
        if (comboBoxtahun.Text == "" || comboBoxtahun.Text ==
"Semua Tahun")
        {
            if (comboBoxberdasarkan.Text == "Judul")
            {
                koneksi.showToGrid("select
id_Buku,judul,edisi,publikasi,date_format(tahun_pub,'%d %m %Y') as
tahun_pub,creator,subjek,ISBN from tbuku where judul LIKE '%" +
textBoxkatakunci.Text + "%' Order By id_buku", dataGridView1);
            }
            else if (comboBoxberdasarkan.Text == "Pengarang")
            {
                koneksi.showToGrid("select
id_Buku,judul,edisi,publikasi,date_format(tahun_pub,'%d %m %Y') as
tahun_pub,creator,subjek,ISBN from tbuku where creator LIKE '%" +
textBoxkatakunci.Text + "%' Order By id_buku", dataGridView1);
            }
            else if (comboBoxberdasarkan.Text == "Penerbit")
            {
                koneksi.showToGrid("select
id_Buku,judul,edisi,publikasi,date_format(tahun_pub,'%d %m %Y') as
tahun_pub,creator,subjek,ISBN from tbuku where publikasi LIKE '%" +
textBoxkatakunci.Text + "%' Order By id_buku", dataGridView1);
            }
        }
    }
}

```

```

else if (comboBoxberdasarkan.Text == "ISBN")
{
    koneksi.showToGrid("select
id_Buku,judul,edisi,publikasi,date_format(tahun_pub,'%d %m %Y') as
tahun_pub,creator,subjek,ISBN from tbuku where ISBN LIKE '%" +
textBoxkatakunci.Text + "%' Order By id_buku", dataGridView1);
}
}
else
{
    if (comboBoxberdasarkan.Text == "Judul")
    {
        koneksi.showToGrid("select
id_Buku,judul,edisi,publikasi,date_format(tahun_pub,'%d %m %Y') as
tahun_pub,creator,subjek,ISBN from tbuku where judul LIKE '%" +
textBoxkatakunci.Text + "%' AND tahun_pub = '%" + comboBoxtahun.Text +
"%'" Order By id_buku", dataGridView1);
    }
    else if (comboBoxberdasarkan.Text == "Pengarang")
    {
        koneksi.showToGrid("select
id_Buku,judul,edisi,publikasi,date_format(tahun_pub,'%d %m %Y') as
tahun_pub,creator,subjek,ISBN from tbuku where creator LIKE '%" +
textBoxkatakunci.Text + "%' AND tahun_pub = '%" + comboBoxtahun.Text +
"%'" Order By id_buku", dataGridView1);
    }
    else if (comboBoxberdasarkan.Text == "Penerbit")
    {
        koneksi.showToGrid("select
id_Buku,judul,edisi,publikasi,date_format(tahun_pub,'%d %m %Y') as
tahun_pub,creator,subjek,ISBN from tbuku where publikasi LIKE '%" +
textBoxkatakunci.Text + "%' AND tahun_pub = '%" + comboBoxtahun.Text +
"%'" Order By id_buku", dataGridView1);
    }
    else if (comboBoxberdasarkan.Text == "ISBN")
    {
        koneksi.showToGrid("select
id_Buku,judul,edisi,publikasi,date_format(tahun_pub,'%d %m %Y') as
tahun_pub,creator,subjek,ISBN from tbuku where ISBN LIKE '%" +
textBoxkatakunci.Text + "%' AND tahun_pub = '%" + comboBoxtahun.Text +
"%'" Order By id_buku", dataGridView1);
    }
}
dataGridView1.Focus();
}
else
{
    MessageBox.Show("Apabila ingin mencari buku Masukkan NIM
Terlebih dahulu");
}
}

private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    var kon = new ClassKoneksi();
    if (e.KeyCode == Keys.Enter)
    {
        DataSet ds = new DataSet();
        ds = kon.getRows("Select nama_mahasiswa,fakultas,jurusan

```

```

from tmahasiswa where id_mahasiswa = '" + textBox1.Text + "'";
        if (ds.Tables["hasil"].Rows.Count > 0)
        {
            textnama.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["nama_mahasiswa"]);
            txtjurusan.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["jurusan"]);
            label1.Text =
Convert.ToString(ds.Tables["hasil"].Rows[0]["fakultas"]);
        }
    }

    private void dataGridView1_CellContentDoubleClick(object sender,
DataGridViewCellEventArgs e)
    {
        DataSet ds = new DataSet();
        ds = koneksi.getRows("select id_mahasiswa from tmahasiswa
where id_mahasiswa = '" + textBox1.Text + "'");
        if (ds.Tables["hasil"].Rows.Count > 0)
        {
            if (idmhscatalog != "" && idmhscatalog != textBox1.Text)
            {
                ds.Dispose();
                ds = koneksi.getRows("Select count(id_mahasiswa) from
trating where id_mahasiswa = '" + idmhscatalog + "' AND jenis = 0");
                if (Convert.ToInt32(ds.Tables["hasil"].Rows[0][0]) > 1)
                {
                    koneksi.exeNonQuery("update trating SET rating = 0
WHERE jenis = 1 AND id_mahasiswa = '" + idmhscatalog + "'");
                    koneksi.exeNonQuery("update trating SET jenis = 0
WHERE id_mahasiswa = '" + idmhscatalog + "'");
                }
                else if (Convert.ToInt32(ds.Tables["hasil"].Rows[0][0])
== 0)
                {
                    koneksi.exeNonQuery("delete from trating where
id_mahasiswa = '" + idmhscatalog + "' AND jenis = 1");
                }
                ds.Dispose();
                idbukucatalog =
Convert.ToString(dataGridView1.SelectedCells[0].Value);
                idmhscatalog = textBox1.Text;
                namamhscatalog = textnama.Text;
                ds = koneksi.getRows("select rating from trating where
id_mahasiswa = '" + textBox1.Text + "' AND id_buku = '" + idbukucatalog +
"'");
                FormDetailBuku fdb = new FormDetailBuku();
                //fdb.LogicalParent(this);
                if (ds.Tables["hasil"].Rows.Count > 0)
                {
                    if (Convert.ToString(ds.Tables["hasil"].Rows[0][0])
== "0")
                    {
                        koneksi.exeNonQuery("update trating SET rating =
1, jenis = 1 WHERE id_buku = '" + idbukucatalog + "' AND id_mahasiswa = '"
+ textBox1.Text + "'");
                    }
                }
            }
        }
    }

```

```

    }
    fdb.ShowDialog();
    ds.Dispose();

    }
    else
    {
        koneksi.exeNonQuery("Insert Ignore into trating
        (Select tbuku.id_buku, tmahasiswa.id_mahasiswa, '0' as rating, '1' as
        jenis from tbuku,tmahasiswa where tmahasiswa.id_mahasiswa = '" +
        textBox1.Text + "'");
        //idbukucatalog =
        Convert.ToString(dataGridView1.SelectedCells[0].Value);
        //idmhscatalog = textBox1.Text;
        koneksi.exeNonQuery("update trating SET rating = 1
        WHERE id_buku ='" + idbukucatalog + "' AND id_mahasiswa ='" +
        textBox1.Text + "'");
        fdb.ShowDialog();
    }
    //idbukucatalog =
    Convert.ToString(dataGridView1.SelectedCells[0].Value);
    //idmhscatalog = textBox1.Text;
    //FormDetailBuku fdb = new FormDetailBuku();
    //fdb.Show();
}
else
{
    MessageBox.Show("Masukan NIM Terlebih Dahulu");
}
}

private void toolStripButton1_Click(object sender, EventArgs e)
{
    FormLogin fr = new FormLogin();
    fr.Show();
}
private void toolStripButtonkeluar_Click(object sender, EventArgs
e)
{
    if (MessageBox.Show("Apakah Anda Yakin Ingin Keluar ???",
    "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
    DialogResult.Yes)
        this.Close();
    FormPeminjaman fp = new FormPeminjaman();
    fp.Close();
    FormMain m = new FormMain();
    m.Close();
    FormDataBuku fb = new FormDataBuku();
    fb.Close();
    FormDataMahasiswa mhs = new FormDataMahasiswa();
    mhs.Close();
    FormImportData imp = new FormImportData();
    imp.Close();
}
}
}
}

```

LAMPIRAN B SURAT PENELITIAN

B.1 Surat Pengantar Penelitian di Perpustakaan Universitas Brawijaya



KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
FAKULTAS ILMU KOMPUTER

Gedung A FILKOM Lt. 1, Jl. Veteran No.8, Malang, 65145, Indonesia
Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : ptiik@ub.ac.id

Nomor : 3008 /UN10.36/AK/2016
Perihal : *Permohonan data skripsi*

19 APR 2016

Yth. Kepala Perpustakaan UB
Drs. Johan A.E. Noor, MSc., Ph.D.
Jl. Veteran No. 8
Malang

Untuk mendukung penyelesaian skripsi mahasiswa berikut :

Nama : Rahman Anam
NIM : 135150109111003
Judul Skripsi : Sistem Rekomendasi Peminjaman Buku di Perpustakaan Menggunakan Metode Item-based & User-based Collaborative Filtering
Dosen Pembimbing : 1. Rekyan Regasari MP, ST., MT.
2. Agus Wahyu Widodo, S.T., M.Cs.
Prodi : Informatika / Ilmu Komputer

Guna melakukan survey atau pengambilan data untuk skripsi mahasiswa tersebut di instansi saudara, jenis survey atau data yang diperlukan dan rencana waktu pelaksanaan adalah :

Data : Wawancara dan pengumpulan data berupa :
1. Data transaksi peminjaman buku
2. Data pencarian buku
3. Data buku perpustakaan UB
Waktu : 18 April 2016 - 18 Juli 2016

Atas perhatian dan kerjasamanya kami ucapkan terimakasih.

Mengetahui,
a.n. Dekan
Kepala Tata Usaha Fakultas Ilmu Komputer,



Dra. Jang Budha Mahadyawati, M.A.P.
NIK/ NIP. 2011027704142001

Dosen Pembimbing I

Rekyan Regasari MP, ST., MT.
NIK/ NIP. 2011027704142001

Tembusan Kepada Yth:

1. Ketua Program Studi Informatika / Ilmu Komputer
2. Mahasiswa yang bersangkutan



B.2 Surat Balasan Perijinan Penelitian di Perpustakaan Universitas Brawijaya



**KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
PERPUSTAKAAN**

Gedung Perpustakaan, Jl. Veteran Malang 65145, Indonesia
Telp. : + 62-341-551611, 575777; Pes. 308; Fax. Rektorat + 62-341-565420
Langsung : + 62-341-571032 Fax. +62-341-583966
http://www.lib.ub.ac.id e-mail: library@ub.ac.id

Nomor : /457UN.10.22/UM/2016
Hal. : Permohonan Data Skripsi

Yth.
Dekan Fakultas Ilmu Komputer
Prodi Informatika
Universitas Brawijaya

Membalas surat Kepala Tata Usaha, an. Dekan FIK Universitas Brawijaya, Nomor : 3008/UN10.36/AK/2016 dan 3022/UN10.36/AK/2016 hal: Permohonan Data Skripsi, di Perpustakaan Universitas Brawijaya, Yaitu a.n.:

No	Nama	NIM	Waktu	Data	Judul Skripsi
1	Rahman Anam	13515010 9111003	18-04-2016 s/d 18-07-2016	Data transaksi peminjaman buku, data pendacarian buku dan data buku perpus.UB	Sistem rekomendasi peminjaman buku menggunakan metode Item-based & user-based Colaborative Filtering,
2	Fanandi P.R.	12515020 0111130	21-04-2016 s/d 21-07-2016	Data transaksi peminjaman buku, data pendacarian buku dan data buku perpus.UB	Sistem rekomendasi peminjaman buku menggunakan metode content based Filtering,

Bersama ini kami memberikan ijin jadwal tersebut diatas.
Atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Kepala,



Johan Andoyosenteng Noor,
NIP. 19650325 199002 1 004

LAMPIRAN C DATA DAN HASIL PENGUJIAN

C.1 Lampiran Data Transaksi Peminjaman

NIM	Kode Buku	Judul
125150207111019	011305614	Algoritma & pemrograman menggunakan Java
125150207111019	010309363	Java handbook : konsep dasar pemrograman Java
125150207111019	011306739	Mobile web programming : HTML5, CSS3, jQuery mobile
125150207111011	011305614	Algoritma & pemrograman menggunakan Java
125150207111011	011305750	Buku pintar robotika : bagaimana merancang dan membuat robot sendiri
125150207111011	010401408	Pemrograman C++
125150207111011	019800881	Pengenalan komputer
125150200111131	010403717	Small Business Solutions for Networking = Membuat Jaringan Komputer untuk Perusahaan Kecil
125150200111131	011614803	Membangun sistem jaringan komputer
125150200111130	011305614	Algoritma & pemrograman menggunakan Java
125150200111130	010309363	Java handbook : konsep dasar pemrograman Java
125150200111130	019800551	Dasar-dasar pemrograman Pascal : teori program
125150200111130	011306115	Organisasi dan arsitektur komputer
125150200111124	011305614	Algoritma & pemrograman menggunakan Java
125150200111124	010400985	Introduction to web design programming for e-business and e-commerce
125150200111124	011305614	Algoritma & pemrograman menggunakan Java
125150200111124	011306771	Belajar Java secara visual dan interaktif
125150200111120	010700856	Java Software solutions : foundations of program design
125150200111119	011614803	Membangun sistem jaringan komputer untuk pemula
125150200111119	010403717	Small Business Solutions for Networking = Membuat Jaringan Komputer untuk Perusahaan Kecil
125150200111119	010306280	Data abstraction and problem solving with Java : walls and mirrors
125150200111117	010400985	Introduction to web design programming for e-business and e-commerce
125150200111117	011306771	Belajar Java secara visual dan interaktif
125150200111117	011401013	Pro Web project management
125150200111116	010401408	Pemrograman C++
125150200111116	019800881	Pengenalan komputer
125150200111105	011305614	Algoritma & pemrograman menggunakan Java
125150200111105	010309363	Java handbook : konsep dasar pemrograman Java
125150200111105	010400985	Introduction to web design programming for e-business and e-commerce

125150200111105	019800881	Pengenalan komputer
125150200111105	011614803	Membangun sistem jaringan komputer untuk pemula

C.2 Lampiran Data Hasil Pengujian Relevansi dengan DDC dengan 30% Sparsity Data

NIM	Kode Buku yang Dipilih	Kode Buku Yang Direkomendasikan	Nomer DDC Buku yang dipilih	Nomer DDC Buku yang direkomendasikan
125150200111116	019800881	11305614	004	005
125150200111116		11305750		629
125150200111116		10309363		005
125150200111116		10400985		005
125150200111116		11614803		004
125150200111119	011614803	10400985	004	005
125150200111119	010306280	10309363	005	005
125150200111119		19800881		004
125150200111119		11305614		005
125150207111011	011305614	10309363	004	005
125150207111011	011305750	11306115	629	004
125150207111011	019800881	11306739	004	006
125150207111011		10400985		005
125150207111011		10309363		005
125150207111011		11614803		004
125150207111011		10400985		005
125150207111011		11614803		004
125150200111105	011305614	11306115	004	004
125150200111105	010309363	11306771	005	005
125150200111105	010400985	10306280	005	005
125150200111105	019800881	11305750	004	629
125150200111105	011614803	11305750	004	629
125150200111105		11306115		004
125150200111105		11306739		006
125150200111130	011305614	10400985	005	005
125150200111130	010309363	19800881	005	004
125150200111130	011306115	11614803	004	004
125150200111130		19800881		004
125150200111130		11305750		629

125150200111130		11306739		006
125150200111130		10400985		005
125150200111130		11614803		004

C.3 Lampiran Data Hasil Pengujian MAE dengan 30% Sparsity Data

NIM	Kode buku	Penilaian	Penilaian Prediksi Sistem	ERROR MAE
125150200111116	11305614	1	0,704	0,296
125150200111116	11305750	1	0,704	0,296
125150200111116	10309363	1	0,495	0,505
125150200111116	10400985	1	0,495	0,505
125150200111116	11614803	1	0,403	0,597
125150200111119	10400985	1	0,496	0,504
125150200111119	10309363	1	0,495	0,505
125150200111119	19800881	1	0,403	0,597
125150200111119	11305614	1	0,347	0,653
125150207111011	10309363	1	0,815	0,185
125150207111011	11306115	1	0,572	0,428
125150207111011	11306739	1	0,572	0,428
125150207111011	10400985	1	0,496	0,504
125150207111011	10309363	1	0,495	0,505
125150207111011	11614803	1	0,404	0,596
125150207111011	10400985	1	0,402	0,598
125150207111011	11614803	1	0,328	0,672
125150200111105	11306115	1	1	0
125150200111105	11306771	1	1	0
125150200111105	10306280	1	0,706	0,294
125150200111105	11305750	1	0,706	0,294
125150200111105	11305750	1	0,573	0,427
125150200111105	11306115	1	0,573	0,427
125150200111105	11306739	1	0,573	0,427
125150200111130	10400985	1	0,704	0,296
125150200111130	19800881	1	0,664	0,336
125150200111130	11614803	1	0,573	0,427
125150200111130	19800881	1	0,573	0,427
125150200111130	11305750	1	0,572	0,428
125150200111130	11306739	1	0,572	0,428
125150200111130	10400985	1	0,402	0,598
125150200111130	11614803	1	0,328	0,672

C.4 Lampiran Data Hasil Pengujian MAE dengan 50% Sparsity Data

NIM	Kodebuku	Penilaian	Penilaian Prediksi WS	ERROR MAE
125150200111116	11614803	1	0,575	0,425
125150200111116	11305614	1	0,574	0,426
125150200111119	19800881	1	0,498	0,502
125150200111119	11305614	1	0,405	0,595
125150207111011		1	0	1
125150200111105	11306115	1	0,705	0,295
125150200111105	11306739	1	0,705	0,295
125150200111130	11306739	1	0,705	0,295
125150200111130	19800881	1	0,497	0,503
125150200111130	11614803	1	0,405	0,595

C.5 Lampiran Data Hasil Pengujian MAE dengan 60% Sparsity Data

NIM	Kodebuku	Penilaian	Penilaian Prediksi WS	ERROR MAE
125150200111116	-	1	0	1
125150200111119	11305614	1	0,406	0,594
125150207111011	-	1	0	1
125150200111105	-	1	0	1
125150200111130	11614803	1	0,406	0,594