

**SISTEM REKOMENDASI MATA KULIAH PILIHAN
MAHASISWA DENGAN CONTENT-BASED FILTERING
DAN COLLABORATIVE FILTERING (STUDI
KASUS: UNIVERSITAS BRAWIJAYA)**

SAMPUL

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh:

ADITYA FITRI HANANTA PUTRA

NIM. 105060804111008

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

LEMBAR PERSETUJUAN

**SISTEM REKOMENDASI MATA KULIAH PILIHAN
MAHASISWA DENGAN CONTENT-BASED FILTERING
DAN COLLABORATIVE FILTERING (STUDI
KASUS: UNIVERSITAS BRAWIJAYA)**

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUAL



Disusun Oleh:

ADITYA FITRI HANANTA PUTRA

NIM. 105060804111008

Skripsi ini telah disetujui oleh dosen pembimbing pada tanggal 3 Juni 2015:

Pembimbing I

Pembimbing II

Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D.
NIP. 19720919 199702 1 001

Budi Darma Setiawan, S.Kom., M.Cs.
NIP. 19841015 201404 1 002

LEMBAR PENGESAHAN

**SISTEM REKOMENDASI MATA KULIAH PILIHAN
MAHASISWA DENGAN CONTENT-BASED FILTERING
DAN COLLABORATIVE FILTERING (STUDI
KASUS: UNIVERSITAS BRAWIJAYA)**

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh:

ADITYA FITRI HANANTA PUTRA

NIM. 105060804111008

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 11 Juni 2015

Penguji I

Penguji II

Drs. Marji, M.T

NIP. 19670801 199203 1 001

Imam Cholissodin, S.Si, M.Kom

NIK. 85071916110422

Penguji III

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

NIK. 77041406120253

Mengetahui,

Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T

NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

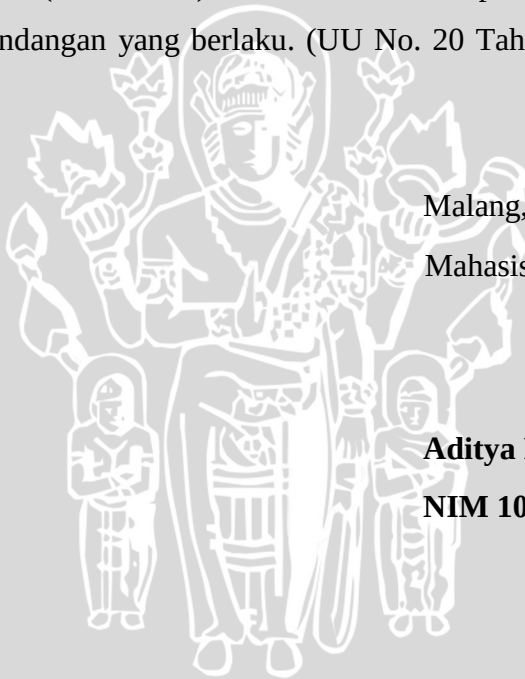
Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat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

Malang, Mei 2015

Mahasiswa,

Aditya Fitri Hananta P.

NIM 105060804111008



KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT, karena atas limpahan rahmat, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini. Tidak lupa shalawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW beserta keluarga dan sahabat Beliau.

Skripsi ini merupakan serangkaian tugas akhir penulis selama mengikuti kegiatan kuliah dan sebagai salah satu syarat untuk mencapai gelar Sarjana Komputer di Fakultas Ilmu Komputer, Universitas Brawijaya. Judul skripsi ini adalah **“Sistem Rekomendasi Mata Kuliah Pilihan Mahasiswa Dengan Content-Based Filtering Dan Collaborative Filtering (Studi Kasus: Universitas Brawijaya)”**.

Penulis menyadari bahwa skripsi ini tidak dapat terselesaikan dengan baik tanpa keterlibatan dari berbagai pihak, karena itu penulis menyampaikan ucapan terima kasih kepada:

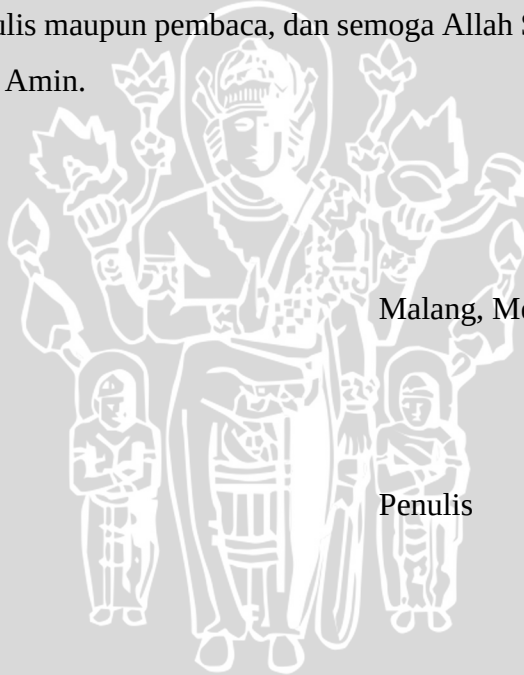
1. Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. dan Budi Darma Setiawan, S.Kom., M.Cs. selaku pembimbing skripsi yang telah membantu dan membimbing penulis dari penyusunan proposal sampai skripsi ini selesai.
2. Eriq Muh. Adams Jonemaro, S.T, M.Kom. selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
3. Drs. Mardji, M.T selaku Ketua Program Teknik Informatika / Ilmu Komputer yang telah memberikan kesempatan kepada penulis untuk menyelesaikan skripsi ini.
4. Orang tua penulis, saudara penulis dan keluarga penulis, Ibu Binti Salamah dan Bapak Yuli Santoso Prihantara, Yusuf Putra Al-Azhar, Lintang Fajrin Arifa, atas segala keikhlasan dan pengorbanannya, mendoakan penulis untuk menyelesaikan skripsi ini.
5. Sahabat penulis, Nurya Aghnia Farda yang selalu memberikan bantuan, dukungan dan memotivasi penulis hingga penyusunan skripsi ini selesai.

6. Segenap bapak dan ibu dosen program studi Teknik Informatika / Ilmu Komputer beserta staff administrasi yang telah membantu penulis selama masa studi.
7. Teman-teman Jurusan Informatika, khususnya Fariz Izzan, Shindu Nata Gama, Hermansyah, terimakasih atas kebersamaan serta motivasi yang telah diberikan.
8. Rekan-rekan beserta pimpinan Unit TIK Universitas Brawijaya yang telah memberikan motivasi serta bantuan dalam menyelesaikan skripsi ini

Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan jauh dari sempurna, karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Maka, saran dan kritik yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Semoga skripsi ini dapat bermanfaat dan berguna bagi semua pihak, baik penulis maupun pembaca, dan semoga Allah SWT meridhoi dan dicatat sebagai ibadah. Amin.

Malang, Mei 2015

Penulis



ABSTRAK

Aditya Fitri Hananta Putra. 2015. : Sistem Rekomendasi Mata Kuliah Pilihan Mahasiswa Dengan Content-Based Filtering Dan Collaborative Filtering (Studi Kasus: Universitas Brawijaya). Skripsi Program Studi Teknik Informatika/Illmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Pembimbing : Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D dan Budi Darma Setiawan, S.Kom., M.Cs.

Universitas Brawijaya merupakan perguruan tinggi di Indonesia yang memiliki lebih dari 50.000 mahasiswa aktif yang tersebar dalam 143 program studi. Setiap awal semester, setiap mahasiswa akan melakukan registrasi akademik dengan memilih dan memprogram beberapa mata kuliah yang akan ditempuh pada semester tersebut. Permasalahan yang dialami oleh mahasiswa adalah kesulitan untuk memilih mata kuliah yang akan ditempuh karena banyaknya pilihan mata kuliah yang ditawarkan. Solusi yang dapat diberikan adalah dengan membuat sistem rekomendasi pemilihan mata kuliah yang dapat merekomendasikan mata kuliah sesuai dengan minat mahasiswa tersebut. Sistem rekomendasi yang dibangun pada penelitian ini menggunakan metode *content-based filtering*, *item-based collaborative filtering*, dan *user-based collaborative filtering* untuk dapat dibandingkan antar ketiganya. Dari ketiga metode tersebut, ditemukan bahwa akurasi rekomendasi yang diberikan terbaik bernilai 67,63% untuk *content-based filtering*, 88,69% untuk *item-based collaborative filtering*, dan 85,71% untuk *user-based collaborative filtering*. Hasil akhir berupa kumpulan mata kuliah yang direkomendasikan kepada mahasiswa.

Kata Kunci : Sistem Rekomendasi, *Content-Based Filtering*, *Item-Based Collaborative Filtering*, *User-Based Collaborative Filtering*, Pemilihan Mata Kuliah.

ABSTRACT

Aditya Fitri Hananta Putra. 2015. : Course Recommender System Using Content-based Filtering and Collaborative Filtering(Case Study: Universitas Brawijaya). Final Paper, Study Program of Information Technique / Computer Science, Program of Information Technology and Computer Science, University of Brawijaya. Advisors: Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D dan Budi Darma Setiawan, S.Kom., M.Cs.

Brawijaya University is a university in Indonesia which has more than 50,000 active students from 143 study programmes. In every beginning of the semester, every student have to accomplish academic registration by choosing and enrolling on several courses which they will be studying for the semester. The problem occurred is that the students are having problem in choosing courses they'd like to program because there are so many courses available to take. The solution to address this problem is to create a course recommender system which able to recommend courses to students according to their preferences. The system which is built upon this research uses several methods, which are content-based filtering, item-based collaborative filtering, and user-based collaborative filtering hence to be compared each other. From those methods, is is found that the best accuracy values for each method are 67,63% for the content-based filtering, 88,69% for the item-based collaborative filtering, and 85,71% for the user-based collaborative filtering. The final result is a compilation of courses which are recommended for certain students.

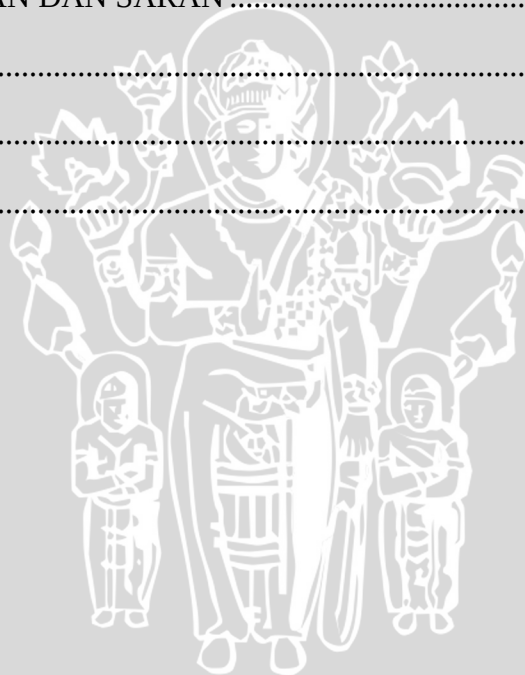
Keywords : Recommender System, Content-Based Filtering, Item-Based Collaborative Filtering, User-Based Collaborative Filtering, Course Enrollment.

DAFTAR ISI

SAMPUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI	iv
KATA PENGANTAR	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR KODE PROGRAM	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Metodologi Penelitian	4
1.7 Sistematika Penulisan	4
BAB II KAJIAN PUSTAKA DAN DASAR TEORI	6
2.1 Kajian Pustaka	6
2.2 Mekanisme Registrasi Akademik Universitas Brawijaya	8
2.3 Sistem Rekomendasi	10
2.4 Sistem Rekomendasi Pemilihan Mata Kuliah	12

2.5	Content-Based Filtering	13
2.5.1	Content-based Filtering dengan Decision Tree.....	14
2.5.2	Pembentukan Decision Tree dengan Algoritma ID3	15
2.6	Collaborative Filtering	16
2.6.1	User-based Collaborative Filtering	17
2.6.2	Item-based Collaborative Filtering	19
2.7	Sistem Rekomendasi Hibrid.....	21
2.8	Pengukuran Tingkat Akurasi.....	22
BAB III METODE PENELITIAN DAN PERANCANGAN		23
3.1	Studi Literatur	23
3.2	Data Penelitian	23
3.3	Deskripsi Umum Sistem.....	24
3.4	Perancangan Sistem.....	24
3.4.1	Content-based Filtering.....	24
3.4.2	Collaborative Filtering	37
3.5	Perancangan Antarmuka	43
3.6	Perancangan Manajemen Data	44
3.7	Perancangan Pengujian	51
BAB IV IMPLEMENTASI		54
4.1	Spesifikasi Sistem	54
4.1.1	Spesifikasi Perangkat Keras.....	54
4.1.2	Spesifikasi Perangkat Lunak.....	54
4.2	Batasan - Batasan Implementasi	55
4.3	Implementasi Algoritma.....	55
4.3.1	Implementasi <i>Content Based Filtering</i>	55
4.3.2	Implementasi <i>User-Based Collaborative Filtering</i>	66

4.3.3	Implementasi <i>Item-Based Collaborative Filtering</i>	74
4.4	Implementasi Antarmuka	86
BAB V PENGUJIAN DAN ANALISIS.....		89
5.1	Pengujian Pengaruh Jumlah Semester terhadap terhadap Tingkat Akurasi Hasil Prediksi <i>Content-Based Filtering</i>	89
5.2	Pengujian Pengaruh Jumlah Semester terhadap terhadap Tingkat Akurasi Hasil Prediksi <i>Item-Based Collaborative Filtering</i>	93
5.3	Pengujian Pengaruh Jumlah Semester terhadap terhadap Tingkat Akurasi Hasil Prediksi <i>User-Based Collaborative Filtering</i>	96
BAB VI KESIMPULAN DAN SARAN		100
6.1	Kesimpulan.....	100
6.2	Saran.....	101
Daftar Pustaka		102



DAFTAR GAMBAR

Gambar 2.1 Alur registrasi mahasiswa UB periode akademik 2014-2015 ganjil. (sumber: http://www.ub.ac.id/pengumuman/detail/registrasi-akademik-mahasiswa-universitas-brawijaya-semester-ganjil-2014-2015)	9
Gambar 2.2 Antarmuka pemrograman KRS mahasiswa pada aplikasi SIAM	10
Gambar 2.3 Arsitektur sistem rekomendasi	11
Gambar 2.4 Proses decision tree classifiers	14
Gambar 2.5 Decision tree.....	15
Gambar 2.6 Proses collaborative filtering	16
Gambar 2.7 Proses komputasi similaritas	19
Gambar 2.8 Linear combination model	21
Gambar 2.9 Sequential combination model.....	22
Gambar 3.1 Decision Tree dengan U1 Sebagai Root Utama.....	27
Gambar 3.2 Decision Tree dengan U2 Sebagai Node Cabang 0	29
Gambar 3.3 Decision Tree dengan B dan B+ Sebagai Node Cabang dari U2.....	29
Gambar 3.4 Decision Tree dengan U2 Sebagai Node Cabang 1	30
Gambar 3.5 Decision Tree dengan U3 Sebagai Node Cabang 0	32
Gambar 3.6 Decision Tree dengan U5 Sebagai Node Cabang 0	33
Gambar 3.7 Decision Tree bernilai C dan A Sebagai leaf dari U5.....	34
Gambar 3.8 Decision Tree dengan C+ Sebagai Node dari Cabang 1 Node U3 ...	35
Gambar 3.9 Decision Tree dengan A Sebagai Node dari Cabang 1 Node U2	36
Gambar 3.10 Decision Tree Final	36
Gambar 3.11 Rancangan antarmuka halaman rekomendasi mata kuliah	44
Gambar 3.12 Perancangan tabel data	45
Gambar 4.1 Tampilan antarmuka rekomendasi dengan content-based filtering ..	86
Gambar 4.2 Tampilan antarmuka rekomendasi dengan user-based collaborative filtering.....	87
Gambar 4.3 Tampilan antarmuka rekomendasi dengan item-based collaborative filtering.....	88
Gambar 5.1 Grafik tingkat akurasi hasil rekomendasi content-based filtering	91
Gambar 5.2 Grafik pengaruh jumlah semester terhadap tingkat akurasi hasil rekomendasi item-based collaborative filtering	95
Gambar 5.3 Grafik pengaruh jumlah semester terhadap tingkat akurasi hasil rekomendasi user-based collaborative filtering	98

DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian-penelitian sebelumnya tentang sistem rekomendasi mata kuliah	6
Tabel 2.2 Contoh tabel data decision tree	14
Tabel 2.3 Contoh data rating pengguna	17
Tabel 3.1 Contoh daftar mata kuliah yang diprogram mahasiswa tertentu	25
Tabel 3.2 Tabel Hasil Entropi dan Information Gain masing-masing Atribut	27
Tabel 3.3 Tabel Mata Kuliah Node U1 Cabang 0.....	28
Tabel 3.4 Tabel Mata Kuliah Node U1 Cabang 1.....	28
Tabel 3.5 Tabel Hasil Entropi dan Information Gain Atribut U2, U3, U4 dan U5 node U1 cabang 0.....	28
Tabel 3.6 Tabel Hasil Entropi dan Information Gain Atribut U2, U3, U4 dan U5 node U1 cabang 1.....	30
Tabel 3.7 Tabel Mata Kuliah Node U2 Cabang 0.....	31
Tabel 3.8 Tabel Mata Kuliah Node U2 Cabang 1.....	31
Tabel 3.9 Tabel Hasil Entropi dan Information Gain Atribut U3, U4 dan U5 node U2 cabang 0	31
Tabel 3.10 Tabel Mata Kuliah Node U3 Cabang 0.....	32
Tabel 3.11 Tabel Mata Kuliah Node U3 Cabang 1.....	32
Tabel 3.12 Tabel Hasil Entropi dan Information Gain Atribut U4 dan U5 node U3 cabang 0	33
Tabel 3.13 Contoh tabel mata kuliah yang telah diprogram oleh para mahasiswa	38
Tabel 3.14 Matriks relasi mahasiswa - nilai MK.....	39
Tabel 3.15 Tambahan pengguna aktif X yang akan diberi rekomendasi.....	39
Tabel 3.16 Mean adjusted scores, user-based collaborative filtering	40
Tabel 3.17 Mean adjusted scores, item-based collaborative filtering.....	42
Tabel 3.18 Tabel USER	46
Tabel 3.19 Tabel M_JENJANG.....	46
Tabel 3.20 Tabel M_JENJANG.....	46
Tabel 3.21 Tabel M_JURUSAN	47
Tabel 3.22 Tabel M_PROG_STUDI	47
Tabel 3.23 Tabel M_NILAI	48

Tabel 3.24 Tabel M_KOMPETENSI	48
Tabel 3.25 Tabel MHS.....	49
Tabel 3.26 Tabel MATA_KULIAH	49
Tabel 3.27 Tabel MK_KOMPETENSI.....	50
Tabel 3.28 Tabel MHS_KRS_KELAS	50
Tabel 3.29 Tabel MHS_KHS.....	51
Tabel 3.30 Contoh mata kuliah yang ditempuh mahasiswa	52
Tabel 3.31 Contoh hasil rekomendasi mata kuliah	53
Tabel 5.1 Statistik data latih dan data uji pada pengujian pengaruh jumlah semester terhadap hasil rekomendasi content-based filtering.....	90
Tabel 5.2 Statistik kecocokan hasil rekomendasi content-based filtering dengan data aktual pada semester yang berkesesuaian	90
Tabel 5.3 Statistik kecocokan hasil rekomendasi content-based filtering dengan data aktual pada semester yang tidak saling berkesesuaian	91
Tabel 5.4 Rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester uji	92
Tabel 5.5 Rata-rata standar deviasi data latih pada setiap semester uji	92
Tabel 5.6 Statistik data latih dan data uji pada pengujian pengaruh jumlah semester terhadap hasil rekomendasi item-based collaborative filtering.....	93
Tabel 5.7 Statistik kecocokan hasil rekomendasi item-based collaborative filtering dengan data aktual pada semester yang berkesesuaian.....	93
Tabel 5.8 Statistik kecocokan hasil rekomendasi item-based collaborative filtering dengan data aktual pada semester yang tidak saling berkesesuaian.....	94
Tabel 5.9 Rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester uji	96
Tabel 5.10 Statistik data latih dan data uji pada pengujian pengaruh jumlah semester terhadap hasil rekomendasi user-based filtering	96
Tabel 5.11 Statistik kecocokan hasil rekomendasi user-based collaborative filtering dengan data aktual pada semester yang berkesesuaian	97
Tabel 5.12 Statistik kecocokan hasil rekomendasi user-based collaborative filtering dengan data aktual pada semester yang tidak saling berkesesuaian.....	98

Tabel 5.13 Rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester uji

99



DAFTAR KODE PROGRAM

Kode Program 4.1 Implementasi content-based filtering	66
Kode Program 4.2 Implementasi user-based collaborative filtering.....	74
Kode Program 4.3 Implementasi item-based filtering	85



BAB I

PENDAHULUAN

1.1 Latar Belakang

Mahasiswa semester kedua dan selanjutnya biasanya bertanggung jawab terhadap pemilihan mata kuliah yang diisikan di KRS (Kartu Rencana Studi) masing-masing. Proses yang biasa disebut dengan registrasi akademik ini dilakukan di awal semester. Di Universitas Brawijaya, proses registrasi akademik dilakukan oleh mahasiswa melalui aplikasi Sistem Informasi Akademik Mahasiswa (SIAM) yang dapat diakses di <http://siam.ub.ac.id>.

Permasalahan yang biasa dialami oleh mahasiswa pada saat melakukan registrasi akademik adalah kebingungan memilih mata kuliah yang sesuai dengan minat dan kemampuannya. Banyak mahasiswa yang belum mempunyai gambaran keminatan (konsentrasi) studi untuk semester selanjutnya. Bahkan, banyak mahasiswa yang sudah mempunyai keminatan studi tertentu juga masih kebingungan memilih mata kuliah yang akan ditempuh.

Sistem rekomendasi adalah perangkat lunak dan teknik yang menyediakan saran mengenai item tertentu untuk digunakan oleh pengguna [RIC-11]. Belakangan ini, sistem rekomendasi telah menjadi sangat populer dan menjadi bagian yang penting dari berbagai layanan situs jual beli, sosial media, hiburan, bahkan situs pencarian yang sering digunakan oleh masyarakat.

Sistem rekomendasi pada umumnya terbagi menjadi tiga pendekatan, yaitu *collaborative filtering*, *content-based filtering*, dan hibrid [ADO-05]. *Collaborative filtering* bekerja dengan membuat suatu database *preferences* suatu item dari para pengguna [SAR-01], dalam kasus ini adalah bagaimana perilaku mahasiswa sebelumnya dalam melakukan pemilihan mata kuliah. *Content-based filtering* memberikan rekomendasi berdasarkan kemiripan karakter antar item mata kuliah yang tersedia [ADO-05]. *Collaborative filtering* akan sesuai apabila diterapkan untuk mata kuliah yang sudah pernah diambil sebelumnya, namun tidak dapat memberikan rekomendasi terhadap mata kuliah yang baru dan belum pernah diambil oleh mahasiswa. Sedangkan, *content-based filtering* dapat digunakan untuk memberikan rekomendasi terhadap

semua mata kuliah, dengan mengabaikan perilaku pengguna sebelumnya. Sistem rekomendasi hibrid mencoba mengatasi batasan kemampuan dari *collaborative filtering* dan *content-based filtering* dengan menggabungkan kedua metode tersebut.

Dalam penelitian ini, penulis membandingkan *content-based filtering* dan *collaborative filtering* untuk memberikan rekomendasi pemilihan mata kuliah kepada pengguna. Kedua metode tersebut dipilih untuk mengamati perbedaan di antara keduanya dalam memberikan rekomendasi kepada pengguna.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka penulis merumuskan masalah sebagai berikut:

1. Bagaimana sistem rekomendasi dengan *collaborative filtering* dapat memberikan rekomendasi pemilihan mata kuliah pada mata kuliah pilihan yang sudah pernah diambil sebelumnya secara tepat sasaran
2. Bagaimana sistem rekomendasi dengan *content-based filtering* dapat memberikan rekomendasi pemilihan mata kuliah pada mata kuliah pilihan baru yang belum pernah diambil sebelumnya secara tepat sasaran
3. Bagaimana mengetahui keakuratan / kelayakan sistem rekomendasi yang dibangun

1.3 Batasan Masalah

Penelitian ini dibatasi oleh hal-hal sebagai berikut:

1. Sistem rekomendasi pemilihan mata kuliah ini hanya digunakan sebagai rekomendasi bagi pengguna, keputusan pemilihan mata kuliah ada pada pengguna.
2. Penelitian ini difokuskan pada aplikasi berbasis web.
3. Metode yang digunakan adalah *item-based collaborative filtering*, *user-based collaborative filtering*, dan *content-based filtering*.
4. Data latih yang digunakan adalah data kartu hasil studi mahasiswa Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya tahun 2014/2015 genap.

5. Sistem rekomendasi yang dibangun tidak mengakomodasi perubahan kurikulum akademik dan kesetaraan mata kuliah.
6. Sistem rekomendasi yang dibangun tidak mengakomodasi keminatan studi mahasiswa.
7. Sistem rekomendasi yang dibangun tidak mengakomodasi kesamaan nilai yang diperoleh mahasiswa yang menjadi data latih dalam pemberian rekomendasi dengan *content-based filtering*.

1.4 Tujuan

Adapun yang menjadi tujuan dalam penelitian ini adalah sebagai berikut:

1. Merancang dan membuat sistem rekomendasi dengan *collaborative filtering* sehingga dapat memberikan rekomendasi pemilihan mata kuliah pada mata kuliah pilihan yang sudah pernah diambil sebelumnya secara tepat sasaran
2. Merancang dan membuat sistem rekomendasi dengan *content-based filtering* sehingga dapat memberikan rekomendasi pemilihan mata kuliah pada mata kuliah pilihan baru yang belum pernah diambil sebelumnya secara tepat sasaran
3. Mengetahui keakuratan / kelayakan sistem rekomendasi yang dibangun

1.5 Manfaat

Hasil penelitian yang diharapkan oleh penulis adalah agar memberikan manfaat sebagai berikut:

1. Memberikan kemudahan bagi mahasiswa Universitas Brawijaya dalam melakukan pemilihan mata kuliah yang tepat.
2. Memberikan manfaat bagi pengembangan ilmu pengetahuan dan teknologi serta menambah wawasan ilmu bagi penelitian dan pengembangan selanjutnya.

1.6 Metodologi Penelitian

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi penelitian yang digunakan dalam penulisan skripsi ini adalah:

1. Studi literatur
Membaca dan mempelajari literatur (jurnal, buku dan artikel dari internet) mengenai sistem rekomendasi.
2. Pendefinisian dan analisis masalah
Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.
3. Perancangan dan implementasi sistem
Membuat perancangan perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil rancangan yang telah dibuat.
4. Uji coba dan analisa hasil implementasi
Menguji perangkat lunak, dan menganalisa hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian menarik kesimpulannya.

1.7 Sistematika Penulisan

Penelitian ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN
Berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.
2. BAB II KAJIAN PUSTAKA DAN DASAR TEORI
Menjelaskan tentang dasar teori dan referensi yang berhubungan dengan sistem rekomendasi.
3. BAB III METODE PENELITIAN DAN PERANCANGAN
Membahas tentang studi literatur, perancangan sistem dan uraian permasalahan.

4. BAB IV IMPLEMENTASI

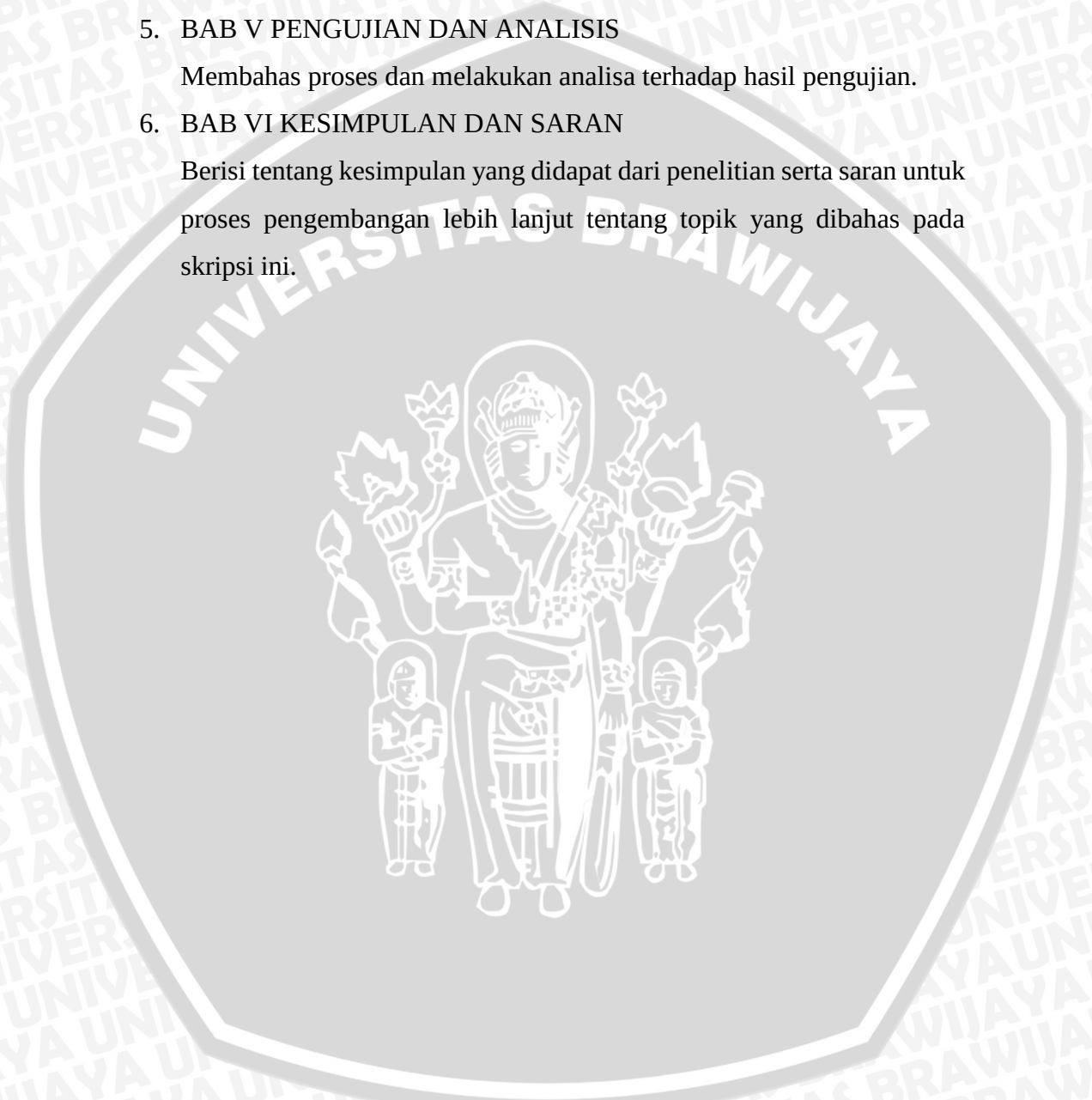
Membahas tentang implementasi dari aplikasi yang dibuat dan memberikan penjelasan dari rancangan input dan output, serta melakukan pengujian terhadap aplikasi yang dibuat.

5. BAB V PENGUJIAN DAN ANALISIS

Membahas proses dan melakukan analisa terhadap hasil pengujian.

6. BAB VI KESIMPULAN DAN SARAN

Berisi tentang kesimpulan yang didapat dari penelitian serta saran untuk proses pengembangan lebih lanjut tentang topik yang dibahas pada skripsi ini.



BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

Sistem rekomendasi saat ini banyak diimplementasikan dalam berbagai bidang, di antaranya adalah dalam pemberian rekomendasi produk pada pengunjung toko *online* Amazon serta pemberian rekomendasi film pada Netflix.

Kebutuhan saran atau rekomendasi mengenai *item* pada suatu layanan *online* menjadi dasar utama dari berkembangnya sistem rekomendasi. Dengan adanya sistem rekomendasi, aktifitas penjualan serta kepuasan pengguna dari suatu layanan web akan semakin meningkat.

Sistem rekomendasi dibagi menjadi tiga pendekatan yaitu *content-based recommendation*, *collaborative recommendation*, dan *hybrid recommendation* [ADO-05]. Pada Tabel 2.1 dijabarkan mengenai penelitian-penelitian yang telah dilakukan sebelumnya dalam perancangan dan implementasi sistem rekomendasi.

Tabel 2.1 Perbandingan penelitian-penelitian sebelumnya tentang sistem rekomendasi

Penelitian	Masukan	Metode	Keluaran
A Recommender System for On-line Course Enrolment: An Initial Study	Data historis pilihan mata kuliah mahasiswa	<i>Item-based collaborative filtering</i>	Daftar mata kuliah yang direkomendasikan
Perancangan dan Implementasi Sistem Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Metode Hybrid	Data rating mata kuliah, data nilai mata kuliah, dokumen mata kuliah	<i>Switching hybrid recommender</i> dengan kombinasi antara <i>collaborative filtering (cosine similarity dan weighted sum)</i> serta <i>content-based filtering (TF-IDF)</i>	Daftar mata kuliah yang direkomendasikan
Item-Based Collaborative Filtering Recommendation Algorithms	Data rating film dari pengguna MovieLens	<i>Item-based collaborative filtering</i>	Daftar film yang direkomendasikan
Content-Based Book Recommending Using Learning for Text Categorization	Data deskripsi dan rating buku	<i>Content-based filtering</i>	Daftar buku yang direkomendasikan
A Decision Tree Based Recommender System	Data rating film dari pengguna MovieLens	<i>Content-based filtering</i> dengan <i>decision tree</i>	Daftar film yang direkomendasikan

O'Mahony dan Smyth [OMA-07] dari University College Dublin telah melakukan studi awal mengenai pengembangan sistem rekomendasi mata kuliah dengan menggunakan *collaborative recommender*. Hasil penelitian yang telah mereka lakukan memberikan performa *recall* sebesar 66%.

Naufal [NAU-13] juga telah melakukan penelitian serupa dengan menggunakan *switching hybrid recommender system* dengan melakukan kombinasi antara *collaborative filtering* dan *content-based filtering*. *Collaborative filtering* dilakukan dengan menghitung *similarity* dengan *cosine similarity* dan prediksi dengan *weighted sum*. *Content-based filtering* dilakukan dengan menghitung bobot TF-IDF pada dokumen mata kuliah. Nilai *recall* yang dicapai oleh metode *collaborative filtering* adalah 67,5% dan *content-based filtering* 57,5%.

Sarwar dkk. pada tahun 2001 [SAR-01] telah melakukan penelitian mengenai *Item-based collaborative filtering* terhadap pemberian rekomendasi film dengan menggunakan data dari MovieLens. Penelitian tersebut menguji bagaimana teknik perhitungan similaritas berpengaruh pada tingkat error dari sistem yang dibangun. Hasilnya, *adjusted cosine similarity* memberikan nilai MAE 0,72 yang lebih rendah dari *cosine similarity* maupun *correlation similarity*.

Mooney dan Roy dari University of Texas [MOO-00] melakukan penelitian mengenai *content-based recommender system* untuk memberikan rekomendasi buku dari pengguna Amazon. Penelitian yang dilakukan dengan melakukan *learning* terhadap data deskripsi buku. Penelitian tersebut menghasilkan nilai uji akurasi bervariasi dari 59% hingga 86,6% dan *recall* bernilai 38,3% hingga 93,2%.

Gershman dkk. dari University of Negev [GER-10] melakukan penelitian untuk memberikan rekomendasi film dari pengguna MovieLens dengan memanfaatkan *Content-based filtering* dengan *decision tree*. Peneliti menggunakan metode ini karena *decision tree* dapat memberikan daftar beberapa *item* yang direkomendasikan, bukan hanya satu *item* sehingga mengurangi jumlah pencarian rekomendasi yang dilakukan. Dengan mengamati perubahan *depth* dari *tree* yang dibentuk, didapatkan hasil bahwa *decision tree*

akan optimal pada kedalaman *tree* tertentu, dan kemudian cenderung menurun seiring dengan penambahan *tree depth*.

Menurut Pazzani [PAZ-07], sistem rekomendasi dengan *collaborative filtering* bekerja dengan baik pada data terstruktur dengan beberapa atribut. Hal ini sesuai dengan data yang tersedia pada penelitian yang dilakukan oleh Penulis. Ricci [RIC-11] mengungkapkan bahwa kemiripan *item* dihitung berdasarkan pada fitur-fitur yang ada pada *item* yang dibandingkan. Oleh karenanya, metode ini tidak bergantung pada situasi apakah *item* tersebut merupakan *item* baru (yang belum pernah dipilih oleh pengguna manapun) maupun bukan *item* baru.

Collaborative filtering telah sukses dalam segi penelitian maupun implementasinya [SAR-01]. *Collaborative filtering* mampu memberikan rekomendasi yang lebih dari sekedar kemiripan *item*, melainkan dapat memberikan rekomendasi dengan mempelajari selera pengguna dan mencari pengguna yang memiliki selera yang kurang lebih sama untuk kemudian merekomendasikan *item* yang kemungkinan akan disukai oleh pengguna.

Dari beberapa penelitian terdahulu, dibuktikan bahwa *collaborative filtering* dan *content-based filtering* dapat memberikan rekomendasi kepada pengguna sesuai dengan *preference* dari pengguna.

2.2 Mekanisme Registrasi Akademik Universitas Brawijaya

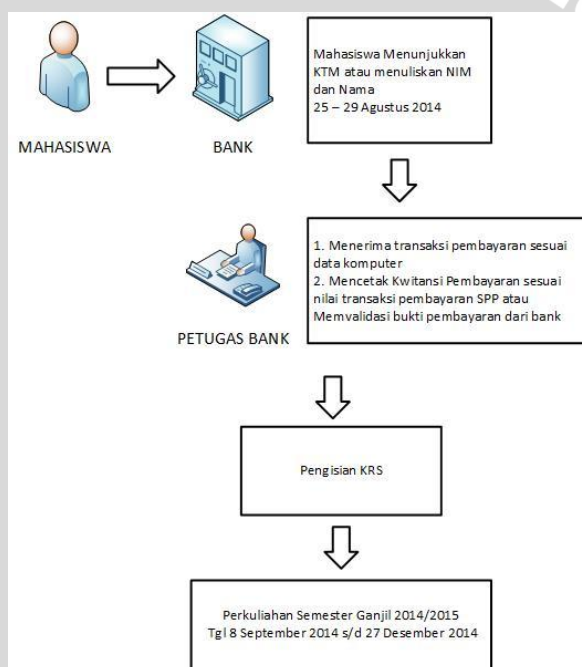
Pelaksanaan registrasi akademik mahasiswa di Universitas Brawijaya dilaksanakan di setiap awal semester. Proses registrasi akademik ini melibatkan mahasiswa, bank sebagai penerima pembayaran, dan operator akademik di tingkat fakultas maupun universitas.

Setiap melakukan registrasi akademik, mahasiswa diwajibkan mengikuti alur seperti pada gambar 2.1. Setelah melakukan pembayaran ke bank, mahasiswa akan berstatus “terdaftar” dan dapat memprogram Kartu Rencana Studi (KRS) pada semester tersebut. Mekanisme pemrograman KRS terbagi menjadi dua, yaitu KRS paket dan KRS biasa. KRS paket diberlakukan kepada mahasiswa baru pada semester pertama dan kedua dan dilakukan oleh operator

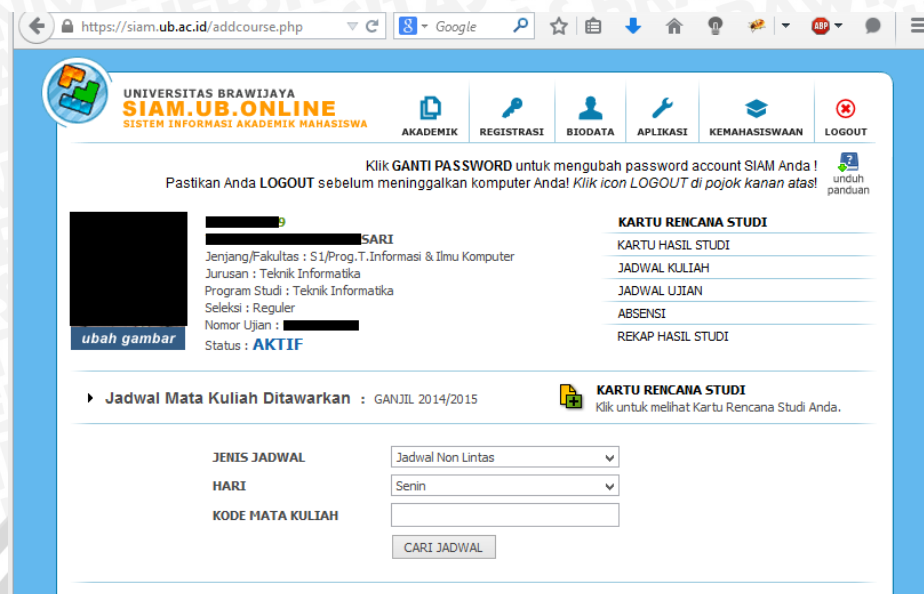
akademik fakultas, bukan melalui aplikasi SIAM. Penelitian ini difokuskan pada pemrograman KRS yang dilakukan oleh mahasiswa (bukan KRS paket).

Pemrograman KRS dilakukan melalui aplikasi SIAM yang dapat diakses di <http://siam.ub.ac.id>. Mahasiswa Universitas Brawijaya memprogram KRS untuk pertama kali pada awal semester ketiga.

Pada saat memprogram KRS di SIAM, mahasiswa dapat melihat serta mencari mata kuliah yang ditawarkan pada semester tersebut. Pada saat ini, tidak ada mekanisme pemberian rekomendasi kepada mahasiswa mengenai saran mata kuliah yang akan diprogram.



Gambar 2.1 Alur registrasi mahasiswa UB periode akademik 2014-2015 ganjil. (sumber: <http://www.ub.ac.id/pengumuman/detail/registrasi-akademik-mahasiswa-universitas-brawijaya-semester-ganjil-2014-2015>)



Gambar 2.2 Antarmuka pemrograman KRS mahasiswa pada aplikasi SIAM

2.3 Sistem Rekomendasi

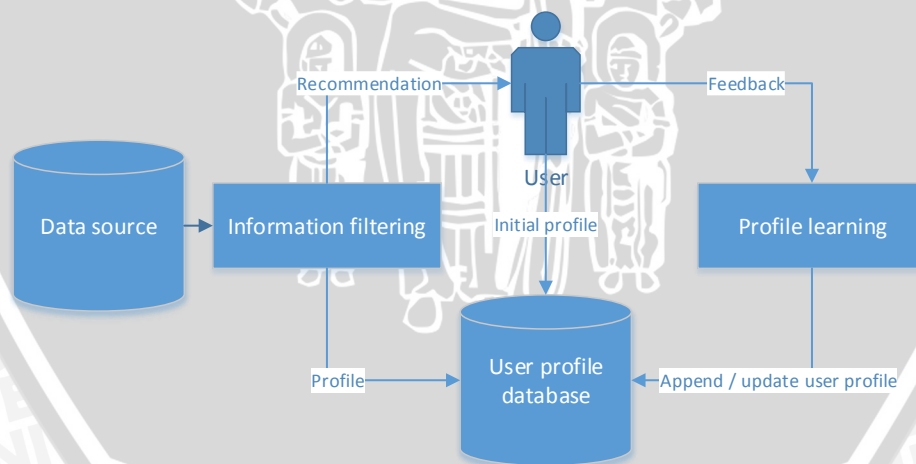
Sistem Rekomendasi adalah alat perangkat lunak dan teknik yang menyediakan saran untuk item yang akan sesuai untuk pengguna [RIC-11]. Saran yang diberikan berkaitan dengan berbagai pembuatan keputusan oleh pengguna (*decision making*), seperti barang yang akan dibeli, musik yang akan didengarkan, ataupun bacaan yang sesuai untuk dibaca.

Sistem rekomendasi akan memberikan rekomendasi yang berbeda kepada setiap pengguna, bukan sekedar memberikan daftar item paling banyak diminati, melainkan memberikan saran mengenai item-item yang mungkin sesuai untuk pengguna. Artinya, setiap pengguna akan mendapatkan rekomendasi yang berbeda, sesuai dengan profil dan minat pengguna tersebut.

Meskipun demikian, penentuan rekomendasi personal mensyaratkan bahwa sistem harus memiliki pengetahuan tentang pengguna. Setiap sistem rekomendasi harus membuat dan *me-maintain* suatu model pengguna atau profil pengguna yang, misalnya, memuat informasi mengenai minat atau *preferences* dari pengguna [JAN-11].

Saat ini, sistem rekomendasi telah banyak digunakan di berbagai bidang kehidupan komputasi sehari-hari, di antaranya: [RAJ-11]

1. Rekomendasi produk: Sistem rekomendasi dapat dikatakan paling bermanfaat untuk toko *online*. Amazon dan berbagai toko *online* lainnya selalu berusaha untuk menarik minat calon pembeli untuk kembali berbelanja di tokonya dengan memberikan rekomendasi produk yang sesuai bagi penggunaanya.
2. Rekomendasi film: Situs web Netflix menawarkan penggunaanya mengenai film-film yang mungkin mereka sukai. Rekomendasi yang diberikan berdasarkan pada *rating* yang diberikan oleh para pengguna situs web tersebut.
3. Rekomendasi artikel berita: Berbagai situs layanan berita juga memberikan rekomendasi kepada penggunaanya berdasarkan artikel yang telah merka baca sebelumnya. Kemiripan (*similarity*) artikel yang direkomendasikan dapat dikalkulasi dari kemiripan kata-kata penting di dalam artikel, atau dengan cara membandingkan selera pengguna dengan pengguna lainnya. Cara yang sama juga juga diaplikasikan di YouTube, maupun berbagai situs lain yang memiliki *update* konten secara reguler.



Gambar 2.3 Arsitektur sistem rekomendasi

Sistem rekomendasi bekerja dengan melakukan analisa terhadap sumber data yang ada. Data akan dikalkulasi dan menghitung nilai kemiripan antar *item* maupun antar selera / profil pengguna.

Pada gambar 2.3 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 pengguna untuk kemudian dinilai oleh pengguna dengan memberikan *feedback*. *Feedback* inilah yang digunakan oleh sistem untuk meng-update profil pengguna. Dalam perkembangan sistem, *feedback* yang diberikan pengguna disimpan dan diproses untuk menghasilkan rekomendasi yang lebih baik.

Menurut Breese [BRE-98], 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 digunakan oleh sistem untuk melakukan prediksi terhadap data yang *real*.

2.4 Sistem Rekomendasi Pemilihan Mata Kuliah

Penerapan sistem rekomendasi dalam pemilihan mata kuliah / *course enrolment* belum banyak dilakukan. O'Mahony dalam publikasi pada tahun 2007 menemukan bahwa sistem rekomendasi pemilihan mata kuliah di University College Dublin menggunakan *collaborative filtering* dapat memberikan ketepatan *recall* hingga 66% [OMA-07].

Penelitian lain yang dilakukan oleh Naufal dkk di IT Telkom Bandung dengan menggunakan *switching hybrid recommender system*, menemukan bahwa sistem yang dibangun dengan *collaborative filtering* memberikan ketepatan *precision* dan *recall* 20.25% dan 67.5% lebih tinggi 3.05% dan 10% dari sistem yang dibangun dengan *content-based filtering*[NAU-13]. Naufal

melakukan *collaborative filtering* dengan menggunakan teknik *cosine-based similarity* dan dikombinasikan dengan *content-based filtering* dengan teknik $TF*IDF$.

Penelitian ini akan mencoba mengkombinasikan *collaborative filtering* dengan teknik *adjusted cosine similarity* digabungkan dengan *content-based filtering* dengan menggunakan *decision tree* secara *linear hybrid*.

2.5 Content-Based Filtering

Sistem rekomendasi dengan metode *content-based* melakukan proses *learning* untuk merekomendasikan *item* yang mirip dengan *item* sebelumnya yang disukai atau dipilih oleh *user*. Kemiripan *item* dihitung berdasarkan pada fitur-fitur yang ada pada *item* yang dibandingkan [RIC-11]. Oleh karenanya, metode ini tidak bergantung pada situasi apakah *item* tersebut merupakan *item* baru (yang belum pernah dipilih oleh pengguna manapun) maupun bukan *item* baru. Sebagai contoh dalam suatu sistem rekomendasi toko buku *online*, jika seorang *user* telah membeli suatu buku di kategori komputer, maka sistem akan mencoba merekomendasikan buku yang ada di kategori yang sama, yang mungkin akan disukai oleh pengguna tersebut.

Metode ini pada umumnya banyak digunakan dalam untuk penyaringan informasi pada sistem berbasis teks, namun akhir-akhir ini juga banyak digunakan pada kasus-kasus yang lain untuk mengatasi kelemahan sistem rekomendasi dengan metode yang lain. Antara lain, metode ini dapat mengatasi kelemahan *collaborative filtering* yang tidak dapat memberikan rekomendasi terhadap *item* baru.

Metode *content-based* berakar dari bidang ilmu *information retrieval* (sistem temu kembali informasi) dan *information filtering* [ADO-05]. Penelitian mengenai *information retrieval* dan *information filtering* pada awal mulanya banyak menggunakan teks sebagai objek penelitian. Hal ini yang menyebabkan metode *content-based* banyak digunakan pada sistem berbasis teks.

Content yang menjadi objek *learning* pada metode ini umumnya adalah informasi fitur mengenai *item* yang tersedia (*item-based*), namun pada perkembangannya, profil pengguna juga dapat menjadi *content* yang dipelajari kemiripannya (*user-based*). Sebagai contoh dalam kasus toko buku *online*, profil pengguna A memiliki kemiripan dengan profil pengguna B (memiliki kemiripan buku yang dilihat, disukai, dan dibeli) maka sistem rekomendasi akan mencoba merekomendasikan buku yang disukai oleh pengguna A kepada pengguna B dan sebaliknya.

Ada beberapa teknik yang biasa digunakan dalam *content-based filtering*, di antaranya adalah *bayesian classifiers*, *clustering*, *decision trees*, dan *TF*IDF* [ADO-05].

2.5.1 Content-based Filtering dengan Decision Tree

Menurut Pazzani [PAZ-07], *content-based filtering* pada kumpulan *item* yang memiliki atribut terstruktur (bukan atribut teks) dengan jumlah atribut yang sedikit akan diuntungkan dengan penggunaan teknik *decision trees* baik dari sisi performa, kemudahan, dan *understandability*-nya.

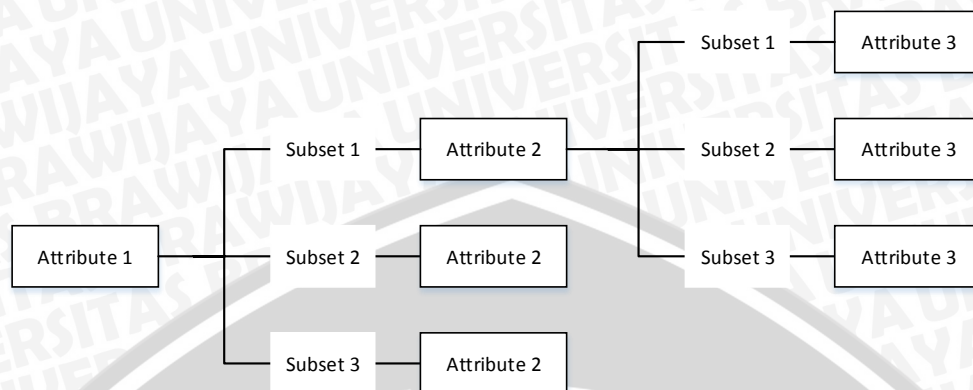
DATA → DECISION TREE → RULES

Gambar 2.4 Proses *decision tree classifiers*

Decision tree classifiers adalah salah satu metode yang digunakan untuk membantu pembuatan keputusan secara bertahap (*multistage*). Pada umumnya, *decision tree classifiers* melakukan analisa dan konversi data menjadi *decision tree* dan sekumpulan *rules* [SAF-91].

Tabel 2.2 Contoh tabel data *decision tree*

Identity Attribute	Attribute 1	Attribute 2	...	Attribute n	Target Attribute
Item a					
Item b					
...					
Item m					



Gambar 2.5 Decision tree

2.5.2 Pembentukan Decision Tree dengan Algoritma ID3

Algoritma ID3 (*Iterative Dichotomiser 3*) adalah algoritma *decision tree learning* yang ditemukan oleh Ross Quinlan [QUI-86]. Algoritma ini digunakan untuk menghasilkan *decision tree* dari suatu dataset. Algoritma ID3 adalah algoritma yang sederhana sehingga relatif cepat.

Algoritma ini bekerja dengan tahapan sebagai berikut:

1. Menghitung nilai entropi dan *information gain* dari masing-masing atribut dari suatu dataset S
2. Membagi dataset S ke dalam suatu subset dengan menggunakan suatu atribut yang memiliki *information gain* yang paling besar
3. Membuat *decision tree node* dari atribut tersebut
4. Kembali ke langkah 1 untuk menghitung subset yang belum dihitung

ID3 bekerja dengan baik terhadap data diskrit. Oleh karenanya, ID3 dapat digunakan untuk memproses data mata kuliah pada penelitian ini.

$$Entropi(S) = \sum_i^c -p_i \log_2 p_i, \text{ dimana} \quad (2.1)$$

c = jumlah kelas dari atribut target

p_i = peluang untuk kelas i

$$Gain(S, A) = Entropi(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropi(S_v) \quad (2.2)$$

dimana

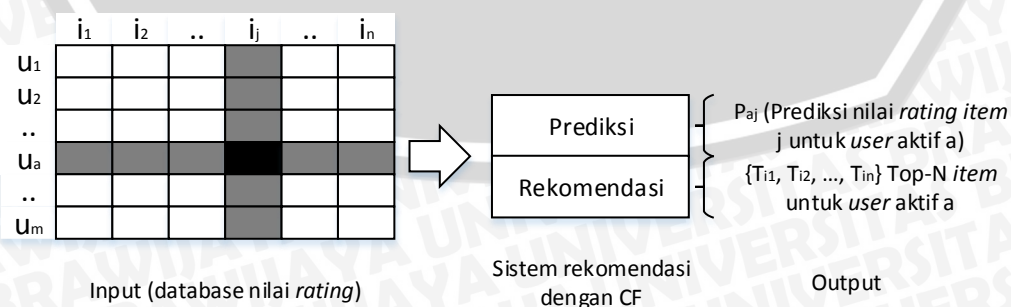
- A = atribut
 v = nilai yang mungkin untuk atribut A
 $Values(A)$ = himpunan nilai yang mungkin untuk atribut A
 $|S_v|$ = banyaknya dataset S dengan kelas v
 $|S|$ = banyaknya dataset S
 $Entropi(S_v)$ = entropi untuk dataset S yang memiliki kelas v

Untuk menghitung nilai Entropi dari suatu dataset S , digunakan persamaan (2.1). Sedangkan untuk menghitung *information gain* digunakan persamaan (2.2).

2.6 Collaborative Filtering

Collaborative filtering bekerja dengan membuat suatu *database* yang berisi *item preferences* dari pengguna [SAR-01]. Sebagai contoh, *user A* dicocokkan terhadap data yang ada di *database* untuk menemukan *neighbor*, yaitu *user* lain yang memiliki kemiripan *preference* dengan *user A*. Dari kecocokan yang didapatkan, *user A* akan diberikan rekomendasi *item* yang sebelumnya pernah dipilih oleh *neighbor* dari *user A*.

Metode ini telah sukses dalam segi penelitian maupun implementasinya [SAR-01]. *Collaborative filtering* mampu memberikan rekomendasi yang lebih dari sekedar kemiripan *item*, melainkan dapat memberikan rekomendasi dengan mempelajari selera pengguna dan mencari pengguna yang memiliki selera yang kurang lebih sama.



Gambar 2.6 Proses collaborative filtering

Pada umumnya, *collaborative filtering* akan melakukan prediksi dan rekomendasi. Prediksi yang dimaksud adalah prediksi nilai *rating* yang akan diberikan oleh *user* aktif (yang akan diberikan rekomendasi) terhadap *item* yang tersedia. Seluruh prediksi nilai *rating* ini akan diurutkan untuk menghasilkan daftar top-*n item* yang akan diberikan ke *user* aktif sebagai rekomendasi.

2.6.1 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 pengguna-pengguna yang memiliki *similar preferences* dari pengguna aktif. Kemudian, dari setiap pengguna yang memiliki kesamaan *preferences* tersebut, didapatkan *items* yang dapat ditawarkan kepada pengguna aktif. Masing-masing *item* tersebut akan dihitung prediksi *rating*-nya berdasarkan *rating* dari para pengguna sebelumnya.

Asumsinya adalah sebagai berikut: (a) jika para pengguna memiliki kesamaan *tastes* di masa lampau, maka mereka juga akan memiliki kesamaan *tastes* di masa depan dan (b) *user preferences* akan tetap stabil dan sama dari waktu ke waktu [JAN-11].

Tabel 2.3 Contoh data rating pengguna

User	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	?

Sebagai contoh, Tabel 2.2 menunjukkan matriks hubungan antara pengguna dan *item* beserta nilai *rating* yang mereka berikan. Tugas dari sistem rekomendasi adalah memberikan prediksi nilai yang akan diberikan oleh pengguna aktif X pada *item* yang belum pernah diberikan nilai *rating* olehnya. Untuk melakukannya, sistem melakukan pencarian pengguna-pengguna lain yang memiliki kesamaan *preferences*, dilihat dari nilai *rating* yang telah mereka

berikan. Kemudian dari sekelompok pengguna-pengguna tersebut, sistem melakukan penghitungan untuk mengetahui apakah pengguna aktif X akan menyukai *item* tersebut.

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

$sim(a, b)$ = similaritas *preference* pengguna a dan b

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

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

$$P_{u,i} = \frac{\sum_{all \text{ similar users } N} (S_{u,N} * R_{N,i})}{\sum_{all \text{ similar users } N} (S_{u,N})}, \text{ dimana} \quad (2.4)$$

$P_{u,i}$ = prediksi *rating* dari pengguna u terhadap item i

$S_{u,N}$ = similaritas pengguna u dengan item N

$R_{N,i}$ = rating user N terhadap item i

$$P_{u,i} = \bar{R}_u + \frac{\sum_{all \text{ similar users } N} (S_{u,N} * (R_{N,i} - \bar{R}_N))}{\sum_{all \text{ similar users } N} (S_{u,N})}, \text{ dimana} \quad (2.5)$$

$P_{u,i}$ = prediksi *rating* dari pengguna u terhadap item i

$S_{u,N}$ = similaritas user u dengan item N

$R_{N,i}$ = rating user N terhadap item i

Salah satu cara yang banyak digunakan dalam menghitung kemiripan $sim(a, b)$ dari dua vektor pengguna a dan b adalah dengan menggunakan *Pearson's corellation coefficient* seperti ditunjukkan pada persamaan (2.3). Nilai similaritas yang dihasilkan pada persamaan (2.3) berkisar antara -1 dan $+1$, dimana -1 berarti korelasi negatif dan $+1$ korelasi positif, mengindikasikan bahwa kedua pengguna yang dibandingkan memiliki tingkat kesamaan *preference* yang tinggi [KWO-09].

Untuk menghitung prediksi nilai *rating*, biasa digunakan persamaan *weighted sum* (2.4) atau persamaan (2.5) dari Herlocker[HER-99]. Kedua persamaan tersebut menghasilkan nilai prediksi *rating* sesuai dengan *range* nilai *rating* yang telah ada pada sistem.

2.6.2 Item-based Collaborative Filtering

Teknik *item-based collaborative filtering* melakukan analisa terhadap matriks *user-item* untuk mengetahui hubungan antar item, kemudian menggunakan hubungan antar item tersebut untuk melakukan komputasi rekomendasi kepada pengguna [SAR-01].

	Item 1	Item 2	...	Item i	...	Item j	...	...	Item n-1	Item n
User u1				R		R				
User u2				-		R				
...										
...										
User u				R		R				
...										
...										
User m-2				R		-				
User m-1				R		R				
User m				-		-				

Similarity antar item dihitung dengan melihat *item* yang telah diberi nilai *rating* oleh pengguna. Pada contoh ini, similaritas s_{ij} dari *item* i dan j didapatkan dengan melihat nilai *rating* dari user $u1$, u , dan $m-1$.

Gambar 2.7 Proses komputasi similaritas

Salah satu tahap yang penting dalam *item-based collaborative filtering* adalah menghitung *similarity* atau kemiripan antar *item* untuk kemudian memilih beberapa *item* yang paling mirip. Penghitungan similaritas dilakukan dengan melihat *co-rated items* (*item-item* yang diberikan nilai *rating* oleh *user*). Ilustrasi proses ini dapat dilihat pada Gambar 2.7.

Ada beberapa teknik yang biasa digunakan dalam *item-based collaborative filtering* untuk menghitung kemiripan *item*. Dalam penelitian ini, akan digunakan *adjusted cosine similarity* untuk menghitung kemiripan antar *item*.

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| * \|\vec{j}\|}, \text{ dimana} \quad (2.6)$$

$sim(i, j)$ = similarity item i dan j

$\vec{i}$ = vektor item i

$\vec{j}$ = vektor item j

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}, \text{ dimana} \quad (2.7)$$

$sim(i, j)$ = similarity item i dan j

$\bar{R}_u$ = rata-rata rating dari user u

$R_{u,i}, R_{u,j}$ = rating user u terhadap item i, j

$$P_{u,i} = \frac{\sum_{all \text{ similar items}, N} (S_{i,N} * R_{u,N})}{\sum_{all \text{ similar items}, N} (|S_{i,N}|)}, \text{ dimana} \quad (2.8)$$

$P_{u,i}$ = prediksi *rating* dari pengguna u terhadap item i

$S_{i,N}$ = similaritas item i dengan item N

$R_{u,N}$ = rating user u terhadap item N

$$P_{u,i} = \bar{R}_i + \frac{\sum_{all \text{ similar items}, N} (S_{i,N} * (R_{u,N} - \bar{R}_N))}{\sum_{all \text{ similar items}, N} (|S_{i,N}|)}, \text{ dimana} \quad (2.9)$$

$P_{u,i}$ = prediksi *rating* dari pengguna u terhadap item i

$S_{i,N}$ = similaritas item i dengan item N

$R_{u,N}$ = rating user u terhadap item N

$\bar{R}_N$ = rata-rata rating item N

Adjusted cosine similarity (2.7) adalah penyempurnaan dari *cosine-based similarity* (2.6). *Cosine-based similarity* tidak mempertimbangkan skala *rating* yang biasa diberikan oleh *user*. Pada *cosine-based similarity*, item i dan j dianggap sebagai dua vektor. Kemiripan antara item i dan j didapatkan dari nilai *cosine* dari kedua vektor tersebut. *Adjusted cosine similarity* menyempurnakan algoritma ini dengan masing-masing nilai *rating* dengan rata-rata nilai *rating* yang diberikan oleh *user*.

Pada kebanyakan sistem rekomendasi, nilai similaritas biasanya belum cukup untuk dapat memberikan rekomendasi kepada pengguna aktif. Setelah mendapatkan nilai similaritas antar *item*, biasanya sistem akan melakukan perhitungan prediksi nilai *rating* yang akan diberikan pengguna aktif terhadap *item* yang akan direkomendasikan. Persamaan *weighted sum* (2.8) dan

persamaan (2.9) dari Herlocker[HER-99] biasa digunakan untuk melakukan penghitungan prediksi nilai *rating* tersebut.

2.7 Sistem Rekomendasi Hibrid

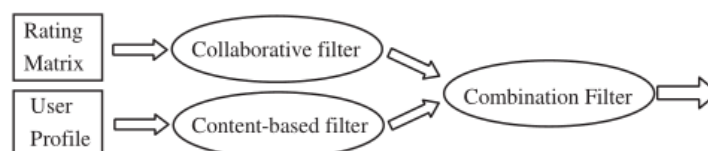
Sistem rekomendasi hibrid bekerja dengan mengkombinasikan metode-metode sistem rekomendasi yang ada [RIC-11]. Metode yang dimaksud antara lain adalah *collaborative filtering*, *content-based*, *demographic*, *knowledge-based*, maupun *community-based*.

Sistem rekomendasi yang menggabungkan metode A dan metode B bertujuan untuk saling mengatasi keterbatasan yang terdapat pada masing-masing metode sehingga menghasilkan rekomendasi yang lebih baik. Sebagai contoh, sistem dengan metode *collaborative filtering* tidak dapat merekomendasikan *item* baru kepada pengguna karena *item* baru tersebut belum pernah dipilih dan dinilai oleh pengguna. Keterbatasan ini tidak akan menjadi masalah bagi metode *content-based* karena metode ini hanya perlu mengamati kemiripan *item* ataupun *user*. Dengan demikian, *item* baru tetap dapat direkomendasikan kepada pengguna.

Terdapat beberapa pendekatan dalam mengkombinasikan *collaborative filtering* dan *content-based*. Pada umumnya, pendekatan yang digunakan terbagi menjadi tiga [KIM-06], sebagai berikut:

1. *Linear combination model*

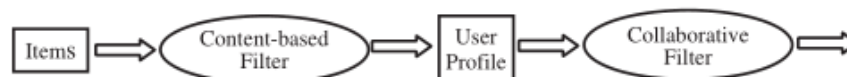
Pendekatan ini dilakukan dengan menggabungkan hasil akhir nilai *rating* dari masing-masing metode. Gambar 2.8 menunjukkan ilustrasi pendekatan ini.



Gambar 2.8 *Linear combination model*

2. Sequential combination model

Pendekatan ini dilakukan secara sekuensial (berurutan). Artinya, sistem melakukan perhitungan nilai *rating* dengan salah satu metode, kemudian baru melakukan perhitungan dengan metode yang lainnya. Ilustrasi pendekatan ini dapat dilihat di gambar 2.9.



Gambar 2.9 Sequential combination model

3. Mixed combination model

Pendekatan ini biasanya dilakukan pada sistem yang menggunakan lebih dari satu teknik perhitungan nilai *rating*. Sistem dapat dibangun dengan membuat rekomendasi yang dikombinasikan secara linear, kemudian dilakukan kombinasi kembali secara sekuensial.

2.8 Pengukuran Tingkat Akurasi

Pada sistem rekomendasi yang memiliki urutan *ranking*, misalkan sistem rekomendasi pemilihan film, buku, musik, dan sebagainya, penghitungan tingkat akurasi hasil rekomendasi dilakukan dengan metode *Mean Absolute Error (MAE)*. MAE mengevaluasi tingkat akurasi sistem dengan membandingkan prediksi nilai yang diberikan terhadap nilai *rating* sebenarnya yang diberikan oleh pengguna [SAR-01].

Pada penelitian ini, data yang akan direkomendasikan tidak memiliki urutan *ranking*, sehingga penghitungan tingkat akurasi hasil rekomendasi dilakukan dengan membandingkan banyaknya hasil rekomendasi yang cocok dengan data aktual terhadap banyaknya keseluruhan data yang diujikan.

$$Akurasi = \frac{N_m}{N} \quad (2.10)$$

N_m = Banyaknya hasil rekomendasi yang cocok dengan data uji

N = Banyaknya data uji

BAB III

METODE PENELITIAN DAN PERANCANGAN

Pada bab ini menjelaskan langkah-langkah yang akan ditempuh dalam menyusun penelitian. Tahapan penelitian ini dirancang untuk membangun sistem rekomendasi yang dapat memberikan rekomendasi mata kuliah pilihan kepada mahasiswa yang melakukan registrasi akademik menggunakan sistem rekomendasi.

3.1 Studi Literatur

Tahapan ini dilakukan untuk mendalami konsep dan teori-teori yang akan diterapkan pada penelitian ini. Teori-teori yang berkaitan dengan sistem rekomendasi yang digunakan sebagai dasar teori penelitian didapatkan dari buku, jurnal, dan berbagai sumber di internet. Metode yang akan dipelajari dalam studi literatur yaitu:

1. *Content-based filtering* dengan *decision tree*
2. *Item-based collaborative filtering*

3.2 Data Penelitian

Untuk mengetahui kinerja sistem rekomendasi yang dibangun, dibutuhkan dataset mata kuliah yang telah ditempuh oleh mahasiswa Universitas Brawijaya. Dataset mata kuliah mahasiswa yang digunakan dalam penelitian ini didapatkan dari Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya pada tahun akademik 2014/2015 ganjil.

Data yang digunakan mencakup dua jenis data. Pertama, data KHS mahasiswa yang digunakan sebagai data *sample* untuk mengetahui apakah mahasiswa PTIIK menempuh mata kuliah yang ditawarkan. Data kedua adalah tabel kompetensi utama lulusan mata kuliah yang didapatkan dari Buku Pedoman Pendidikan Program Teknologi Informasi dan Ilmu Komputer Tahun Akademik 2012/2013.

3.3 Deskripsi Umum Sistem

Secara umum, sistem yang dibangun adalah perangkat lunak yang memberikan rekomendasi mata kuliah pilihan kepada mahasiswa yang melakukan registrasi akademik. Dengan sistem rekomendasi ini, diharapkan mahasiswa dapat memilih mata kuliah pilihan yang tepat. Metode yang akan digunakan dalam penelitian ini adalah *collaborative filtering* dan *content-based filtering*.

3.4 Perancangan Sistem

Pembuatan sistem rekomendasi pada penelitian ini terbagi menjadi dua bagian, yaitu *content-based filtering* dan *collaborative filtering*.

3.4.1 Content-based Filtering

Pada penelitian ini, *content-based filtering* mencoba mengatasi kelemahan dari *collaborative filtering* yang tidak dapat merekomendasikan mata kuliah yang baru yang belum pernah ditempuh oleh mahasiswa manapun sebelumnya.

Tahapan yang dilakukan untuk menghasilkan rekomendasi dengan *content-based filtering* dengan menggunakan *decision tree* adalah mempersiapkan data, membentuk *decision tree*, dan membentuk *rules*.

3.4.1.1 Persiapan Data

Content-based filtering dengan *decision tree* membutuhkan informasi atribut-atribut suatu mata kuliah yang digunakan untuk membuat *branch*. Dalam penelitian ini, setiap *item* mata kuliah pilihan memiliki beberapa atribut kompetensi utama lulusan mata kuliah serta atribut nilai yang menjadi *target attribute*.

Content-based filtering pada penelitian ini melakukan *filtering* terhadap mata kuliah yang telah diprogram sebelumnya oleh pengguna aktif tersebut tanpa memperhatikan mata kuliah yang diprogram oleh pengguna yang lainnya.

Tabel 3.1 Contoh daftar mata kuliah yang diprogram mahasiswa tertentu

Mata Kuliah	Kompetensi Utama Lulusan					Nilai (atribut target)
	U1	U2	U3	U4	U5	
MK 1	1	0	0	1	1	A
MK 2	1	0	1	1	1	C+
MK 3	0	1	1	0	0	B
MK 4	1	1	0	1	0	A
MK 5	1	0	0	1	0	C
MK 6	1	1	1	1	1	A
MK 7	0	0	0	0	0	B+

Sebagai contoh, Tabel 3.1 menampilkan daftar mata kuliah yang telah diprogram oleh seorang mahasiswa. Angka 1 berarti memenuhi, dan angka 0 berarti tidak memenuhi kompetensi utama lulusan yang dimaksud. Data inilah yang akan digunakan untuk memberikan rekomendasi mata kuliah pilihan kepada mahasiswa tersebut.

3.4.1.2 Pembentukan Decision Tree dengan Algoritma ID3

Data dari Tabel 3.1 selanjutnya disusun menjadi decision tree dengan menggunakan algoritma ID3 (Iterative Dichotomizer 3). Atribut target adalah nilai, yang memiliki value yakni A, B+, B, C+, C, D+, D dan E. Sedangkan atribut lainnya adalah U1, U2, U3, U4 dan U5 yang masing-masing memiliki value 0 atau 1.

Langkah pertama yang dilakukan adalah menghitung besar entropi dan *information gain* dari setiap atribut. Tahap ini digunakan untuk menentukan atribut mana yang akan menjadi cabang atau simpul akar pada *decision tree*. Atribut dengan *information gain* terbesar akan terpilih menjadi simpul akar. Berikut menunjukkan perhitungan entropi seluruh data *sample* dengan menggunakan rumus (2.1).

$$\begin{aligned}
 Entropy(S) &= \left(-\frac{n(A)}{n} \log_2 \left(\frac{n(A)}{n} \right) \right) + \left(-\frac{n(B+)}{n} \log_2 \left(\frac{n(B+)}{n} \right) \right) + \left(-\frac{n(B)}{n} \log_2 \left(\frac{n(B)}{n} \right) \right) \\
 &\quad + \left(-\frac{n(C+)}{n} \log_2 \left(\frac{n(C+)}{n} \right) \right) + \left(-\frac{n(C)}{n} \log_2 \left(\frac{n(C)}{n} \right) \right) \\
 &\quad + \left(-\frac{n(D+)}{n} \log_2 \left(\frac{n(D+)}{n} \right) \right) + \left(-\frac{n(D)}{n} \log_2 \left(\frac{n(D)}{n} \right) \right) \\
 &\quad + \left(-\frac{n(E)}{n} \log_2 \left(\frac{n(E)}{n} \right) \right) \\
 Entropy(S) &= \left(-\frac{3}{7} \log_2 \left(\frac{3}{7} \right) \right) + \left(-\frac{1}{7} \log_2 \left(\frac{1}{7} \right) \right) + \left(-\frac{1}{7} \log_2 \left(\frac{1}{7} \right) \right) + \left(-\frac{1}{7} \log_2 \left(\frac{1}{7} \right) \right) \\
 &\quad + \left(-\frac{1}{7} \log_2 \left(\frac{1}{7} \right) \right) + \left(-\frac{0}{7} \log_2 \left(\frac{0}{7} \right) \right) + \left(-\frac{0}{7} \log_2 \left(\frac{0}{7} \right) \right) + \left(-\frac{0}{7} \log_2 \left(\frac{0}{7} \right) \right) \\
 &= 2,128085279
 \end{aligned}$$

Perhitungan entropi dan information gain masing-masing atribut dengan menggunakan rumus (2.1) dan (2.2) adalah sebagai berikut, sebagai contoh untuk atribut U1, dengan nilai 0 yang memiliki 2 data dan nilai 1 yang memiliki 5 data.

$$\begin{aligned}
 Entropy(S(0)) &= \left(-\frac{n(0,A)}{n(0)} \log_2 \left(\frac{n(0,A)}{n(0)} \right) \right) + \left(-\frac{n(0,B+)}{n(0)} \log_2 \left(\frac{n(0,B+)}{n(0)} \right) \right) \\
 &\quad + \left(-\frac{n(0,B)}{n(0)} \log_2 \left(\frac{n(0,B)}{n(0)} \right) \right) + \left(-\frac{n(0,C+)}{n(0)} \log_2 \left(\frac{n(0,C+)}{n(0)} \right) \right) \\
 &\quad + \left(-\frac{n(0,C)}{n(0)} \log_2 \left(\frac{n(0,C)}{n(0)} \right) \right) + \left(-\frac{n(0,D+)}{n(0)} \log_2 \left(\frac{n(0,D+)}{n(0)} \right) \right) \\
 &\quad + \left(-\frac{n(0,D)}{n(0)} \log_2 \left(\frac{n(0,D)}{n(0)} \right) \right) + \left(-\frac{n(0,E)}{n(0)} \log_2 \left(\frac{n(0,E)}{n(0)} \right) \right) \\
 Entropy(S(0)) &= \left(-\frac{0}{2} \log_2 \left(\frac{0}{2} \right) \right) + \left(-\frac{1}{2} \log_2 \left(\frac{1}{2} \right) \right) + \left(-\frac{1}{2} \log_2 \left(\frac{1}{2} \right) \right) + \left(-\frac{0}{2} \log_2 \left(\frac{0}{2} \right) \right) \\
 &\quad + \left(-\frac{0}{2} \log_2 \left(\frac{0}{2} \right) \right) + \left(-\frac{0}{2} \log_2 \left(\frac{0}{2} \right) \right) + \left(-\frac{0}{2} \log_2 \left(\frac{0}{2} \right) \right) + \left(-\frac{0}{2} \log_2 \left(\frac{0}{2} \right) \right) = 1 \\
 Entropy(S(1)) &= \left(-\frac{3}{5} \log_2 \left(\frac{3}{5} \right) \right) + \left(-\frac{0}{5} \log_2 \left(\frac{0}{5} \right) \right) + \left(-\frac{0}{5} \log_2 \left(\frac{0}{5} \right) \right) + \left(-\frac{1}{5} \log_2 \left(\frac{1}{5} \right) \right) \\
 &\quad + \left(-\frac{1}{5} \log_2 \left(\frac{1}{5} \right) \right) + \left(-\frac{0}{5} \log_2 \left(\frac{0}{5} \right) \right) + \left(-\frac{0}{5} \log_2 \left(\frac{0}{5} \right) \right) + \left(-\frac{0}{5} \log_2 \left(\frac{0}{5} \right) \right) \\
 &= 1,370951
 \end{aligned}$$

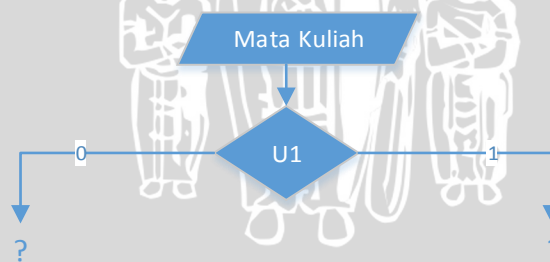
$$Gain(S, U1) = 2.128085279 - \left(-\frac{2}{7} \times 1 \right) + \left(-\frac{5}{7} \times 1,370951 \right) = 0,86312057$$

Tabel 3.2 menunjukkan hasil perhitungan entropi dan *information gain* dari masing-masing atribut untuk memilih *root node*.

Tabel 3.2 Tabel Hasil Entropi dan Information Gain masing-masing Atribut

	Value	Entropi	Gain
U1	0	1	0.863121
	1	1.370951	
U2	0	2	0.591673
	1	0.918296	
U3	0	1.5	0.591673
	1	1.584963	
U4	0	1	0.863121
	1	1.370951	
U5	0	2	0.591673
	1	0.918296	

Setelah menghitung *information gain* dari masing-masing atribut, maka hasil yang terbesar akan menjadi simpul utama. Pada Tabel 3.2 ditemukan bahwa *information gain* tertinggi adalah U1 dan U4. Maka, dipilih salah satu dari U1 dan U4 untuk menjadi simpul utama. Dalam sistem ini, akan dipilih data pertama yaitu U1 sebagai simpul utama. Gambar 3.1 merupakan decision tree awal dengan U1 sebagai *root node*.



Gambar 3.1 Decision Tree dengan U1 Sebagai Root Utama

Cabang dari node U1 adalah 0 dan 1. Maka dilakukan perhitungan untuk mencari *information gain* selanjutnya. Tabel 3.3 adalah tabel yang akan digunakan untuk perhitungan cabang dengan U1 bernilai 0 (nol) sedangkan Tabel 3.4 digunakan untuk perhitungan cabang dengan U1 bernilai 1 (satu).

Tabel 3.3 Tabel Mata Kuliah Node U1 Cabang 0

Mata Kuliah	Kompetensi Utama Lulusan					Nilai (atribut target)
	U1	U2	U3	U4	U5	
MK 3	0	1	1	0	0	B
MK 7	0	0	0	0	0	B+

Tabel 3.4 Tabel Mata Kuliah Node U1 Cabang 1

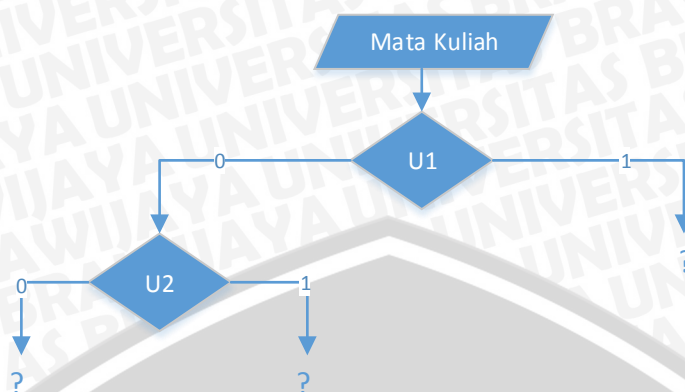
Mata Kuliah	Kompetensi Utama Lulusan					Nilai (atribut target)
	U1	U2	U3	U4	U5	
MK 1	1	0	0	1	1	A
MK 2	1	0	1	1	1	C+
MK 4	1	1	0	1	0	A
MK 5	1	0	0	1	0	C
MK 6	1	1	1	1	1	A

Untuk cabang pertama dari *root node* U1 bernilai 0, dilakukan perhitungan untuk mencari *information gain* tertinggi lagi. Tabel 3.5 menunjukkan hasil perhitungan *information gain* U2, U3, U4 dan U5 dengan *root node* U1 bernilai 0.

Tabel 3.5 Tabel Hasil Entropi dan Information Gain Atribut U2, U3, U4 dan U5 node U1 cabang 0

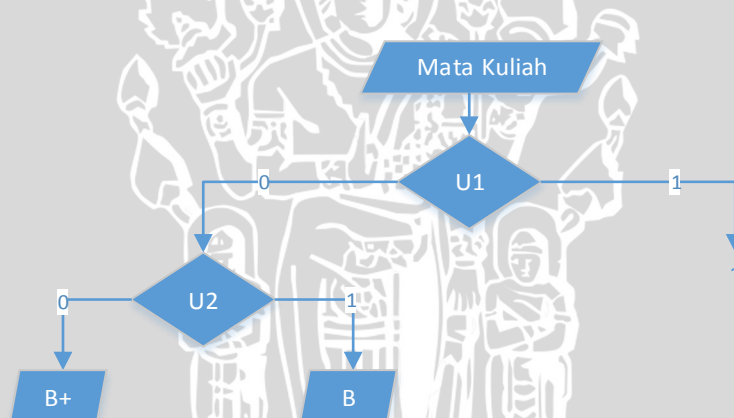
	Value	Entropi	Gain
U2	0	0	1
	1	0	
U3	0	0	1
	1	0	
U4	0	1	0
	1	0	
U5	0	1	0
	1	0	

Setelah menghitung *information gain* dari masing-masing atribut, maka atribut dengan *information gain* yang terbesar akan menjadi simpul selanjutnya. Pada Tabel 3.5 *information gain* tertinggi adalah U2 dan U3. Maka, kembali dipilih salah satu dari keduanya untuk menjadi *node* selanjutnya. Dalam contoh ini, U2 dipilih menjadi *node* selanjutnya. Gambar 3.2 merupakan *decision tree* dengan U2 sebagai *node* dari cabang U1 yang bernilai 0.



Gambar 3.2 Decision Tree dengan U2 Sebagai Node Cabang 0

Cabang dari node U2, adalah 0 dan 1. Pada Tabel 3.3 hanya terdapat dua data dimana U2 pada MK3 bernilai 1 sedangkan pada MK7 bernilai 0, maka *node* U2 ini menjadi cabang terakhir pada *decision tree* untuk cabang U1 yang bernilai nol. Gambar 3.3 merupakan *decision tree* dengan nilai B+ sebagai *node* terakhir dari cabang 0 dan nilai B sebagai *node* terakhir pada cabang 1.



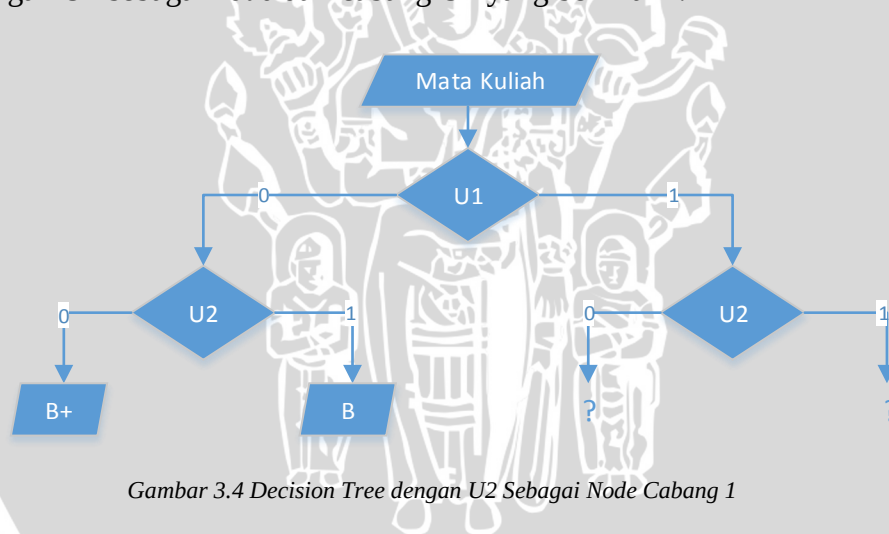
Gambar 3.3 Decision Tree dengan B dan B+ Sebagai Node Cabang dari U2

Untuk cabang kedua dari *root node* U1 yang bernilai 1, dilakukan perhitungan untuk mencari *information gain* tertinggi lagi. Tabel 3.6 Tabel Hasil Entropi dan Information Gain Atribut U2, U3, U4 dan U5 node U1 cabang 1 menunjukkan hasil perhitungan *information gain* U2, U3, U4 dan U5 dengan cabang U1 yang bernilai 1.

Tabel 3.6 Tabel Hasil Entropi dan Information Gain Atribut U2, U3, U4 dan U5 node U1 cabang 1

	Value	Entropi	Gain
U2	0	1.584963	0.419973
	1	0	
U3	0	0.918296	0.419973
	1	1	
U4	0	0	0
	1	1.370951	
U5	0	1	0.419973
	1	0.918296	

Setelah menghitung *information gain* dari masing-masing atribut, maka atribut dengan *information gain* yang terbesar akan menjadi simpul selanjutnya. Pada Tabel 3.6 *information gain* tertinggi adalah U2, U3, dan U5. Maka, salah satu dari ketiganya dipilih menjadi *node* selanjutnya. Gambar 3.4 merupakan *decision tree* dengan U2 sebagai *node* dari cabang U1 yang bernilai 1.



Gambar 3.4 Decision Tree dengan U2 Sebagai Node Cabang 1

Cabang dari node U2 adalah 0 dan 1. Selanjutnya, dilakukan perhitungan kembali untuk mencari *information gain* untuk memilih *node* selanjutnya. Tabel 3.7 adalah tabel yang akan digunakan untuk perhitungan cabang 0 sedangkan Tabel 3.8 digunakan untuk perhitungan cabang 1.

Tabel 3.7 Tabel Mata Kuliah Node U2 Cabang 0

Mata Kuliah	Kompetensi Utama Lulusan					Nilai (atribut target)
	U1	U2	U3	U4	U5	
MK 1	1	0	0	1	1	A
MK 2	1	0	1	1	1	C+
MK 5	1	0	0	1	0	C

Tabel 3.8 Tabel Mata Kuliah Node U2 Cabang 1

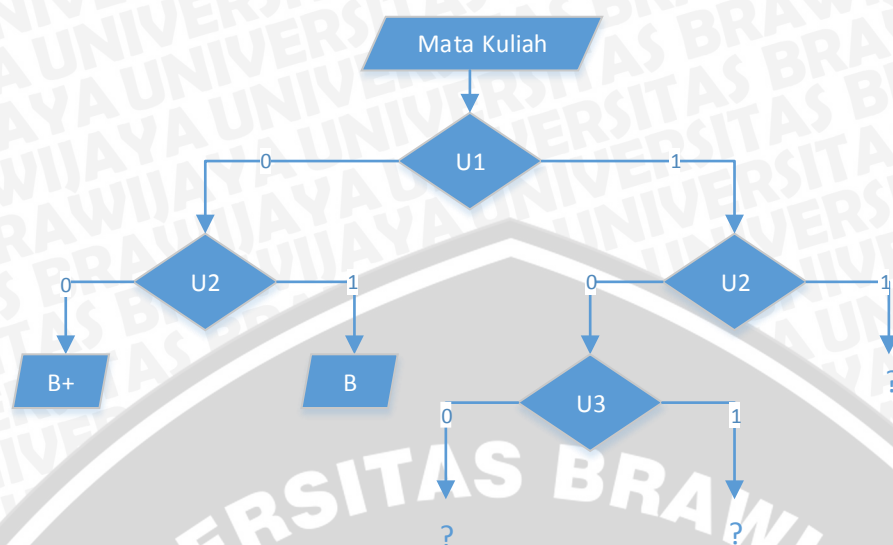
Mata Kuliah	Kompetensi Utama Lulusan					Nilai (atribut target)
	U1	U2	U3	U4	U5	
MK 4	1	1	0	1	0	A
MK 6	1	1	1	1	1	A

Untuk cabang pertama dari *node* U2 yang bernilai 0, dilakukan perhitungan untuk mencari *information gain* tertinggi dari atribut U3, U4, dan U5. Tabel 3.9 menunjukkan hasil perhitungan entropi dan *information gain* U3, U4 dan U5 dengan *node* U1 bernilai 1, dan *node* U2 bernilai 0.

Tabel 3.9 Tabel Hasil Entropi dan Information Gain Atribut U3, U4 dan U5 node U2 cabang 0

	Value	Entropi	Gain
U3	0	1	0.918296
	1	0	
U4	0	0	0
	1	1.584963	
U5	0	0	0.918296
	1	1	

Dari Tabel 3.9, ditemukan bahwa U3 dan U5 menghasilkan *information gain* yang sama besar. Maka, salah satu dari keduanya dipilih menjadi *node* selanjutnya. Dalam contoh ini, U3 dipilih menjadi *node* selanjutnya. Gambar 3.5 merupakan *decision tree* dengan U3 sebagai *node* selanjutnya.



Gambar 3.5 Decision Tree dengan U3 Sebagai Node Cabang 0

Cabang dari node U3 adalah 0 dan 1. Selanjutnya, kembali dilakukan perhitungan untuk mencari *information gain* dari atribut-atribut selanjutnya. Tabel 3.10 adalah tabel yang akan digunakan untuk perhitungan *information gain* dari cabang U3 yang bernilai 0 sedangkan Tabel 3.11 digunakan untuk perhitungan cabang U3 yang bernilai 1.

Tabel 3.10 Tabel Mata Kuliah Node U3 Cabang 0

Mata Kuliah	Kompetensi Utama Lulusan					Nilai (atribut target)
	U1	U2	U3	U4	U5	
MK 1	1	0	0	1	1	A
MK 5	1	0	0	1	0	C

Tabel 3.11 Tabel Mata Kuliah Node U3 Cabang 1

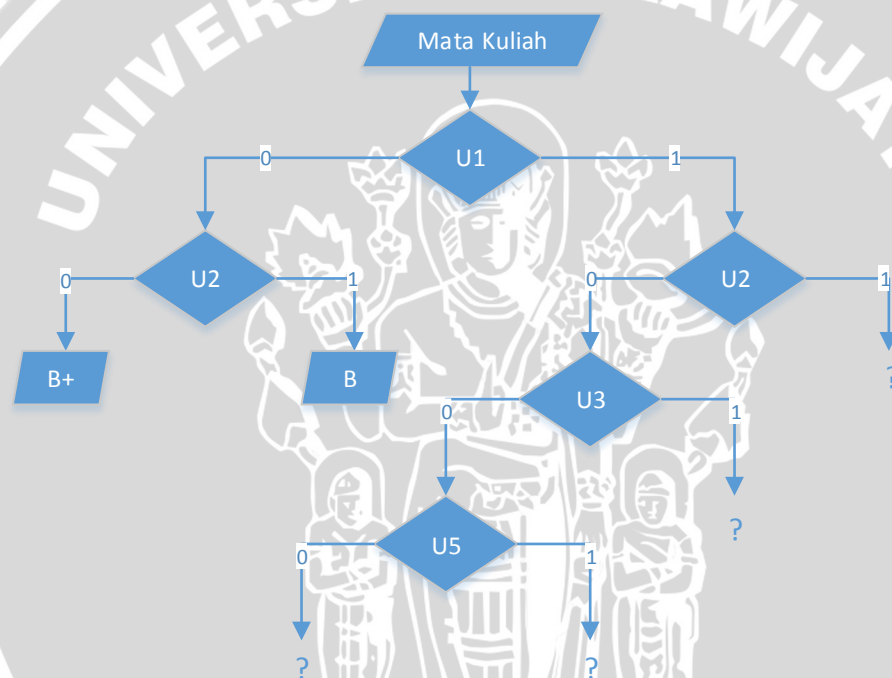
Mata Kuliah	Kompetensi Utama Lulusan					Nilai (atribut target)
	U1	U2	U3	U4	U5	
MK 2	1	0	1	1	1	C+

Untuk cabang dari node U3 bernilai 0, dilakukan perhitungan untuk mencari *information gain* tertinggi selanjutnya. Tabel 3.12 menunjukkan hasil perhitungan *information gain* U4 dan U5 dengan root utama U1 yakni 1, node cabang U2 yakni 0, dan node cabang U3 yakni 0.

Tabel 3.12 Tabel Hasil Entropi dan Information Gain Atribut U4 dan U5 node U3 cabang 0

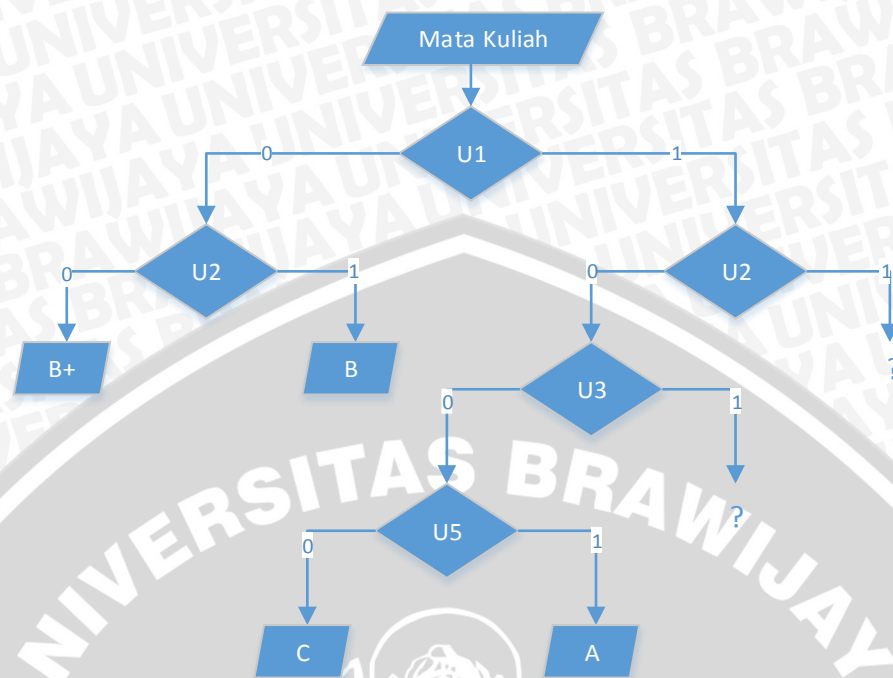
	Value	Entropi	Gain
U4	0	0	0
	1	1	
U5	0	0	0,75
	1	0,5	

Pada Tabel 3.12 ditemukan bahwa *information gain* tertinggi adalah U5. Maka, U5 dipilih menjadi *node* selanjutnya. Gambar 3.6 merupakan decision tree dengan U5 sebagai node dari cabang 0.



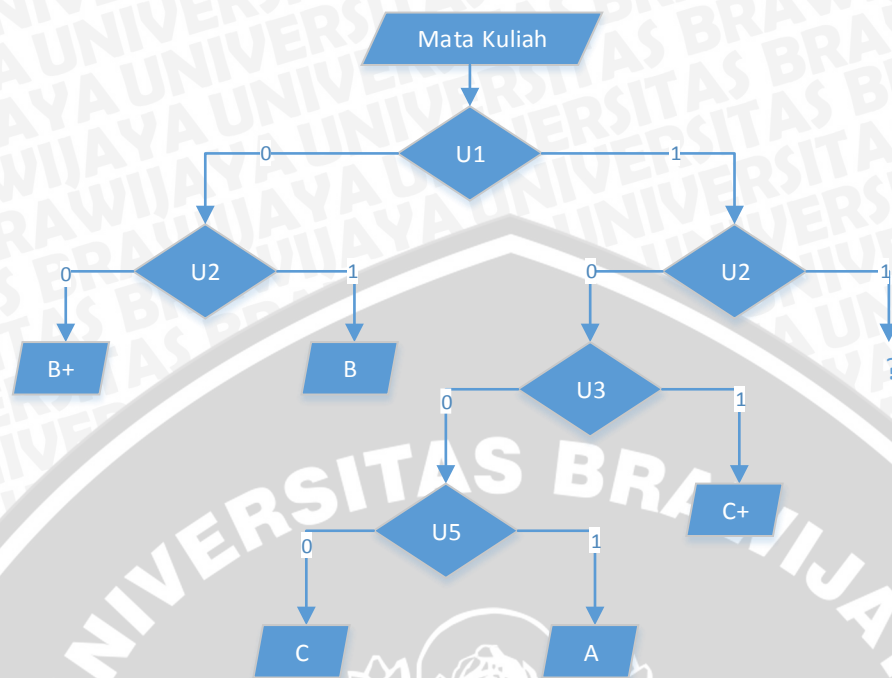
Gambar 3.6 Decision Tree dengan U5 Sebagai Node Cabang 0

Pada Tabel 3.10 hanya terdapat dua data dimana U5 pada MK1 bernilai 1 sedangkan pada MK5 bernilai 0. Dengan demikian, maka U5 akan menjadi *node* terakhir. *Node* ini berakhir dengan *leaf* dengan nilai C pada cabang 0 dan nilai A pada cabang 1. Gambar 3.7 merupakan ilustrasi *tree* yang terbentuk.



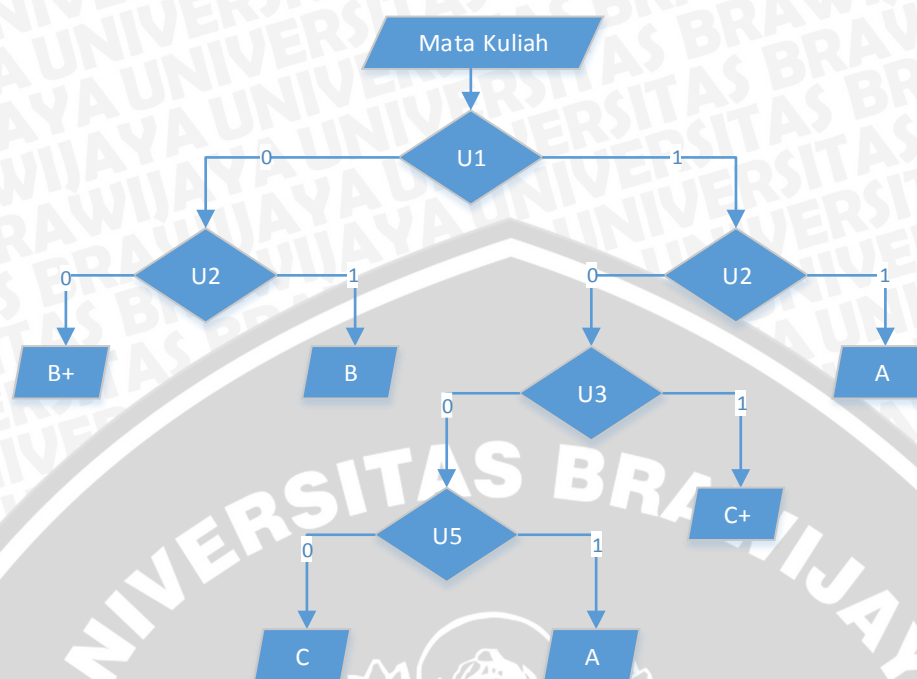
Gambar 3.7 Decision Tree bernilai C dan A Sebagai leaf dari U5

Setelah tidak diperlukan untuk membentuk cabang lanjutan dari U5, iterasi akan kembali ke cabang kedua dari *node* U3 yang bernilai 1 untuk menentukan cabang selanjutnya dari U3 yang bernilai 1. Pada Tabel 3.11 hanya terdapat satu data yaitu U3 pada MK1 bernilai 1. Dengan demikian, maka akan terbentuk *branch* dengan nilai C+ yang akan menjadi *leaf* dari *node* U3 yang bernilai 1. Gambar 3.8 merupakan perkembangan *decision tree* yang terbentuk.

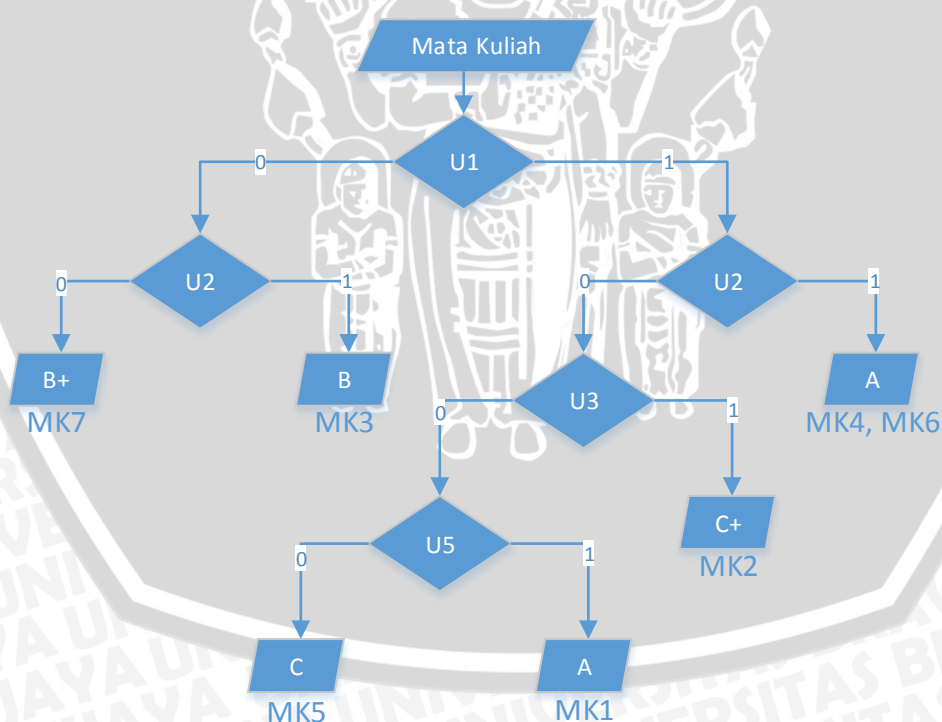


Gambar 3.8 Decision Tree dengan C+ Sebagai Node dari Cabang 1 Node U3

Tahap selanjutnya adalah menentukan turunan dari *node* U2 yang bernilai 1. Pada Tabel 3.8 terdapat dua data dengan atribut target yang sama yakni MK4 dan MK6 yang keduanya memiliki nilai A. Dengan demikian, akan terbentuk *branch* nilai A yang menjadi *leaf* pada cabang 1 *node* U2. Gambar 3.9 menunjukkan perkembangan *decision tree* yang terbentuk. Dapat diamati bahwa *tree* yang terbentuk sudah lengkap, dengan demikian pembentukan *decision tree* sudah lengkap. Gambar 3.10 menunjukkan *decision tree* final beserta data mata kuliah yang ada pada kelas tersebut.



Gambar 3.9 Decision Tree dengan A Sebagai Node dari Cabang 1 Node U2



Gambar 3.10 Decision Tree Final

3.4.1.3 Pembentukan Rules

Setelah terbentuk *decision tree*, tahapan selanjutnya adalah membentuk *rules* yang didapatkan dari hasil pembentukan *tree*. *Rules* didapatkan dengan melihat nilai dari *branch* paling awal hingga *target attribute*. Sebagai contoh, dari *tree* pada Gambar 3.10 akan didapatkan *rules* sebagai berikut:

- IF $U1 = 0$ AND $U2 = 0$ THEN nilai = 'B+'
- IF $U1 = 0$ AND $U2 = 1$ THEN nilai = 'B'
- IF $U1 = 1$ AND $U2 = 0$ AND $U3 = 0$ AND $U5 = 0$ THEN nilai = 'C'
- IF $U1 = 1$ AND $U2 = 0$ AND $U3 = 0$ AND $U5 = 1$ THEN nilai = 'A'
- IF $U1 = 1$ AND $U2 = 0$ AND $U3 = 1$ THEN nilai = 'C+'
- IF $U1 = 1$ AND $U2 = 1$ THEN nilai = 'A'

Dari *rules* yang diperoleh ini, akan diberikan rekomendasi pilihan mata kuliah yang lain dengan memprioritaskan mata kuliah yang sekiranya dapat memberikan nilai yang baik kepada pengguna.

3.4.2 Collaborative Filtering

Collaborative filtering yang dilakukan pada penelitian ini bertujuan untuk memberikan rekomendasi mata kuliah kepada pengguna berdasarkan pilihan mata kuliah dari pengguna lain sebelumnya. *Filtering* yang dilakukan melibatkan data dari banyak pengguna secara kolaboratif. Pada akhirnya, pengguna akan mendapatkan rekomendasi *item* yang dipilih dari pengguna lain yang mempunyai kesamaan *preference*. *Preference* yang dimaksud adalah apakah seorang pengguna (mahasiswa) memprogram suatu mata kuliah tertentu.

Collaborative filtering pada penelitian ini melibatkan pengguna aktif sebagai penerima rekomendasi, pengguna-pengguna sebelumnya sebagai objek *learning*, mata kuliah pilihan yang dipandang sebagai *item*, dan *user preference* yang menjelaskan apakah seorang pengguna memprogram mata kuliah tertentu. *User preference* inilah yang dipandang sebagai rating.

Tahapan-tahapan yang dilakukan pada *collaborative filtering* adalah pembentukan matriks, perhitungan nilai *similarity*, dan perhitungan nilai prediksi.

3.4.2.1 Pembentukan Matriks

Informasi yang dibutuhkan dalam memberikan rekomendasi dengan *collaborative filtering* adalah informasi *rating* berupa angka 1 dan 0 yang menunjukkan apakah mata kuliah tersebut diprogram atau tidak diprogram oleh mahasiswa tertentu. Informasi ini dibutuhkan untuk menghitung kemiripan antar suatu mata kuliah.

Misalkan diketahui matriks hubungan antara mahasiswa dan mata kuliah sebagai berikut:

Tabel 3.13 Contoh tabel mata kuliah yang telah diprogram oleh para mahasiswa

Mahasiswa	Mata Kuliah	Diprogram dalam KRS
11111112	MK 1	1
11111112	MK 2	1
11111112	MK 3	1
11111112	MK 4	0
11111112	MK 5	0
11111113	MK 1	1
11111113	MK 2	0
11111113	MK 3	1
11111113	MK 4	1
11111113	MK 5	1
11111114	MK 1	1
11111114	MK 2	0
11111114	MK 3	1
11111114	MK 4	1
11111114	MK 5	0
11111115	MK 1	1
11111115	MK 2	0
11111115	MK 3	1
11111115	MK 4	1
11111115	MK 5	0

maka informasi tersebut dikonversi menjadi tabel sebagai berikut:

Tabel 3.14 Matriks relasi mahasiswa - nilai MK

Mahasiswa	Nilai				
	MK 1	MK 2	MK 3	MK 4	MK 5
11111112	1	1	1	0	0
11111113	1	0	1	1	1
11111114	1	0	1	1	0
11111115	1	0	1	1	0

Matriks pada Tabel 3.14 nantinya akan diolah dalam menghitung nilai *similarity* antar mata kuliah dan prediksi nilai yang akan diberikan.

Tabel 3.15 Tambahan pengguna aktif X yang akan diberi rekomendasi

Mahasiswa	Nilai				
	MK 1	MK 2	MK 3	MK 4	MK 5
11111112	1	1	1	0	0
11111113	1	0	1	1	1
11111114	1	0	1	1	0
11111115	1	0	1	1	0
X	1	0	1	?	?

3.4.2.2 User-based Collaborative Filtering

Tahapan-tahapan yang dilakukan pada *user-based collaborative filtering* adalah pembentukan matriks, perhitungan nilai *similarity*, dan perhitungan nilai prediksi.

3.4.2.2.1 Perhitungan Nilai Similarity

Tahap pertama yang dilakukan sistem dalam memberikan rekomendasi dengan *user-based collaborative filtering* adalah menghitung nilai *similarity* antar user dengan menggunakan persamaan (2.3).

Untuk mempermudah visualisasi perhitungan, perlu dibuat tabel nilai *mean adjusted score* dari Tabel 3.15 yang sudah di-*tranpose* sebagai berikut:

Tabel 3.16 Mean adjusted scores, user-based collaborative filtering

Mata Kuliah	$(R_{u,i} - \bar{R}_u)$				
	11111112	11111113	11111114	11111115	X
MK 1	0.4	0.2	0.4	0.4	0.6
MK 2	0.4	-0.8	-0.6	-0.6	-0.4
MK 3	0.4	0.2	0.4	0.4	0.6
MK 4	-0.6	0.2	0.4	0.4	n/a
MK 5	-0.6	0.2	-0.6	-0.6	n/a

Selanjutnya, dilakukan perhitungan nilai *similarity* antar *user* dengan menggunakan persamaan (2.3). Sebagai contoh, untuk menghitung *similarity*(X, 11111113) dari pengguna baru X dan 11111113, dilakukan perhitungan sebagai berikut:

$$\text{sim}(X, 11111113) = \frac{0.6 \cdot 0.2 + (-0.4) \cdot (-0.8) + 0.6 \cdot 0.2}{\sqrt{0.6^2 + (-0.4)^2 + 0.6^2} \sqrt{0.2^2 + (-0.8)^2 + 0.2^2}} = 0.703526471$$

Secara ringkas, hasil perhitungan *similarity* dari kombinasi *user* X dan *user* lainnya adalah sebagai berikut:

$$\text{sim}(X, 11111112) = 0.492365964$$

$$\text{sim}(X, 11111113) = 0.703526471$$

$$\text{sim}(X, 11111114) = 0.930757842$$

$$\text{sim}(X, 11111115) = 0.930757842$$

Dari perhitungan tersebut, ditemukan bahwa nilai *user* 11111114 dan 11111115 memiliki *similarity* paling besar terhadap *user* X.

3.4.2.2.2 Perhitungan Nilai Prediksi

Setelah didapatkan nilai *similarity*, dilakukan perhitungan nilai prediksi kecenderungan apakah suatu mata kuliah akan diprogram oleh mahasiswa.

Sebagai perbandingan, dilakukan penghitungan prediksi nilai rating dengan menggunakan persamaan (2.4) dan (2.5) dan didapatkan hasil sebagai berikut:

- Hasil perhitungan dengan persamaan *weighted sum* (2.4):
 - $P(X, MK\ 4) = 0.838959686$
 - $P(X, MK\ 5) = 0.230105515$

- Hasil perhitungan dengan persamaan (2.5):

- $P(X, MK\ 4) = 0.859605249$

- $P(X, MK\ 5) = 0.250751079$

Dari kedua perhitungan tersebut, ditemukan bahwa nilai prediksi kecenderungan pemilihan mata kuliah MK 4 oleh pengguna X secara konsisten selalu lebih besar daripada pemilihan mata kuliah MK 5 sehingga MK 4 lebih direkomendasikan daripada MK 5.

3.4.2.3 Item-based Collaborative Filtering

Tahapan-tahapan yang dilakukan pada *item-based collaborative filtering* adalah pembentukan matriks, perhitungan nilai *similarity*, dan perhitungan nilai prediksi.

3.4.2.3.1 Perhitungan Nilai Similarity

Pada bagian ini, sistem menghitung kemiripan antar *item* mata kuliah. Kemiripan mata kuliah dikalkulasi dari data mata kuliah yang diprogram oleh mahasiswa seperti pada tabel 3.2 dan 3.3 yang diasumsikan sebagai *rating* dari *item* mata kuliah.

Persamaan yang digunakan untuk menghitung *similarity* antara item i dan j adalah persamaan (2.7).

Item-based collaborative filtering dibangun dengan menghitung *similarity* antar *item* mata kuliah dan prediksi apakah *item* yang ditawarkan akan diprogram oleh mahasiswa. Sebagai contoh, misal pengguna X sudah pernah memprogram mata kuliah MK 1, MK 2, dan MK 3. Maka, mata kuliah yang mungkin untuk direkomendasikan kepada *user* tersebut adalah MK 4 dan MK 5. Contoh ini dapat dilihat di Tabel 3.15.

Untuk mempermudah visualisasi perhitungan, langkah pertama yang dilakukan adalah menghitung *mean adjusted score* untuk masing-masing elemen nilai pada Tabel 3.15. Hasil perhitungan *mean adjusted score* adalah sebagai berikut:

Tabel 3.17 Mean adjusted scores, item-based collaborative filtering

Mahasiswa	$(R_{u,i} - \bar{R}_u)$				
	MK 1	MK 2	MK 3	MK 4	MK 5
11111112	0.4	0.4	0.4	-0.6	-0.6
11111113	0.2	-0.8	0.2	0.2	0.2
11111114	0.4	-0.6	0.4	0.4	-0.6
11111115	0.4	-0.6	0.4	0.4	-0.6
X	0.6	-0.4	0.6	n/a	n/a

Penghitungan nilai *similarity* dilakukan untuk setiap pasangan mata kuliah yang mungkin dari mata kuliah yang akan ditawarkan. Sebagai contoh untuk kasus pada Tabel 3.15, maka penghitungan nilai *similarity* akan dilakukan pada pasangan mata kuliah MK 4 – MK 1, MK 4 – MK 2, MK 4 – MK 3, MK 5 – MK 1, MK 5 – MK 2, dan MK 5 – MK 3. Contoh perhitungan nilai *similarity* untuk pasangan MK 4 – MK 1 dengan memanfaatkan *mean adjusted scores* seperti pada persamaan (2.7) adalah sebagai berikut:

$$\text{sim}(\text{MK 4}, \text{MK 1}) = \frac{(-0.6) \cdot 0.4 + 0.2 \cdot 0.2 + 0.4 \cdot 0.4 + 0.4 \cdot 0.4}{\sqrt{(-0.6)^2 + 0.2^2 + 0.4^2 + 0.4^2} \sqrt{0.4^2 + 0.2^2 + 0.4^2 + 0.4^2}} = 0.196116135$$

Secara ringkas, hasil perhitungan antar mata kuliah adalah sebagai berikut:

$$\text{sim}(\text{MK 4}, \text{MK 1}) = 0.196116135$$

$$\text{sim}(\text{MK 4}, \text{MK 2}) = -0.841191024$$

$$\text{sim}(\text{MK 4}, \text{MK 3}) = 0.196116135$$

$$\text{sim}(\text{MK 5}, \text{MK 1}) = -0.891042111$$

$$\text{sim}(\text{MK 5}, \text{MK 2}) = 0.245255736$$

$$\text{sim}(\text{MK 5}, \text{MK 3}) = -0.891042111$$

Dari perhitungan *similarity*, ditemukan bahwa nilai *similarity* paling besar adalah MK 5 dengan MK 2.

3.4.2.3.2 Perhitungan Nilai Prediksi

Tahapan selanjutnya adalah menghitung prediksi nilai kecenderungan suatu mata kuliah akan diprogram oleh mahasiswa. Untuk menghitung prediksi nilai, digunakan persamaan (2.8) serta (2.9). Perhitungan prediksi hanya melibatkan *item-item* yang memiliki similaritas bernilai lebih dari atau sama dengan nol.

Sebagai contoh, untuk menghitung prediksi kecenderungan pemilihan mata kuliah MK 4 oleh pengguna X dengan menggunakan *weighted sum* (2.8), maka dilakukan perhitungan sebagai berikut:

$$P_{X,MK\ 4} = \frac{sim(MK\ 4,MK\ 1)*R(X,MK\ 1) + sim(MK\ 4,MK\ 3)*R(X,MK\ 3)}{sim(MK\ 4,MK\ 1) + sim(MK\ 4,MK\ 3)}$$

$$P_{X,MK\ 4} = \frac{0.196116135*1 + 0.196116135*1}{0.196116135 + 0.196116135}$$

$$P_{X,MK\ 4} = 1$$

- Hasil perhitungan dengan persamaan *weighted sum* (2.8):
 - $P(X, MK\ 4) = 1$
 - $P(X, MK\ 5) = 0$
- Hasil perhitungan dengan persamaan (2.9):
 - $P(X, MK\ 4) = 0.886399406$
 - $P(X, MK\ 5) = 0.225805169$

Dari kedua perhitungan tersebut, ditemukan bahwa nilai prediksi kecenderungan pemilihan mata kuliah MK 4 oleh pengguna X secara konsisten selalu lebih besar daripada pemilihan mata kuliah MK 5 sehingga MK 4 lebih direkomendasikan daripada MK 5.

3.5 Perancangan Antarmuka

Pada penelitian ini, sistem akan dibangun pada aplikasi yang sudah ada, dengan memberikan tambahan satu halaman untuk menampilkan daftar mata kuliah yang direkomendasikan. Gambar 3.11 menunjukkan rancangan antarmuka rekomendasi mata kuliah.

Pada rancangan Gambar 3.11, sistem menampilkan mata kuliah yang direkomendasikan dalam bentuk tabular dengan kolom kode mata kuliah, nama mata kuliah, dan satu kolom *action* untuk mengecek ketersediaan kelas dari mata kuliah yang dimaksud. Setelah pengguna mengklik tombol “Cari Kelas”, maka pengguna akan diarahkan ke halaman pencarian kelas mata kuliah dengan mengirimkan parameter kode mata kuliah yang dipilih.

UNIVERSITAS BRAWIJAYA
SIAM.UB.ONLINE
SISTEM INFORMASI AKADEMIK MAHASISWA

105060804111008
Aditya Fitri Hananta Putra
Jenjang / Fakultas: S1 / Prog. T. Informasi & Ilmu Komputer
Jurusan: Teknik Informatika
Seleksi: Reguler
Nomor Ujian: 11111111
Status: Aktif

KARTU RENCANA STUDI
KARTU HASIL STUDI
JADWAL KULIAH
JADWAL UJIAN
ABSENSI
REKAP HASIL STUDI

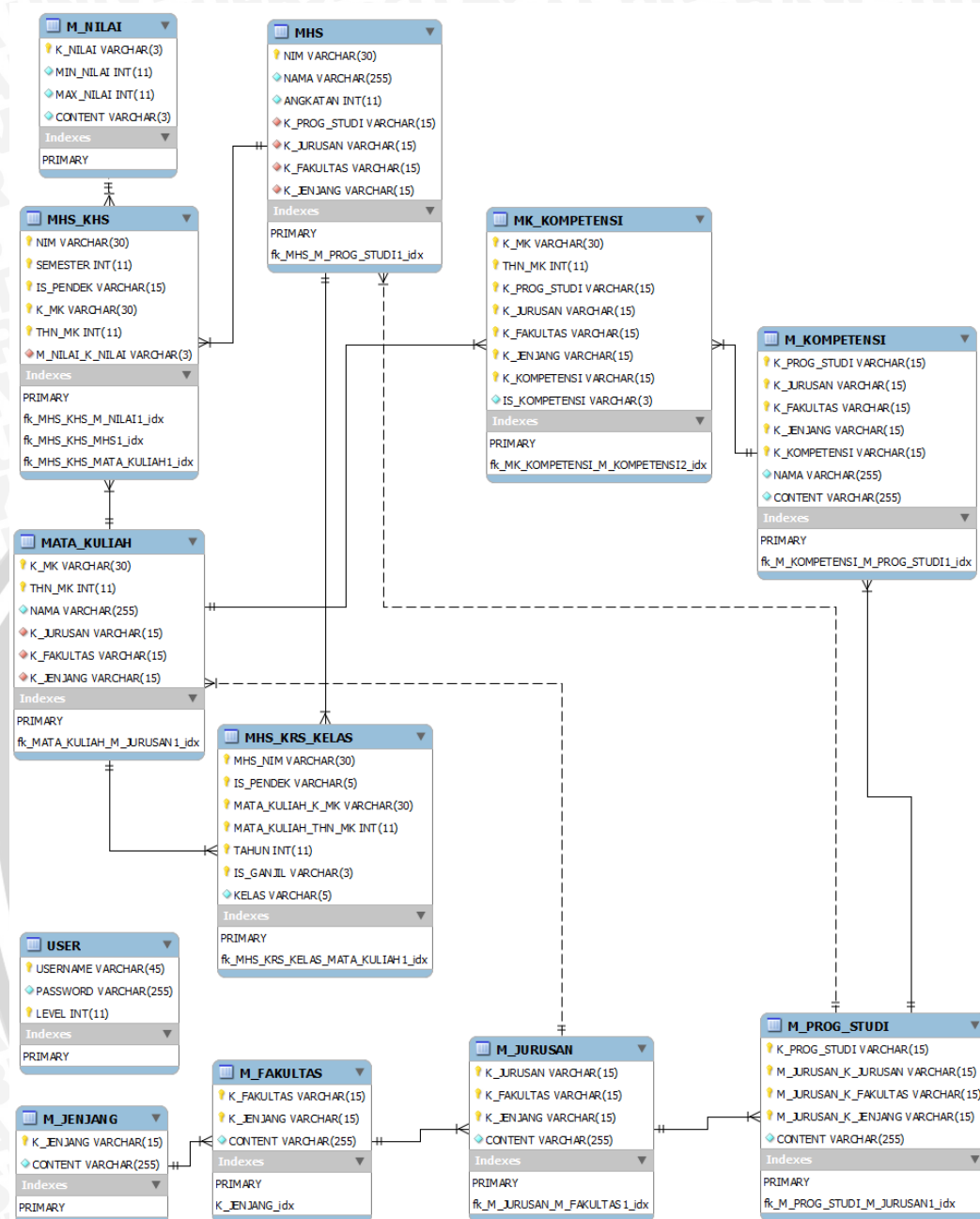
Rekomendasi Mata Kuliah Pilihan: Genap 2014 / 2015

KODE MK	NAMA MK	
MK1	Mata Kuliah Satu	Cari Kelas
MK2	Mata Kuliah Dua	Cari Kelas
MK3	Mata Kuliah Tiga	Cari Kelas
MK4	Mata Kuliah Empat	Cari Kelas
MK5	Mata Kuliah Lima	Cari Kelas
MK6	Mata Kuliah Enam	Cari Kelas
MK7	Mata Kuliah Tujuh	Cari Kelas
MK8	Mata Kuliah Delapan	Cari Kelas
MK9	Mata Kuliah Sembilan	Cari Kelas
MK10	Mata Kuliah Sepuluh	Cari Kelas

Gambar 3.11 Rancangan antarmuka halaman rekomendasi mata kuliah

3.6 Perancangan Manajemen Data

Pada bagian ini akan dirancang tentang manajemen data yang digunakan sebagai tempat penyimpanan data dari sistem rekomendasi yang dibangun. Penyimpanan data dilakukan dengan *Database Management System* (DBMS) MySQL. Perancangan tabel basis data ditampilkan dalam Gambar 3.12.



Gambar 3.12 Perancangan tabel data

Pada perancangan tabel data Gambar 3.12 mendeskripsikan detail dari tabel-tabel yang digunakan dalam pengembangan sistem rekomendasi ini. Perancangan ini menjelaskan *field* dari tabel tersebut, tipe data, beserta ukuran dari

masing-masing *field*. Berikut ini akan dideskripsikan detail tabel-tabel yang dibutuhkan dalam pengembangan sistem rekomendasi ini.

1. Tabel USER

Tabel USER menyimpan data *user* / pengguna yang dapat *log in* ke sistem.

Tabel 3.18 Tabel USER

Field	Tipe data dan ukuran	Keterangan
USERNAME	varchar(45)	Nama pengguna
PASSWORD	varchar(255)	Kata kunci pengguna
LEVEL	int(11)	Level pengguna

2. Tabel M_JENJANG

Tabel M_JENJANG menyimpan data master jenjang akademik perkuliahan, misalnya jenjang D3, S1, S2, dan sebagainya.

Tabel 3.19 Tabel M_JENJANG

Field	Tipe data dan ukuran	Keterangan
K_JENJANG	varchar(15)	Kode jenjang
CONTENT	varchar(255)	Nama jenjang

3. Tabel M_FAKULTAS

Tabel M_FAKULTAS menyimpan data master fakultas serta berelasi dengan tabel M_JENJANG.

Tabel 3.20 Tabel M_JENJANG

Field	Tipe data dan ukuran	Keterangan
K_FAKULTAS	varchar(15)	Kode fakultas

K_JENJANG	varchar(15)	Kode jenjang
CONTENT	varchar(255)	Nama fakultas

4. Tabel M_JURUSAN

Tabel M_JURUSAN menyimpan data master jurusan serta berelasi dengan tabel M_FAKULTAS.

Tabel 3.21 Tabel M_JURUSAN

<i>Field</i>	Tipe data dan ukuran	Keterangan
K_JURUSAN	varchar(15)	Kode jurusan
K_FAKULTAS	varchar(15)	Kode fakultas
K_JENJANG	varchar(15)	Kode jenjang
CONTENT	varchar(255)	Nama jurusan

5. Tabel M_PROG_STUDI

Tabel M_PROG_STUDI menyimpan data master program studi serta berelasi dengan tabel M_JURUSAN.

Tabel 3.22 Tabel M_PROG_STUDI

<i>Field</i>	Tipe data dan ukuran	Keterangan
--------------	---------------------------------	-------------------

K_PROG_STUDI	varchar(15)	Kode program studi
K_JURUSAN	varchar(15)	Kode jurusan
K_FAKULTAS	varchar(15)	Kode fakultas
K_JENJANG	varchar(15)	Kode jenjang
CONTENT	varchar(255)	Nama program studi

6. Tabel M_NILAI

Tabel M_NILAI menyimpan data master nilai huruf yang terdapat pada sistem beserta *range* nilai angka untuk nilai tersebut misalnya nilai A dengan *range* nilai 80-100, B+ dengan *range* nilai 70-80, dan sebagainya.

Tabel 3.23 Tabel M_NILAI

Field	Tipe data dan ukuran	Keterangan
K_NILAI	varchar(3)	Kode nilai huruf
MIN_NILAI	int(11)	Batasan nilai angka minimal
MAX_NILAI	int(11)	Batasan nilai angka maksimal
CONTENT	varchar(3)	Nama nilai huruf

7. Tabel M_KOMPETENSI

Tabel M_KOMPETENSI menyimpan data master kompetensi lulusan dari yang berlaku pada suatu program studi, jurusan, fakultas, dan jenjang tertentu.

Tabel 3.24 Tabel M_KOMPETENSI

Field	Tipe data dan ukuran	Keterangan
--------------	-----------------------------	-------------------

K_PROG_STUDI	varchar(15)	Kode program studi
K_JURUSAN	varchar(15)	Kode jurusan
K_FAKULTAS	varchar(15)	Kode fakultas
K_JENJANG	varchar(15)	Kode jenjang
K_KOMPETENSI	varchar(15)	Kode kompetensi lulusan
NAMA	varchar(255)	Nama kompetensi lulusan
CONTENT	varchar(255)	Isi kompetensi lulusan

8. Tabel MHS

Tabel MHS menyimpan data umum mahasiswa. Tabel ini mereferensi tabel M_PROG_STUDI.

Tabel 3.25 Tabel MHS

Field	Tipe data dan ukuran	Keterangan
NIM	varchar(30)	Nomor induk mahasiswa
NAMA	varchar(255)	Nama mahasiswa
ANGKATAN	int(11)	Angkatan mahasiswa
K_PROG_STUDI	varchar(15)	Kode program studi
K_JURUSAN	varchar(15)	Kode jurusan
K_FAKULTAS	varchar(15)	Kode fakultas
K_JENJANG	varchar(15)	Kode jenjang

9. Tabel MATA_KULIAH

Tabel MATA_KULIAH menyimpan daftar mata kuliah yang dimiliki oleh suatu jurusan, fakultas, dan jenjang tertentu.

Tabel 3.26 Tabel MATA_KULIAH

Field	Tipe data dan ukuran	Keterangan
--------------	-----------------------------	-------------------

K_MK	varchar(25)	Kode mata kuliah
THN_MK	int(11)	Tahun kurikulum mata kuliah
NAMA	varchar(255)	Nama mata kuliah
K_JURUSAN	varchar(15)	Kode jurusan
K_FAKULTAS	varchar(15)	Kode fakultas
K_JENJANG	varchar(15)	Kode jenjang

10. Tabel MK_KOMPETENSI

Tabel MK_KOMPETENSI menyimpan data hubungan antara mata kuliah dan kompetensi-kompetensinya yang berlaku pada suatu program studi, jurusan, fakultas, dan jenjang tertentu.

Tabel 3.27 Tabel MK_KOMPETENSI

Field	Tipe data dan ukuran	Keterangan
K_MK	varchar(15)	Kode mata kuliah
THN_MK	int(11)	Tahun kurikulum mata kuliah
K_PROG_STUDI	varchar(15)	Kode program studi
K_JURUSAN	varchar(15)	Kode jurusan
K_FAKULTAS	varchar(15)	Kode fakultas
K_JENJANG	varchar(15)	Kode jenjang
K_KOMPETENSI	varchar(15)	Kode kompetensi lulusan
IS_KOMPETENSI	varchar(3)	

11. Tabel MHS_KRS_KELAS

Tabel MHS_KRS_KELAS menyimpan daftar mata kuliah yang ditempuh oleh mahasiswa tertentu pada periode akademik tertentu.

Tabel 3.28 Tabel MHS_KRS_KELAS

Field	Tipe data	Keterangan
--------------	------------------	-------------------

	dan ukuran	
MHS_NIM	varchar(30)	Nomor induk mahasiswa
IS_PENDEK	varchar(5)	<i>Flag</i> semester pendek / normal
MATA_KULIAH_ K_MK	varchar(25)	Kode mata kuliah
MATA_KULIAH_ THN_MK	int(11)	Tahun kurikulum mata kuliah
TAHUN	int(11)	Tahun akademik
IS_GANJIL	varchar(3)	<i>Flag</i> semester ganjil / genap
KELAS	varchar(5)	Kelas perkuliahan

12. Tabel MHS_KHS

Tabel MHS_KHS menyimpan data hasil studi suatu mata kuliah yang ditempuh oleh mahasiswa pada semester tertentu.

Tabel 3.29 Tabel MHS_KHS

Field	Tipe data dan ukuran	Keterangan
NIM	varchar(30)	Nomor induk mahasiswa
SEMESTER	int(11)	Semester
IS_PENDEK	varchar(15)	<i>Flag</i> semester pendek / normal
K_MK	varchar(25)	Kode mata kuliah
THN_MK	int(11)	Tahun kurikulum mata kuliah
M_NILAI_K_NIL AI	varchar(3)	Kode nilai

3.7 Perancangan Pengujian

Setelah sistem selesai dibuat, langkah selanjutnya adalah melakukan pengujian terhadap sistem yang dibangun. Pengujian dilakukan untuk mengetahui

tingkat akurasi dari sistem yang dibangun. Tingkat akurasi sistem dilihat dari nilai *MAE* yang dihasilkan.

Data yang digunakan pada pengujian penelitian ini adalah data dari Universitas Brawijaya khususnya pada Program Studi Informatika periode akademik 2010 hingga 2015 dari angkatan 2010 hingga 2012.

Pengujian dilakukan dengan melihat pengaruh jumlah data latih yang diasumsikan sebagai jumlah semester yang telah ditempuh oleh mahasiswa tertentu terhadap tingkat akurasi dari sistem yang dibangun. Pengujian dilakukan terhadap masing-masing teknik yang digunakan.

Pada umumnya, sistem rekomendasi diuji dengan menghitung nilai *MAE*. Namun karena tidak terdapatnya data urutan *ranking* dari *preference* mahasiswa dalam pemilihan mata kuliah, maka pengujian dilakukan dengan membandingkan banyaknya hasil rekomendasi yang tepat terhadap keseluruhan data uji seperti yang ditunjukkan pada persamaan (2.10).

Tabel 3.30 Contoh mata kuliah yang ditempuh mahasiswa

NIM	Mata Kuliah Semester							
	1	2	3	4	5	6	7	8
11111111	MK1	MK4	MK7	MK21	MK24	MK26	MK29	MK9
	MK2	MK5	MK8	MK22	MK25	MK27	MK30	MK31
	MK3	MK6	MK10	-	MK23	MK28	-	-
11111112	MK1	MK4	MK7	MK11	MK14	MK17	MK19	MK10
	MK2	MK5	MK8	MK12	MK15	MK18	MK20	MK31
	MK3	MK6	MK9	MK13	MK16	-	-	MK32

Tabel 3.31 Contoh hasil rekomendasi mata kuliah

NIM	Rekomendasi Mata Kuliah Semester				
	4	5	6	7	8
11111111	MK13	MK24	MK26	MK29	MK9
	MK22	MK14	MK15	MK30	MK31
	-	MK28	MK23	-	-
11111112	MK11	MK14	MK19	MK22	MK10
	MK12	MK15	MK18	MK20	MK31
	MK32	MK16	-	-	MK22

Sebagai contoh, Tabel 3.30 menampilkan data aktual MK yang ditempuh oleh beberapa mahasiswa dan Tabel 3.31 menampilkan hasil rekomendasi yang didapatkan mahasiswa-mahasiswa tersebut. Pada semester keempat, terdapat tiga mata kuliah hasil rekomendasi yang cocok terhadap lima data aktual KHS yang ditempuh oleh mahasiswa pada semester yang sama, yaitu MK 22, MK 11, dan MK12. Untuk menghitung akurasi hasil rekomendasi pada semester keempat terhadap semester yang sama, dilakukan perhitungan dengan persamaan (2.10) sebagai berikut:

$$Akurasi = \frac{3}{5}$$

$$Akurasi = 0.6 = 60 \%$$

Pengujian kedua dilakukan dengan menghitung kecocokan hasil rekomendasi terhadap seluruh data aktual tanpa memperhatikan kesesuaian semesternya. Sebagai contoh pada semester keempat, terdapat empat mata kuliah hasil rekomendasi terhadap data aktual yang ditempuh oleh mahasiswa pada semester yang sama, yaitu MK22, MK11, MK12, dan MK32. Untuk menghitung akurasi hasil rekomendasi pada semester keempat pada semester yang tidak saling berkesesuaian, dilakukan perhitungan dengan persamaan (2.10) sebagai berikut:

$$Akurasi = \frac{4}{5}$$

$$Akurasi = 0.8 = 80 \%$$

BAB IV

IMPLEMENTASI

Pada bab ini menjelaskan tentang implementasi dari sistem yang telah dirancang pada bab metodologi penelitian dan perancangan. Pada bab implementasi ini tentang bagaimana spesifikasi dari sistem, batasan-batasan implementasi, implementasi antarmuka dan sistem yang telah dirancang pada bab sebelumnya.

4.1 Spesifikasi Sistem

Spesifikasi sistem ini didasarkan dari hasil analisis terhadap kebutuhan dan perancangan sistem yang telah dijabarkan dalam bab metode penelitian dan perancangan. Tahap ini bertujuan agar sistem dapat bekerja sesuai dengan kebutuhan sistem. Sistem rekomendasi ini dikembangkan dalam lingkungan implementasi berupa lingkungan perangkat keras dan lingkungan perangkat lunak.

4.1.1 Spesifikasi Perangkat Keras

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

1. Prosesor : Intel(R) Xeon(R) CPU E5-2630L 0 @ 2.00GHz
2. Memori (RAM) : 512 MB
3. Harddisk : 20GB SSD

4.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan sistem rekomendasi ini adalah sebagai berikut:

1. Sistem operasi : Ubuntu Server 14.04 Trusty Tahr LTS 64 bit
2. Bahasa pemrograman : PHP
3. Tool pemrograman : Aptana Studio 3
4. Web server : Apache 2.4.7 (Ubuntu)
5. DBMS : MySQL Ver 14.14 Distrib 5.5.38
6. Tool DBMS : PhpMyAdmin 4.0.10deb1

4.2 Batasan - Batasan Implementasi

Terdapat beberapa batasan yang membatasi pengembangan sistem rekomendasi pemilihan mata kuliah ini, di antaranya adalah:

1. *Input* yang diterima oleh sistem berupa daftar mata kuliah yang ditempuh oleh sejumlah mahasiswa pada program studi yang sama dengan program studi dari pengguna aktif
2. *Output* yang dihasilkan oleh sistem berupa prediksi kecenderungan penempuhan mata kuliah yang ditawarkan serta prediksi nilai yang akan didapatkan oleh pengguna aktif
3. Pengembangan sistem dilakukan dengan bahasa pemrograman PHP dengan bantuan *framework* CodeIgniter
4. Basis data akan diimplementasikan dengan menggunakan *database management system* MySQL

4.3 Implementasi Algoritma

Implementasi sistem rekomendasi pemilihan perangkat lunak ini terbagi menjadi tiga bagian, yaitu perhitungan prediksi dengan *content based*, *item-based collaborative filtering*, dan *user-based collaborative filtering*. Berikut ini akan dijelaskan implementasi dari algoritma yang telah dirancang sebelumnya.

4.3.1 Implementasi Content Based Filtering

Pada implementasi algoritma ini, input yang diterima oleh sistem adalah parameter pengguna aktif, di antaranya adalah nomor induk mahasiswa dari pengguna aktif tersebut. Output yang dihasilkan adalah tabel prediksi nilai dari setiap mata kuliah yang ditawarkan.

Berikut ini merupakan kode program dari implementasi *content based filtering* yang ditunjukkan dalam Kode Program 4.1.

1. Fungsi `getCbRekomendasi` pada baris 1 hingga 17 merupakan fungsi utama yang dipanggil untuk menghasilkan rekomendasi dengan metode *content based filtering*. Fungsi ini dipanggil melalui *AJAX request*. Fungsi ini merupakan fungsi induk yang memanggil fungsi-fungsi lainnya. Pertama, fungsi ini

menyiapkan data *sample* dan atribut-atribut yang diperlukan (baris 2 hingga 6), kemudian membuat *decision tree* pada baris ke-9, selanjutnya membuat prediksi nilai dari mata kuliah yang ditawarkan (baris 11 dan 12), dan terakhir menampilkan output berupa tabel (baris 14).

2. Fungsi `getCbID3` pada baris 54 hingga 143 merupakan fungsi yang digunakan untuk menghasilkan *decision tree* dari data sampel yang diberikan. Parameter *input* yang diberikan adalah data sampel, atribut-atribut kompetensi utama lulusan mata kuliah, serta atribut target nilai. Fungsi ini merupakan fungsi rekursif yang memanggil dirinya sendiri. Di dalamnya terdapat beberapa proses, termasuk perhitungan *information gain* dari setiap atributnya. *Output* yang dihasilkan dari fungsi ini adalah *array* yang merupakan struktur *tree* yang terbentuk.
3. Fungsi `getCbID3InformationGain` pada baris ke-193 hingga 249 digunakan untuk menghitung *information gain* dari setiap atribut kompetensi utama lulusan mata kuliah. Baris 199 hingga 208 melakukan penjumlahan terhadap setiap nilai dari setiap atribut yang ada. Selanjutnya, pada baris ke-210 hingga 237 melakukan perhitungan entropi dari seluruh atribut U1 sampai U7. Fungsi ini kemudian memanggil fungsi `getCbID3Entropi` untuk menghitung entropi dari seluruh data sampel (baris ke-239). Terakhir, fungsi ini menghitung *information gain* dari setiap atributnya seperti ditunjukkan pada baris ke-242 hingga 246.
4. Fungsi `getCbID3Entropi` pada baris ke-251 hingga 268 melakukan perhitungan entropi dari data sampel yang diberikan. Pertama, fungsi ini akan melakukan perulangan dari terhadap seluruh data sampel, kemudian melakukan perhitungan terhadap setiap data sampel menggunakan persamaan (2.1). Setiap nilai entropi yang terbentuk kemudian dijumlahkan seperti ditunjukkan pada baris 253 hingga 282.
5. Tahap terakhir dari pembentukan rekomendasi dengan *content based filtering* adalah membentuk tabel prediksi nilai mata kuliah yang dilakukan oleh fungsi `getCbPredictionTable` pada baris ke-145 hingga 172. Fungsi ini melakukan iterasi untuk menyusuri *tree* melalui fungsi `listArrayRecursive` untuk membentuk *rules*. Dari setiap *rules* yang terbentuk, kemudian dilakukan

perulangan untuk menghitung prediksi dengan memanggil fungsi `getCbPrediction` terhadap setiap mata kuliah yang ditawarkan. Proses ini ditunjukkan pada baris ke-166 hingga 170. Fungsi `getCbPrediction` pada baris ke-293 hingga 311 melakukan *traversal* / penyusuran terhadap *rules* dan mencocokkan dengan parameter berupa atribut-atribut kompetensi yang dikirimkan.

```
1 function getCbRekomendasi(){
2     $student = $this->getMhsInfo();
3     $mkKhsKomp = $this->getCbMkKhsKompKHS($student);
4     $sample = $this->getCbID3DataSample($mkKhsKomp);
5     $target = 'K_NILAI';
6     $attr = array('U1', 'U2', 'U3', 'U4', 'U5', 'U6', 'U7');
7
8     $iterasi = 0;
9     $tree = $this->getCbID3($mkKhsKomp, $target, $attr,
10     $iterasi);
11
12     $coursesTawar = $this->m_rekomendasi-
13     >getCbMkTawar($student);
14     $prediksi = $this->getCbPredictionTable($coursesTawar,
15     $tree);
16
17     echo $this->getGenericTable($prediksi, 'table-prediksi-cb',
18     'Tabel Prediksi Content-Based');
19 }
20
21 function getCbMkKhsKompKHS($student){
22     return $this->m_rekomendasi->getCbMkKhsKompKHS($student);
23 }
24
25 function getCbID3DataSample($sample){
26     $arrNilai = array();
27     foreach ($sample as $key => $value){
28         if(!isset($arrNilai[$value['K_NILAI']])){
```

```
27         $arrNilai[$value['K_NILAI']]['U1'][0] = 0;
28         $arrNilai[$value['K_NILAI']]['U1'][1] = 0;
29         $arrNilai[$value['K_NILAI']]['U2'][0] = 0;
30         $arrNilai[$value['K_NILAI']]['U2'][1] = 0;
31         $arrNilai[$value['K_NILAI']]['U3'][0] = 0;
32         $arrNilai[$value['K_NILAI']]['U3'][1] = 0;
33         $arrNilai[$value['K_NILAI']]['U4'][0] = 0;
34         $arrNilai[$value['K_NILAI']]['U4'][1] = 0;
35         $arrNilai[$value['K_NILAI']]['U5'][0] = 0;
36         $arrNilai[$value['K_NILAI']]['U5'][1] = 0;
37         $arrNilai[$value['K_NILAI']]['U6'][0] = 0;
38         $arrNilai[$value['K_NILAI']]['U6'][1] = 0;
39         $arrNilai[$value['K_NILAI']]['U7'][0] = 0;
40         $arrNilai[$value['K_NILAI']]['U7'][1] = 0;
41     }
42
43     $arrNilai[$value['K_NILAI']]['U1'][$value['U1']] =
44     $arrNilai[$value['K_NILAI']]['U1'][$value['U1']] + 1;
45     $arrNilai[$value['K_NILAI']]['U2'][$value['U2']] =
46     $arrNilai[$value['K_NILAI']]['U2'][$value['U2']] + 1;
47     $arrNilai[$value['K_NILAI']]['U3'][$value['U3']] =
48     $arrNilai[$value['K_NILAI']]['U3'][$value['U3']] + 1;
49     $arrNilai[$value['K_NILAI']]['U4'][$value['U4']] =
50     $arrNilai[$value['K_NILAI']]['U4'][$value['U4']] + 1;
51     $arrNilai[$value['K_NILAI']]['U5'][$value['U5']] =
52     $arrNilai[$value['K_NILAI']]['U5'][$value['U5']] + 1;
53     $arrNilai[$value['K_NILAI']]['U6'][$value['U6']] =
54     $arrNilai[$value['K_NILAI']]['U6'][$value['U6']] + 1;
55     $arrNilai[$value['K_NILAI']]['U7'][$value['U7']] =
56     $arrNilai[$value['K_NILAI']]['U7'][$value['U7']] + 1;
57 }
58
59 return $arrNilai;
60 }
61
62 function getCbID3($sample, $target, $attr, &$iterasi){
63     $iterasi++;
64     $tree = array();
65
66     $root = null;
```



```
59
60     $temp = $sample[0][$target];
61     $sama = true;
62     foreach ($sample as $key => $value){
63         if($value[$target] != $temp){
64             $sama = false;
65             break;
66         }
67     }
68     if ($sama == true){
69         $tree[$target] = null;
70         return $tree;
71     }
72
73     if(count($attr) == 0){
74         $nilaiCount = array();
75         foreach ($sample as $key => $value){
76             if(!isset($nilaiCount[$value[$target]])){
77                 $nilaiCount[$value[$target]] = 0;
78             }
79             $nilaiCount[$value[$target]] =
$nilaiCount[$value[$target]] + 1;
80         }
81         $maxNilai = (array_keys($nilaiCount,
max($nilaiCount), true));
82
83         foreach ($maxNilai as $value){
84             $tree[$value] = null;
85         }
86         return $tree;
87     }
88
89     $informationgain = ($this-
>getCbID3InformationGain($sample));
90     if(max($informationgain) == 0){
91         $nilaiCount = array();
92         foreach ($sample as $key => $value){
```

```
93         if(!isset($nilaiCount[$value[$target]])){
94             $nilaiCount[$value[$target]] = 0;
95         }
96         $nilaiCount[$value[$target]] =
$nilaiCount[$value[$target]] + 1;
97     }
98     $maxNilai = (array_keys($nilaiCount,
max($nilaiCount), true));
99
100     foreach ($maxNilai as $value){
101         $tree[$value] = null;
102     }
103     return $tree;
104 }
105
106 else{
107     $informationgain = ($this->
getCbID3InformationGain($sample));
108
109     $A = array_keys($informationgain,
max($informationgain), true);
110     $A = $A[0];
111     $tree[$A] = null;
112
113     $V = array();
114     foreach ($sample as $key => $value){
115         array_push($V, $value[$A]);
116     }
117     $V = array_unique($V);
118     sort($V);
119
120     foreach ($V as $key => $value){
121         $tree[$A][$value] = null;
122
123         $subSample = $this->getCbID3SubSample($sample,
$A, $value);
124         $subAttr = $this->getCbID3SubAttr($attr, $A);
125     }
```

```
126         if(count($subSample) == 0){
127             $nilaiCount = array();
128             foreach ($sample as $key2 => $value2){
129                 if(!isset($nilaiCount[$value2[$target]])){
130                     $nilaiCount[$value2[$target]] = 0;
131                 }
132                 $nilaiCount[$value2[$target]] =
133                 $nilaiCount[$value2[$target]] + 1;
134             }
135             $maxNilai = (array_keys($nilaiCount,
136                 max($nilaiCount), true));
137             $tree[$A][$value][$maxNilai] = null;
138         }
139         else{
140             $tree[$A][$value] = $this-
141             >getCbID3($subSample, $target, $subAttr, $iterasi);
142         }
143     }
144     return $tree;
145 }
146
147 function getCbPredictionTable($mkTawar, $tree){
148     $rules = array();
149     $this->listArrayRecursive($tree, $rules);
150
151     $structuredRules = array();
152     foreach ($rules as $key => $value){
153         $arrRules = explode(',', $value);
154         $countArr = count($arrRules);
155         $row = array();
156         for ($i = 0; $i<$countArr ; $i++){
157             if($i == $countArr-1){
158                 $row['K_NILAI'] = $arrRules[$i];
159             }
160             else if($i % 2 == 0){
```



```
159         $row[$arrRules[$i]] = $arrRules[$i+1];
160     }
161 }
162 array_push($structuredRules, $row);
163 }
164
165 $predictionTable = array();
166 foreach ($mkTawar as $key => $value){
167     $prediction = $this->getCbPrediction($value,
168     $structuredRules);
169     $value['PRED'] = $prediction;
170     array_push($predictionTable, $value);
171 }
172 return $predictionTable;
173 }
174
175 function getCbID3SubSample($sample, $attr, $nilai){
176     $subSample = array();
177     foreach ($sample as $key => $value){
178         if($value[$attr] == $nilai){
179             array_push($subSample, $value);
180         }
181     }
182     return $subSample;
183 }
184
185 function getCbID3SubAttr($attr, $unset){
186     $subAttr = array();
187     foreach ($attr as $key => $value){
188         if($value == $unset) continue;
189         array_push($subAttr, $value);
190     }
191     return ($subAttr);
192 }
193
194 function getCbID3InformationGain($sample){
```

```
194     $arrNilai = $this->getCbID3DataSample($sample);
195     $arrSumNilai = array();
196     $arrEntropi = array();
197     $entropi = array();
198
199     foreach ($arrNilai as $key => $value){
200         foreach ($value as $key2 => $value2){
201             foreach ($value2 as $key3 => $value3){
202                 if(!isset($arrSumNilai[$key2][$key3])){
203                     $arrSumNilai[$key2][$key3] = 0;
204                 }
205                 $arrSumNilai[$key2][$key3] =
206                     $arrSumNilai[$key2][$key3] + $value3;
207             }
208         }
209
210         foreach ($arrNilai as $key => $value){
211             foreach ($value as $key2 => $value2){
212                 foreach ($value2 as $key3 => $value3){
213                     if(!isset($entropi[$key2][$key3])){
214                         $entropi[$key2][$key3] = 0;
215                     }
216
217                     if(!isset($arrEntropi[$key][$key2][$key3])){
218                         $arrEntropi[$key][$key2][$key3]
219                             = 0;
220                     }
221                     if($arrSumNilai[$key2][$key3] == 0){
222                         $scur = 0;
223                     }
224                     else{
225                         $scur = 0 -
226                             ($arrNilai[$key][$key2][$key3] / $arrSumNilai[$key2][$key3])
227                             *
228                             log(($arrNilai[$key][$key2][$key3] / $arrSumNilai[$key2][$key3]),
229                                 2);
230                     }
231                 }
232             }
233         }
234     }
235 }
```

```

226
227         if(is_nan($cur)){
228             $arrEntropi[$key][$key2][$key3]
= 0;
229             $entropi[$key2][$key3] =
$entropi[$key2][$key3] + 0;
230         }
231         else{
232             $arrEntropi[$key][$key2][$key3]
= $cur;
233             $entropi[$key2][$key3] =
$entropi[$key2][$key3] + $cur;
234         }
235     }
236 }
237 }
238
239 $entropyS = $this->getCbID3Entropi($sample);
240
241 $arrGain = array();
242 foreach ($entropi as $key => $value){
243     $arrGain[$key] = $entropyS
244         - (($arrSumNilai[$key][0] / ($arrSumNilai[$key][0] +
$arrSumNilai[$key][1])) * $entropi[$key][0])
245         - (($arrSumNilai[$key][1] / ($arrSumNilai[$key][0] +
$arrSumNilai[$key][1])) * $entropi[$key][1]);
246 }
247
248 return $arrGain;
249 }
250
251 function getCbID3Entropi($sample){
252     $arrNilai = $this->getCbID3Dataset($sample);
253     $entropyArr = array();
254     foreach ($arrNilai as $key => $value){
255         if(! (isset($entropyArr[$key]))){
256             $entropyArr[$key] = 0;
257         }

```



```
258         $entropyArr[$key] = $entropyArr[$key] +
259             (-(sizeof($value) / sizeof($sample)) *
260              log(sizeof($value) / sizeof($sample), 2));
261     }
262     $entropy = 0;
263     foreach ($entropyArr as $key => $val){
264         $entropy = $entropy + $val;
265     }
266
267     return $entropy;
268 }
269
270 function getCbID3Dataset($sample){
271     $arrNilai = array();
272     foreach ($sample as $key => $value){
273         $arrNilai[$value['K_NILAI']] [$value['K_MK']] .=
274         '.$value['THN_MK']] = 1;
275     }
276     return $arrNilai;
277 }
278
279 function listArrayRecursive($someArray, &$rules) {
280     foreach ($iterator as $k => $v) {
281         if ($iterator->hasChildren()) {
282             } else {
283                 for ($p = array(), $i = 0, $z = $iterator-
284                 >getDepth(); $i <= $z; $i++) {
285                     $p[] = $iterator->getSubIterator($i)-
286                     >key();
287                 }
288                 $path = implode(',', $p);
289                 array_push($rules, $path);
290             }
291         }
292     }
```

```

291 }
292
293 function getCbPrediction($mk, $rules){
294     $prediction = array();
295     foreach ($rules as $key => $rule){
296         $flag = 0;
297         foreach ($rule as $key2 => $value2){
298             if($key2 != 'K_NILAI'){
299                 if($value2 == $mk[$key2]){
300                     $flag = 1;
301                 }
302                 else {
303                     $flag = 0;
304                     break;
305                 }
306             }
307         }
308         if($flag == 1){
309             break;
310         }
311     }
312     return $rule['K_NILAI'];
313 }

```

Kode Program 4.1 Implementasi content-based filtering

4.3.2 Implementasi User-Based Collaborative Filtering

Pada implementasi *user-based collaborative filterin*, sistem akan melakukan perhitungan similaritas antar mahasiswa dan prediksi kecenderungan penempuhan suatu mata kuliah.

Berikut ini merupakan kode program dari implementasi *user based collaborative filtering* yang ditunjukkan dalam Kode Program 4.2.

1. Fungsi `getUcfRekomendasi` pada baris pertama hingga baris ke-30 merupakan fungsi induk yang dipanggil melalui *AJAX request*. Fungsi ini menghasilkan *output* berupa tabel prediksi kecenderungan penempuhan mata kuliah yang

ditawarkan dengan melakukan perhitungan similaritas antar *user* / mahasiswa terlebih dahulu.

2. Fungsi `getUcfSimilarityTable` pada baris ke-81 hingga 97 digunakan untuk menghitung similaritas antar pengguna dengan parameter masukan berupa *array* yang berisi nilai deviasi yang telah dihitung sebelumnya. Untuk mempersingkat waktu pemrosesan, perhitungan similaritas telah dilakukan sebelumnya sehingga program langsung membaca hasil perhitungan similaritas dari *text file* (baris ke-4 hingga 6).
3. Fungsi `getUcfPredictionTable` pada baris ke-158 hingga 171 menghasilkan keluaran berupa *array* yang berisi tabel prediksi kecenderungan penempuhan dari mata kuliah yang ditawarkan. Fungsi ini melakukan perulangan sebanyak jumlah mata kuliah yang ditawarkan dengan memanggil fungsi `getUcfPredictionWeightedSum` atau `getUcfPredictionHerlocker`. Masing-masing fungsi tersebut melakukan perhitungan prediksi dengan menggunakan persamaan (2.4) dan (2.5).

```

1 function getUcfRekomendasi(){
2     $student = $this->getMhsInfo();
3
4     $dirSimilarityArr =
FCPATH."public/files/rec/ucf/similarity/";
5     $filenameSimilarityArr =
$student['K_JURUSAN'].$student['K_FAKULTAS'].$student['K_JENJANG']
."-arr.dat";
6     $similarityArr =
unserialize(file_get_contents($dirSimilarityArr.$filenameSimilarit
yArr));
7
8     $coursesTawar = $this->m_mk->getMKTawar($student);
9
10    $students = $this->m_mahasiswa->getUsersAvailable($student);
11
12    $coursesTempuhJur = $this->m_rekomendasi-
>getIcfMkKhsTempuh($student);
13

```



```

14      $coursesTempuhJur = $this->getUcfCourseTempuh($coursesTempuhJur);
15
16      $similarityArr = $this->getUcfRearrangeArr($similarityArr);
17
18      $avgUserRating = $this->getUcfAvgUserRating($coursesTempuhJur);
19
20      $predictionTableWei = $this->getUcfPredictionTable($student,
21      $students, $coursesTawar, $coursesTempuhJur, $similarityArr, null,
22      'weightedsum');
23
24      $predictionTableHer = $this->getUcfPredictionTable($student,
25      $students, $coursesTawar, $coursesTempuhJur, $similarityArr,
26      $avgUserRating, 'herlocker');
27
28      $html = $this->getIcfPredictionTableHtml($predictionTableWei, "Tabel Prediksi UCF
29      - Weighted Sum", "tabel-prediksi-ucf-wei");
30
31      echo $html;
32
33      $html = $this->getIcfPredictionTableHtml($predictionTableHer, "Tabel Prediksi UCF
34      - Herlocker", "tabel-prediksi-ucf-her");
35
36      echo $html;
37
38      exit;
39
40 }
41
42 function getUcfTransformMkKhsTempuh($mkKhsTempuh) {
43     $deviasi = array();
44     foreach ($mkKhsTempuh as $key => $value){
45
46         $deviasi[$value['K_MK']][$value['THN_MK']][$value['NIM']] ['
47         IS_TEMPUPH'] = $value['IS_TEMPUPH'];
48     }
49     return $deviasi;
50 }
51
52 function getUcfTabelDeviasi($deviasi){
53     $arrSumCountAvg = array();

```

```

42     foreach ($deviasi as $key=>$value){
43         foreach ($value as $key2 => $value2){
44             $sumRating = 0;
45             $countRating = 0;
46             foreach ($value2 as $key3 => $value3){
47                 $sumRating = $sumRating +
$value3['IS_TEMPURH'];
48                 $countRating++;
49             }
50         }
51         $arrSumCountAvg[$key][$key2]['SUMRATING'] =
$sumRating;
52         $arrSumCountAvg[$key][$key2]['COUNTRATING'] =
$countRating;
53         $arrSumCountAvg[$key][$key2]['AVGRATING'] =
$sumRating / $countRating;
54     }
55
56     foreach ($deviasi as $key=>$value){
57         foreach ($value as $key2 => $value2){
58             $sumRating = 0;
59             $countRating = 0;
60             foreach ($value2 as $key3 => $value3){
61                 $deviasi[$key][$key2][$key3]['AVGRATING'] =
$arrSumCountAvg[$key][$key2]['AVGRATING'];
62                 $deviasi[$key][$key2][$key3]['DEVIASI']
= $value3['IS_TEMPURH'] - $arrSumCountAvg[$key][$key2]['AVGRATING'];
63             }
64         }
65     }
66     return $deviasi;
67 }
68
69 function getUcfTransposeTabelDeviasi($deviasi){
70     $deviasiTranspose = array();
71     foreach ($deviasi as $key => $value){
72         foreach ($value as $key2 => $value2){

```

```
73         foreach ($value2 as $key3 => $value3){
74             $deviasiTranspose[$key2][$key3][$key] =
75             $value3;
76         }
77     }
78     return $deviasiTranspose;
79 }
80
81 function getUcfSimilarityTable($arr, $students){
82     $similarity = array();
83     foreach ($students as $key => $value){
84         foreach ($students as $key2 => $value2){
85             if($key2 < $key){
86                 continue;
87             }
88             $similarity[$value['NIM']][$value2['NIM']]
89             = $this->getUcfSimilarity(
90                 array($value['NIM']),
91                 array($value2['NIM']),
92                 $arr
93             );
94         }
95     }
96     return $similarity;
97 }
98
99 function getUcfSimilarity($item1, $item2, $arr){
100     $sum1 = 0;
101     $sum2 = 0;
102     $sum3 = 0;
103     $sum4 = 0;
104
105     $i = 0;
106     foreach ($arr as $key => $value){
```



```

107         foreach ($value as $key2 => $value2){
108             if(isset($value2[$item1[0]]) &&
isset($value2[$item2[0]])){
109                 $sum1 = $sum1 + ($value2[$item1[0]] *
$value2[$item2[0]]);
110                 $sum2 = $sum2 + (($value2[$item1[0]]) *
($value2[$item1[0]]));
111                 $sum3 = $sum3 + (($value2[$item2[0]]) *
($value2[$item2[0]]));
112             }
113             else {
114             }
115         }
116     }
117 }
118 $penyebut = sqrt($sum2) * sqrt($sum3);
119 if ($penyebut == 0) $sim = 0;
120 else $sim = ($sum1) / ($penyebut);
121 return $sim;
122 }
123
124 function getUcfCourseTempuh($courseTempuh){
125     $sarrangedArray = array();
126     foreach ($courseTempuh as $value){
127         $sarrangedArray[$value['NIM']][$value['K_MK'].'-
'. $value['THN_MK']] = $value['IS_TEMPUEH'];
128     }
129     return $sarrangedArray;
130 }
131
132 function getUcfRearrangeArr($similarityArr){
133     $sarrangedSimilarityArr = array();
134     foreach ($similarityArr as $key => $value){
135         foreach ($value as $key2 => $value2){
136             $sarrangedSimilarityArr[$key][$key2] = $value2;
137             $sarrangedSimilarityArr[$key2][$key] = $value2;
138         }

```

```

139     }
140     return $arrangedSimilarityArr;
141 }
142
143 function getUcfAvgUserRating($mkKhs){
144     $avgRating = array();
145     foreach ($mkKhs as $key => $value){
146         if(!isset($avgRating[$key]['SUM'])){
147             $avgRating[$key]['SUM'] = 0;
148         }
149         if(!isset($avgRating[$key]['COUNT'])){
150             $avgRating[$key]['COUNT'] = 0;
151         }
152         if(!isset($avgRating[$key]['AVG'])){
153             $avgRating[$key]['AVG'] = 0;
154         }
155         foreach ($value as $key2 => $value2){
156             $avgRating[$key]['SUM'] =
157             $avgRating[$key]['SUM'] + $value2;
158             $avgRating[$key]['COUNT'] =
159             $avgRating[$key]['COUNT'] + 1;
160             $avgRating[$key]['AVG'] =
161             $avgRating[$key]['SUM'] / $avgRating[$key]['COUNT'];
162         }
163     }
164     return $avgRating;
165 }
166
167 function getUcfPredictionTable($user, $students, $mkTawar,
168 $coursesTempuhJur, $similarityTable, $avgUsersRating, $type){
169     $predictionTable = array();
170     $i = 0;
171     foreach ($mkTawar as $key => $value){
172         if($type == 'weightedsum') $pred = $this-
173         >getUcfPredictionWeightedSum($user, $value, $coursesTempuhJur,
174 $similarityTable);
175         else if($type == 'herlocker') $pred = $this-
176         >getUcfPredictionHerlocker($user, $value, $coursesTempuhJur,
177 $similarityTable, $avgUsersRating);
178         $predictionTable[$value['K_MK']][$value['THN_MK']]['PRED']
179         = $pred;
180         $i++;
181     }
182 }

```

```

166         if($i == 20){
167             break;
168         }
169     }
170     return $predictionTable;
171 }
172
173 function getUcfPredictionWeightedSum($user, $mkTawar,
    $coursesTempuh, $similarityTable){
174     $sum1 = 0;
175     $sum2 = 0;
176     $i = 0;
177
178     foreach ($similarityTable as $key => $value){
179         foreach ($value as $key2 => $value2){
180             if($key == $user['NIM'] || $key2 ==
    $user['NIM']){
181                 if($coursesTempuh[$key2][$mkTawar['K_MK']] !=
    '.$mkTawar['THN_MK']] >= 0){ // jika mahasiswa menempuh mk tersebut
182                     $sum1 = $sum1 + (
183                         $value2
    $coursesTempuh[$key2][$mkTawar['K_MK']] != '.$mkTawar['THN_MK']] //
    * tempuh[nim][kmk]
184                     );
185                     $sum2 = $sum2 + abs($value2);
186                 }
187             }
188         }
189     }
190
191     if($sum2 == 0){ $pred = 0;}
192     else $pred = $sum1 / $sum2;
193     return $pred;
194 }
195
196 function getUcfPredictionHerlocker($user, $mkTawar, $coursesTempuh,
    $similarityTable, $avgRating){

```



```

197     $sum1 = 0;
198     $sum2 = 0;
199     $i = 0;
200
201     foreach ($similarityTable as $key => $value){
202         foreach ($value as $key2 => $value2){
203             if($key == $user['NIM'] || $key2 ==
204 $user['NIM']){
205                 if($coursesTempuh[$key2][$mkTawar['K_MK']]['-
206                 '.$mkTawar['THN_MK']] >= 0 && ($value2 >=0)){
207                     $sum1 = $sum1 + (
208                         $value2 * (
209                             $coursesTempuh[$key2][$mkTawar['K_MK']]['-
210                             '.$mkTawar['THN_MK']]
211                             -
212                             $avgRating[$key2]['AVG']
213                         )
214                     );
215                     $sum2 = $sum2 + abs($value2);
216                 }
217             }
218         }
219     }
220
221     if($sum2 == 0){ $pred = 0;}
222     else $pred = $avgRating[$user['NIM']]['AVG'] + ($sum1 /
223 $sum2);
224
225     return $pred;
226 }
227 ?>

```

Kode Program 4.2 Implementasi user-based collaborative filtering

4.3.3 Implementasi Item-Based Collaborative Filtering

Pada implementasi *item-based collaborative filtering*, sistem akan melakukan perhitungan similaritas antar item mata kuliah dan prediksi kecenderungan penempuhan suatu mata kuliah.

Berikut ini merupakan kode program dari implementasi *content based filtering* yang ditunjukkan dalam Kode Program 4.3.

1. Fungsi `getIcfRekomendasi` pada baris pertama hingga baris ke-36 adalah fungsi induk yang dipanggil melalui *AJAX request* yang memberikan keluaran berupa tabel prediksi kecenderungan seorang mahasiswa untuk menempuh mata kuliah yang ditawarkan. Secara umum, fungsi ini melakukan dua macam perhitungan, yaitu perhitungan similaritas antar *item* dan perhitungan prediksi kecenderungan penempuhan dari mata kuliah yang ditawarkan.
2. Fungsi `getIcfSimilarityTable` pada baris ke-105 hingga baris ke-121 menghasilkan *output* berupa tabel similaritas antar *item*. Perhitungan similaritas dilakukan dengan perulangan sejumlah *item* yang dibandingkan dengan memanggil fungsi `getIcfSimilarity` untuk membandingkan similaritas antar dua *item*.
3. Fungsi `getIcfPredictionTable` pada baris ke-143 hingga baris ke-151 menghasilkan *output* berupa tabel prediksi kecenderungan penempuhan dari mata kuliah yang ditawarkan. Fungsi ini melakukan perulangan sebanyak mata kuliah yang ditawarkan dengan memanggil fungsi `getIcfPredictionWeightedSum` atau `getIcfPredictionHerlocker`. Kedua fungsi ini masing-masing melakukan perhitungan prediksi dengan persamaan *weighted sum* (2.8) dan persamaan (2.9) dari Herlocker.

```

1 function getIcfRekomendasi() {
2     $student = $this->getMhsInfo();
3     $mkKhsTempuh = $this->getIcfMkKhsTempuh($student);
4
5     $dirDev = FCPATH."public/files/rec/icf/deviasi/";
6     $filenameDevArr =
7     $student['K_JURUSAN'].$student['K_FAKULTAS'].$student['K_JENJANG']
8     ."-arr.dat";
9     $filenameDevJson =
10    $student['K_JURUSAN'].$student['K_FAKULTAS'].$student['K_JENJANG']
11    ."-json.dat";
12
13    $deviasiArr =
14    unserialize(file_get_contents($dirDev.$filenameDevArr));

```

```
10     $deviasiCrossArr = $this->getIcfTabelDeviasiCrossed($deviasiArr);
11
12     $dirDevCross = FCPATH."public/files/rec/icf/deviasi-cross/";
13
14     $filenameDevCrossArr =
15     $student['K_JURUSAN'].$student['K_FAKULTAS'].$student['K_JENJANG']
16     ."-arr.dat";
17
18     $devCrossArr =
19     unserialize(file_get_contents($dirDevCross.$filenameDevCrossArr));
20
21     $courses = $this->m_mk->getMKAvailable($student);
22
23     $similarityArr = $this->getIcfSimilarityTable($devCrossArr,
24     $courses);
25
26     $coursesTawar = $this->m_mk->getMKTawar($student);
27
28     $coursesKhs = $this->m_khs->getAllKhsMhs($student);
29
30     $avgItemsRating = $this->getIcfAvgItemsRating($mkKhsTempuh);
31
32     $predictionTableWei = $this->getIcfPredictionTable($coursesTawar, $coursesKhs, $similarityArr,
33     $avgItemsRating, 'weightedsum');
34
35     $predictionTableHer = $this->getIcfPredictionTable($coursesTawar, $coursesKhs, $similarityArr,
36     $avgItemsRating, 'herlocker');
37
38     $html = $this->getIcfPredictionTableHtml($predictionTableWei, "Tabel Prediksi
39     ICF - Weighted Sum", "tabel-prediksi-icf-wei");
40
41     echo $html;
42
43     $html = $this->getIcfPredictionTableHtml($predictionTableHer, "Tabel Prediksi
44     ICF - Herlocker", "tabel-prediksi-icf-her");
45
46     echo $html;
47
48     exit;
```



```
36 }
37
38 function getIcfMkKhsTempuh($student){
39     return $this->m_rekomendasi->getIcfMkKhsTempuh($student);
40 }
41
42 function getIcfTransformMkKhsTempuh($mkKhsTempuh){
43     $deviasi = array();
44     foreach ($mkKhsTempuh as $key => $value){
45         $deviasi[$value['NIM']][$value['K_MK']][$value['THN_MK']][
IS_TEMPUEH'] = $value['IS_TEMPUEH'];
46     }
47     return $deviasi;
48 }
49
50 function getIcfTabelDeviasi($deviasi){
51     $arrSumCountAvg = array();
52     foreach ($deviasi as $key=>$value){
53         $sumRating = 0;
54         $countRating = 0;
55         foreach ($value as $key2 => $value2){
56             foreach ($value2 as $key3 => $value3){
57                 $sumRating = $sumRating +
58                 $value3['IS_TEMPUEH'];
59                 $countRating++;
60             }
61             $arrSumCountAvg[$key]['SUMRATING'] = $sumRating;
62             $arrSumCountAvg[$key]['COUNTRATING'] = $countRating;
63             $arrSumCountAvg[$key]['AVGRATING'] = $sumRating /
64             $countRating;
65         }
66     }
67     foreach ($deviasi as $key=>$value){
68         $sumRating = 0;
```

```

68     $countRating = 0;
69     foreach ($value as $key2 => $value2){
70         foreach ($value2 as $key3 => $value3){
71             $deviasi[$key][$key2][$key3]['AVGRATING'] =
72             $arrSumCountAvg[$key]['AVGRATING'];
73             $deviasi[$key][$key2][$key3]['DEVIASI']
74             = $value3['IS_TEMPUH'] - $arrSumCountAvg[$key]['AVGRATING'];
75         }
76     }
77     return $deviasi;
78 }
79 function getIcfTabelDeviasiCrossed($deviasiArr){
80     $deviasiCrossArr = array();
81     foreach ($deviasiArr as $key => $value){
82         foreach ($value as $key2 => $value2){
83             foreach ($value2 as $key3 => $value3){
84                 $deviasiCrossArr[$key][$key2][$key3] =
85                 $value3['DEVIASI'];
86             }
87         }
88     }
89     return $deviasiCrossArr;
90 }
91 function getIcfUserAverageRating($arr){
92     $userAverageRating = array();
93     foreach ($arr as $key => $value){
94         foreach ($value as $key2 => $value2){
95             foreach ($value2 as $key3 => $value3){
96                 $userAverageRating[$key] =
97                 $value3['AVGRATING'];
98                 break;
99             }
100         }
101     }
102     break;

```

```

100         }
101     }
102     return $userAverageRating;
103 }
104
105 function getIcfSimilarityTable($arr, $course){
106     $similarity = array();
107     foreach ($course as $key => $value){
108         foreach ($course as $key2 => $value2){
109             if($key2 < $key){
110                 continue;
111             }
112             $similarity[$value['K_MK']."-
113             ".$value['THN_MK']] [$value2['K_MK']."-".$value2['THN_MK']]
114             = $this->getIcfSimilarity(
115             array($value['K_MK'],
116             $value['THN_MK']),
117             array($value2['K_MK'],
118             $value2['THN_MK']),
119             $arr
120             );
121         }
122     }
123     return $similarity;
124 }
125
126 function getIcfSimilarity($item1, $item2, $arr){
127     $sum1 = 0;
128     $sum2 = 0;
129     $sum3 = 0;
130     $sum4 = 0;
131
132     $i = 0;
133     foreach ($arr as $key => $value){
134         $sum1 = $sum1 + ($value[$item1[0]][$item1[1]] *
135         $value[$item2[0]][$item2[1]]);

```



```

132         $sum2 = $sum2 + (($value[$item1[0]][$item1[1]]) *
($value[$item1[0]][$item1[1]]));
133         $sum3 = $sum3 + (($value[$item2[0]][$item2[1]]) *
($value[$item2[0]][$item2[1]]));
134         $i++;
135         if($i==2){
136         }
137
138     }
139     $sim = ($sum1) / (sqrt($sum2) * sqrt($sum3));
140     return $sim;
141 }
142
143 function getIcfPredictionTable($coursesTawar, $coursesKhs,
$similarityTable, $avgItemsRating, $type){
144     $predictionTable = array();
145     foreach ($coursesTawar as $key => $value){
146         if($type == 'weightedsum') $pred = $this-
>getIcfPredictionWeightedSum($value, $coursesKhs,
$similarityTable);
147         else if($type == 'herlocker') $pred = $this-
>getIcfPredictionHerlocker($value, $coursesKhs, $similarityTable,
$avgItemsRating);
148         $predictionTable[$value['K_MK']] [$value['THN_MK']] ['PRED']
= $pred;
149     }
150     return $predictionTable;
151 }
152
153 function getIcfPredictionWeightedSum($mkTawar, $courseKhs,
$similarityTable){
154     $sum1 = 0;
155     $sum2 = 0;
156     $i = 0;
157     foreach ($courseKhs as $key => $value){
158         if($mkTawar['K_MK']. "-". $mkTawar['THN_MK'] ==
$value['K_MK']. "-". $value['THN_MK']) {
159             continue;
160         }

```

```

161
162         if (isset($similarityTable[$mkTawar['K_MK']]."-"
163         ".$mkTawar['THN_MK']][$value['K_MK']]."-"
164         ".$mkTawar['THN_MK']]) {
165         if($similarityTable
166         [$mkTawar['K_MK']]."-"
167         ".$mkTawar['THN_MK']][$value['K_MK']]."-"
168         ".$value['THN_MK']] >= 0) {
169
170         $sum1 = $sum1 +
171         ($similarityTable
172         [$mkTawar['K_MK']]."-"
173         ".$mkTawar['THN_MK']][$value['K_MK']]."-"
174         ".$value['THN_MK']]
175         * 1);
176         $sum2 = $sum2 +
177         abs($similarityTable
178         [$mkTawar['K_MK']]."-"
179         ".$mkTawar['THN_MK']][$value['K_MK']]."-"
180         ".$value['THN_MK']]);
181         }
182         else if
183         (isset($similarityTable[$value['K_MK']]."-"
184         ".$value['THN_MK']][$mkTawar['K_MK']]."-"
185         ".$mkTawar['THN_MK']])) {
186         if($similarityTable
187         [$value['K_MK']]."-"
188         ".$mkTawar['THN_MK']][$mkTawar['K_MK']]."-"
189         ".$mkTawar['THN_MK']] >= 0) {
190
191         $sum1 = $sum1 +
192         ($similarityTable
193         [$value['K_MK']]."-"
194         ".$mkTawar['THN_MK']][$mkTawar['K_MK']]."-"
195         ".$mkTawar['THN_MK']]
196         * 1);
197         $sum2 = $sum2 +
198         abs($similarityTable
199         [$mkTawar['K_MK']]."-"
200         ".$mkTawar['THN_MK']][$value['K_MK']]."-"
201         ".$mkTawar['THN_MK']]);
202         }
203         }
204     }
205     }
206     }
207     }
208     }
209     }
210     }
211     }
212     }
213     }
214     }
215     }
216     }
217     }
218     }
219     }
220     }
221     }
222     }
223     }
224     }
225     }
226     }
227     }
228     }
229     }
230     }
231     }
232     }
233     }
234     }
235     }
236     }
237     }
238     }
239     }
240     }
241     }
242     }
243     }
244     }
245     }
246     }
247     }
248     }
249     }
250     }
251     }
252     }
253     }
254     }
255     }
256     }
257     }
258     }
259     }
260     }
261     }
262     }
263     }
264     }
265     }
266     }
267     }
268     }
269     }
270     }
271     }
272     }
273     }
274     }
275     }
276     }
277     }
278     }
279     }
280     }
281     }
282     }
283     }
284     }
285     }
286     }
287     }
288     }
289     }
290     }
291     }
292     }
293     }
294     }
295     }
296     }
297     }
298     }
299     }
300     }
301     }
302     }
303     }
304     }
305     }
306     }
307     }
308     }
309     }
310     }
311     }
312     }
313     }
314     }
315     }
316     }
317     }
318     }
319     }
320     }
321     }
322     }
323     }
324     }
325     }
326     }
327     }
328     }
329     }
330     }
331     }
332     }
333     }
334     }
335     }
336     }
337     }
338     }
339     }
340     }
341     }
342     }
343     }
344     }
345     }
346     }
347     }
348     }
349     }
350     }
351     }
352     }
353     }
354     }
355     }
356     }
357     }
358     }
359     }
360     }
361     }
362     }
363     }
364     }
365     }
366     }
367     }
368     }
369     }
370     }
371     }
372     }
373     }
374     }
375     }
376     }
377     }
378     }
379     }
380     }
381     }
382     }
383     }
384     }
385     }
386     }
387     }
388     }
389     }
390     }
391     }
392     }
393     }
394     }
395     }
396     }
397     }
398     }
399     }
400     }
401     }
402     }
403     }
404     }
405     }
406     }
407     }
408     }
409     }
410     }
411     }
412     }
413     }
414     }
415     }
416     }
417     }
418     }
419     }
420     }
421     }
422     }
423     }
424     }
425     }
426     }
427     }
428     }
429     }
430     }
431     }
432     }
433     }
434     }
435     }
436     }
437     }
438     }
439     }
440     }
441     }
442     }
443     }
444     }
445     }
446     }
447     }
448     }
449     }
450     }
451     }
452     }
453     }
454     }
455     }
456     }
457     }
458     }
459     }
460     }
461     }
462     }
463     }
464     }
465     }
466     }
467     }
468     }
469     }
470     }
471     }
472     }
473     }
474     }
475     }
476     }
477     }
478     }
479     }
480     }
481     }
482     }
483     }
484     }
485     }
486     }
487     }
488     }
489     }
490     }
491     }
492     }
493     }
494     }
495     }
496     }
497     }
498     }
499     }
500     }
501     }
502     }
503     }
504     }
505     }
506     }
507     }
508     }
509     }
510     }
511     }
512     }
513     }
514     }
515     }
516     }
517     }
518     }
519     }
520     }
521     }
522     }
523     }
524     }
525     }
526     }
527     }
528     }
529     }
530     }
531     }
532     }
533     }
534     }
535     }
536     }
537     }
538     }
539     }
540     }
541     }
542     }
543     }
544     }
545     }
546     }
547     }
548     }
549     }
550     }
551     }
552     }
553     }
554     }
555     }
556     }
557     }
558     }
559     }
560     }
561     }
562     }
563     }
564     }
565     }
566     }
567     }
568     }
569     }
570     }
571     }
572     }
573     }
574     }
575     }
576     }
577     }
578     }
579     }
580     }
581     }
582     }
583     }
584     }
585     }
586     }
587     }
588     }
589     }
590     }
591     }
592     }
593     }
594     }
595     }
596     }
597     }
598     }
599     }
600     }
601     }
602     }
603     }
604     }
605     }
606     }
607     }
608     }
609     }
610     }
611     }
612     }
613     }
614     }
615     }
616     }
617     }
618     }
619     }
620     }
621     }
622     }
623     }
624     }
625     }
626     }
627     }
628     }
629     }
630     }
631     }
632     }
633     }
634     }
635     }
636     }
637     }
638     }
639     }
640     }
641     }
642     }
643     }
644     }
645     }
646     }
647     }
648     }
649     }
650     }
651     }
652     }
653     }
654     }
655     }
656     }
657     }
658     }
659     }
660     }
661     }
662     }
663     }
664     }
665     }
666     }
667     }
668     }
669     }
670     }
671     }
672     }
673     }
674     }
675     }
676     }
677     }
678     }
679     }
680     }
681     }
682     }
683     }
684     }
685     }
686     }
687     }
688     }
689     }
690     }
691     }
692     }
693     }
694     }
695     }
696     }
697     }
698     }
699     }
700     }
701     }
702     }
703     }
704     }
705     }
706     }
707     }
708     }
709     }
710     }
711     }
712     }
713     }
714     }
715     }
716     }
717     }
718     }
719     }
720     }
721     }
722     }
723     }
724     }
725     }
726     }
727     }
728     }
729     }
730     }
731     }
732     }
733     }
734     }
735     }
736     }
737     }
738     }
739     }
740     }
741     }
742     }
743     }
744     }
745     }
746     }
747     }
748     }
749     }
750     }
751     }
752     }
753     }
754     }
755     }
756     }
757     }
758     }
759     }
760     }
761     }
762     }
763     }
764     }
765     }
766     }
767     }
768     }
769     }
770     }
771     }
772     }
773     }
774     }
775     }
776     }
777     }
778     }
779     }
780     }
781     }
782     }
783     }
784     }
785     }
786     }
787     }
788     }
789     }
790     }
791     }
792     }
793     }
794     }
795     }
796     }
797     }
798     }
799     }
800     }
801     }
802     }
803     }
804     }
805     }
806     }
807     }
808     }
809     }
810     }
811     }
812     }
813     }
814     }
815     }
816     }
817     }
818     }
819     }
820     }
821     }
822     }
823     }
824     }
825     }
826     }
827     }
828     }
829     }
830     }
831     }
832     }
833     }
834     }
835     }
836     }
837     }
838     }
839     }
840     }
841     }
842     }
843     }
844     }
845     }
846     }
847     }
848     }
849     }
850     }
851     }
852     }
853     }
854     }
855     }
856     }
857     }
858     }
859     }
860     }
861     }
862     }
863     }
864     }
865     }
866     }
867     }
868     }
869     }
870     }
871     }
872     }
873     }
874     }
875     }
876     }
877     }
878     }
879     }
880     }
881     }
882     }
883     }
884     }
885     }
886     }
887     }
888     }
889     }
890     }
891     }
892     }
893     }
894     }
895     }
896     }
897     }
898     }
899     }
900     }
901     }
902     }
903     }
904     }
905     }
906     }
907     }
908     }
909     }
910     }
911     }
912     }
913     }
914     }
915     }
916     }
917     }
918     }
919     }
920     }
921     }
922     }
923     }
924     }
925     }
926     }
927     }
928     }
929     }
930     }
931     }
932     }
933     }
934     }
935     }
936     }
937     }
938     }
939     }
940     }
941     }
942     }
943     }
944     }
945     }
946     }
947     }
948     }
949     }
950     }
951     }
952     }
953     }
954     }
955     }
956     }
957     }
958     }
959     }
960     }
961     }
962     }
963     }
964     }
965     }
966     }
967     }
968     }
969     }
970     }
971     }
972     }
973     }
974     }
975     }
976     }
977     }
978     }
979     }
980     }
981     }
982     }
983     }
984     }
985     }
986     }
987     }
988     }
989     }
990     }
991     }
992     }
993     }
994     }
995     }
996     }
997     }
998     }
999     }
1000    }

```

```

189     [$mkTawar['K_MK']]."-"$.mkTawar['THN_MK']]
190         * 1);
191     $sum2 = $sum2 +
192         abs($similarityTable
193             [$value['K_MK']]."-"
194             $.value['THN_MK']]
195             [$mkTawar['K_MK']]."-"$.mkTawar['THN_MK']]);
196     }
197 }
198 }
199 }
200
201 if($sum2 == 0){ $pred = 0;}
202 else $pred = $sum1 / $sum2;
203 return $pred;
204 }
205
206 function getIcfPredictionHerlocker($mkTawar, $courseKhs,
207 $similarityTable, $avgItemsRating){
208     $sum1 = 0;
209     $sum2 = 0;
210     foreach ($courseKhs as $key => $value){
211         if($mkTawar['K_MK']]."-"$.value['THN_MK'] ==
212             $value['K_MK']]."-"$.value['THN_MK']) continue;
213
214         if (isset($similarityTable[$mkTawar['K_MK']]."-"
215             $.value['THN_MK']][$value['K_MK']]."-"$.value['THN_MK']])){
216             if($similarityTable
217                 [$mkTawar['K_MK']]."-"
218                 $.value['THN_MK']]
219                 [$value['K_MK']]."-"
220                 $.value['THN_MK']] >= 0){
221
222                 $sum1 = $sum1 +
223                 ($similarityTable

```


219	\$.value['THN_MK']]	[\$mkTawar['K_MK']]."-
220	\$.value['THN_MK']]	[\$value['K_MK']]."-
221		* (1 -
222	(\$avgItemsRating[\$value['K_MK']][\$value['THN_MK']]['AVG'])));	
223	\$sum2 = \$sum2 +	
224	abs(\$similarityTable	[\$mkTawar['K_MK']]."-
225	\$.value['THN_MK']]	[\$value['K_MK']]."-
226	\$.value['THN_MK']]);	
227	}	
228		
229	}	
230	else if (isset(\$similarityTable[\$value['K_MK']]."-	
231	\$.value['THN_MK']][\$mkTawar['K_MK']]."-\$.value['THN_MK']])){	
232	if(\$similarityTable	[\$value['K_MK']]."-
233	\$.value['THN_MK']]	[\$mkTawar['K_MK']]."-
234	\$.value['THN_MK']] >= 0){	
235	\$sum1 = \$sum1 +	
236	(\$similarityTable	
237	\$.value['THN_MK']]	[\$value['K_MK']]."-
238	\$.value['THN_MK']]	[\$mkTawar['K_MK']]."-
239		* (1 -
240	(\$avgItemsRating[\$value['K_MK']][\$value['THN_MK']]['AVG'])));	
241	\$sum2 = \$sum2 +	
242	abs(\$similarityTable	[\$value['K_MK']]."-
243	\$.value['THN_MK']]	[\$mkTawar['K_MK']]."-
244	\$.value['THN_MK']]);	
245	}	

```

246         }
247     }
248
249     if($sum2 == 0){ $pred =
$avgItemsRating[$mkTawar['K_MK']][$mkTawar['THN_MK']]['AVG'] + 0;}
// $sum1/$sum2 dianggap nol
250
    else $pred =
$avgItemsRating[$mkTawar['K_MK']][$mkTawar['THN_MK']]['AVG']
+($sum1 / $sum2);
251
    return $pred;
252 }
253
254 function getIcfAvgItemsRating($mkKhsTempuh){
255     $avgRating = array();
256     foreach ($mkKhsTempuh as $key => $value){
257
        if(!isset($avgRating[$value['K_MK']][$value['THN_MK']]['SUM
        '])){ $avgRating[$value['K_MK']][$value['THN_MK']]['SUM'] = 0;}
258
        if(!isset($avgRating[$value['K_MK']][$value['THN_MK']]['COU
        NT'])){ $avgRating[$value['K_MK']][$value['THN_MK']]['COUNT'] =
        0;}
259
        if(!isset($avgRating[$value['K_MK']][$value['THN_MK']]['AVG
        '])){ $avgRating[$value['K_MK']][$value['THN_MK']]['AVG'] = 0;}
260
        $avgRating[$value['K_MK']][$value['THN_MK']]['SUM']
= $avgRating[$value['K_MK']][$value['THN_MK']]['SUM'] +
$value['IS_TEMPUPH'];
261
        $avgRating[$value['K_MK']][$value['THN_MK']]['COUNT'] =
$avgRating[$value['K_MK']][$value['THN_MK']]['COUNT'] + 1;
262
        $avgRating[$value['K_MK']][$value['THN_MK']]['AVG']
= $avgRating[$value['K_MK']][$value['THN_MK']]['SUM'] /
$avgRating[$value['K_MK']][$value['THN_MK']]['COUNT'];
263
    }
264
    return $avgRating;
265 }
266
267 function getIcfPredictionTableHtml($data, $title, $id){
268     $html = '';
269     $html .= '<hr/><h4>'.$title.'</h4><br/>';
270
    $html .= '<table class="table table-bordered table-hover
    table-striped datatable" id="'.$id.'">';

```

```

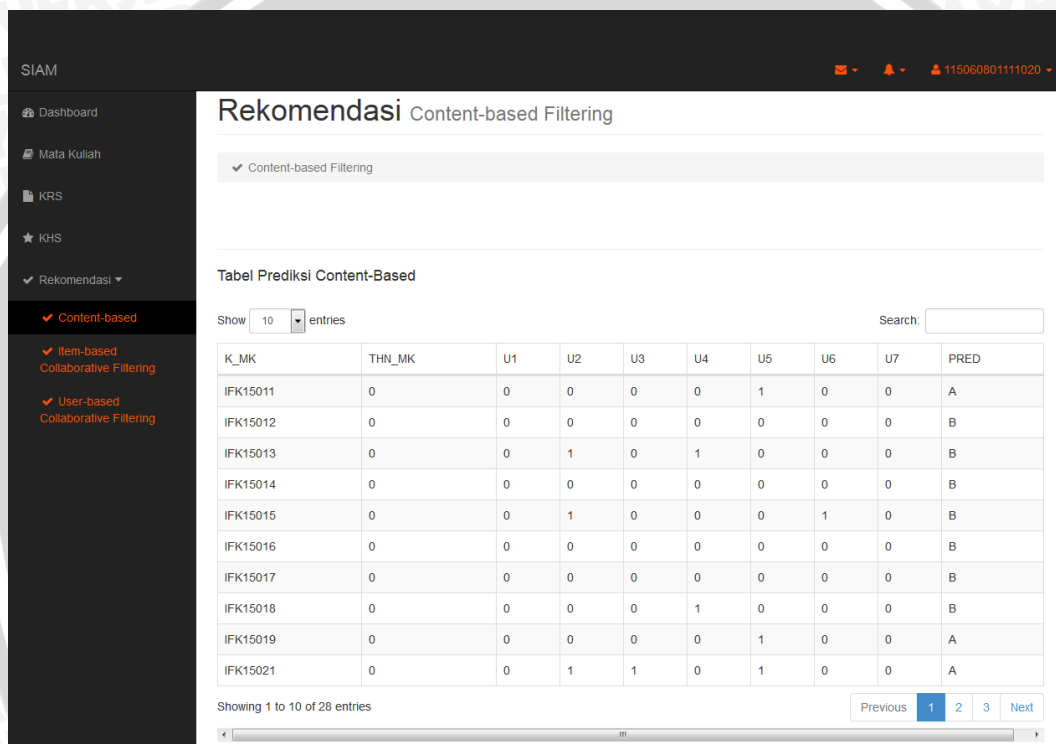
271 $html .= '<thead>';
272 $html .= '<tr>';
273 $html .= '<td>NO</td>';
274 $html .= '<td>K_MK</td>';
275 $html .= '<td>THN_MK</td>';
276 $html .= '<td>PREDIKSI</td>';
277 $html .= '<td>ACTIONS</td>';
278 $html .= '</tr>';
279 $html .= '</thead>';
280
281 $i = 0;
282 foreach ($data as $key => $val){
283     foreach ($val as $key2 => $val2){
284         $i++;
285         $html .= '<tr>';
286         $html .= '<td>'.$i.'</td>';
287         $html .= '<td>'.$key.'</td>';
288         $html .= '<td>'.$key2.'</td>';
289         $html .= '<td>'.$val2['PRED'].'</td>';
290         $html .= '<td>'.$val2['ACTIONS'].'</td>';
291         $html .= '</tr>';
292     }
293 }
294
295 $html .= '</tbody>';
296 $html .= '</table>';
297
298 return $html;
299 }

```

Kode Program 4.3 Implementasi item-based filtering

4.4 Implementasi Antarmuka

Hasil prediksi dari penelitian ini disajikan dalam tiga antarmuka. Gambar 4.1 menyajikan tampilan hasil prediksi dengan *content-based filtering* dari setiap mata kuliah yang ditawarkan beserta atribut kompetensi masing-masing. Sedangkan, prediksi nilai yang dihasilkan oleh *user-based* dan *item-based collaborative filtering* ditampilkan seperti pada Gambar 4.2 dan Gambar 4.3.



SIAM

Dashboard

Mata Kuliah

KRS

KHS

Rekomendasi

Content-based

Item-based Collaborative Filtering

User-based Collaborative Filtering

Rekomendasi Content-based Filtering

Content-based Filtering

Tabel Prediksi Content-Based

Show 10 entries

Search:

K_MK	THN_MK	U1	U2	U3	U4	U5	U6	U7	PRED
IFK15011	0	0	0	0	0	1	0	0	A
IFK15012	0	0	0	0	0	0	0	0	B
IFK15013	0	0	1	0	1	0	0	0	B
IFK15014	0	0	0	0	0	0	0	0	B
IFK15015	0	0	1	0	0	0	1	0	B
IFK15016	0	0	0	0	0	0	0	0	B
IFK15017	0	0	0	0	0	0	0	0	B
IFK15018	0	0	0	0	1	0	0	0	B
IFK15019	0	0	0	0	0	1	0	0	A
IFK15021	0	0	1	1	0	1	0	0	A

Showing 1 to 10 of 28 entries

Previous 1 2 3 Next

Gambar 4.1 Tampilan antarmuka rekomendasi dengan *content-based filtering*

SIAM

Dashboard

Mata Kuliah

KRS

KHS

Rekomendasi

- Content-based
- Item-based Collaborative Filtering
- User-based Collaborative Filtering

Rekomendasi Collaborative Filtering

User Based Collaborative Filtering

Tabel Prediksi UCF - Weighted Sum

Show entries

Search:

NO	K_MK	THN_MK	PREDIKSI	ACTIONS
18	KOM15002	0	0.094239482578779	
19	KOM15007	0	0.088725185119079	
16	IFK15038	0	0.086699872683342	
13	IFK15034	0	0.059978004122246	
20	KOM15031	0	0.058611944667273	
15	IFK15037	0	0.026375379500726	
5	IFK15015	0	0.019931348140717	
17	IFK15039	0	0.017527233691743	
2	IFK15012	0	0.014348165833042	
11	IFK15031	0	0.013687166426397	

Showing 1 to 10 of 20 entries

Previous
1
2
Next

Tabel Prediksi UCF - Herlocker

Show entries

Search:

NO	K_MK	THN_MK	PREDIKSI	ACTIONS
18	KOM15002	0	0.19636678975549	
19	KOM15007	0	0.13265873609336	
16	IFK15038	0	0.13037350922754	
13	IFK15034	0	0.10037002630932	
20	KOM15031	0	0.098999054149031	
15	IFK15037	0	0.063040338327064	
5	IFK15015	0	0.055582779959216	
17	IFK15039	0	0.052846771202365	
2	IFK15012	0	0.049572485975723	
11	IFK15031	0	0.048715897078462	

Showing 1 to 10 of 20 entries

Previous
1
2
Next

Gambar 4.2 Tampilan antarmuka rekomendasi dengan user-based collaborative filtering

SIAM

Dashboard
Mata Kuliah
KRS
KHS
Rekomendasi
Content-based
Item-based Collaborative Filtering
User-based Collaborative Filtering

Rekomendasi Collaborative Filtering

Item Based Collaborative Filtering

Tabel Prediksi ICF - Weighted Sum

Show 10 entries

Search:

NO	K_MK	THN_MK	PREDIKSI	ACTIONS
1	IFK15011	0	1	
2	IFK15012	0	1	
3	IFK15013	0	1	
4	IFK15014	0	1	
5	IFK15015	0	1	
6	IFK15016	0	1	
7	IFK15017	0	1	
8	IFK15018	0	1	
9	IFK15019	0	1	
10	IFK15021	0	1	

Showing 1 to 10 of 26 entries

Previous

1

2

3

Next

Tabel Prediksi ICF - Herlocker

Show 10 entries

Search:

NO	K_MK	THN_MK	PREDIKSI	ACTIONS
1	IFK15011	0	0.96430294259527	
2	IFK15012	0	1.0009647853353	
3	IFK15013	0	0.98986975397974	
4	IFK15014	0	0.97588036661843	
5	IFK15015	0	0.98744290485049	
6	IFK15016	0	0.95658465991317	
7	IFK15017	0	0.95658465991317	
8	IFK15018	0	0.97057404727448	
9	IFK15019	0	0.96382054992764	
10	IFK15021	0	1.0009647853353	

Showing 1 to 10 of 26 entries

Previous

1

2

3

Next

Gambar 4.3 Tampilan antarmuka rekomendasi dengan item-based collaborative filtering

BAB V

PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas berbagai tahapan pengujian dan analisis dari sistem rekomendasi pemilihan mata kuliah ini.

Pengujian sistem dibagi menjadi tiga bagian yaitu pengujian terhadap hasil rekomendasi *content-based filtering*, *item-based collaborative filtering*, dan *user-based collaborative filtering*.

Pengujian dilakukan untuk mengetahui ketepatan pemberian rekomendasi dari masing-masing metode terhadap data aktual mata kuliah yang ditempuh oleh mahasiswa. Data aktual KHS dipilih sebagai data uji dengan asumsi bahwa KHS merupakan daftar mata kuliah yang telah dipilih oleh mahasiswa dengan telah disetujui oleh dosen penasehat akademik.

Pengujian dilakukan dengan menggunakan data uji berupa KHS dari sekelompok mahasiswa yang tersimpan di dalam *database*. Dari hasil rekomendasi, akan dilakukan pengujian untuk mengetahui pengaruh jumlah semester yang telah ditempuh oleh mahasiswa dalam menghasilkan rekomendasi.

Pengujian dilakukan terhadap data KHS mahasiswa yang sedang menempuh semester keempat, kelima, keenam, ketujuh, dan kedelapan. Data latihan yang digunakan pada masing-masing pengujian adalah data KHS mahasiswa pada semester-semester sebelumnya. Sebagai contoh, untuk pengujian pada data KHS semester keempat, maka data latihan yang digunakan adalah data KHS semester pertama hingga semester ketiga.

5.1 Pengujian Pengaruh Jumlah Semester terhadap terhadap Tingkat Akurasi Hasil Prediksi *Content-Based Filtering*

Statistik data uji yang digunakan pada masing-masing skenario pengujian pengaruh jumlah semester terhadap terhadap tingkat akurasi hasil prediksi *content-based filtering* ditampilkan dalam Tabel 5.1.

Tabel 5.1 Statistik data latih dan data uji pada pengujian pengaruh jumlah semester terhadap hasil rekomendasi content-based filtering

Semester	Mahasiswa	KHS Data Uji	KHS Data Latih
4	50	343	444
5	50	313	782
6	50	383	334
7	21	113	302
8	21	45	415

Perhitungan akurasi yang pertama dilakukan dengan membandingkan kecocokan antara hasil rekomendasi dan mata kuliah yang benar-benar diprogram oleh mahasiswa tertentu pada semester yang berkesesuaian. Penghitungan jumlah kecocokan mata kuliah dilakukan dengan melakukan *query* ke basis data. Selanjutnya dilakukan perhitungan dengan menggunakan persamaan (2.10). Berikut adalah contoh perhitungan akurasi hasil rekomendasi pada semester keempat dengan data aktual pada semester yang sama:

$$Akurasi = \frac{119}{343}$$

$$Akurasi = 0.3469 = 34,69 \%$$

Tabel 5.2 menampilkan statistik kecocokan hasil rekomendasi dan mata kuliah aktual pada semester yang berkesesuaian.

Tabel 5.2 Statistik kecocokan hasil rekomendasi content-based filtering dengan data aktual pada semester yang berkesesuaian

Semester	Jumlah MK dalam KHS	Jumlah Rekomendasi Cocok	Akurasi
4	343	119	34,69 %
5	313	50	15,97 %
6	383	42	10,96 %
7	113	16	14,15 %
8	45	5	11,11 %

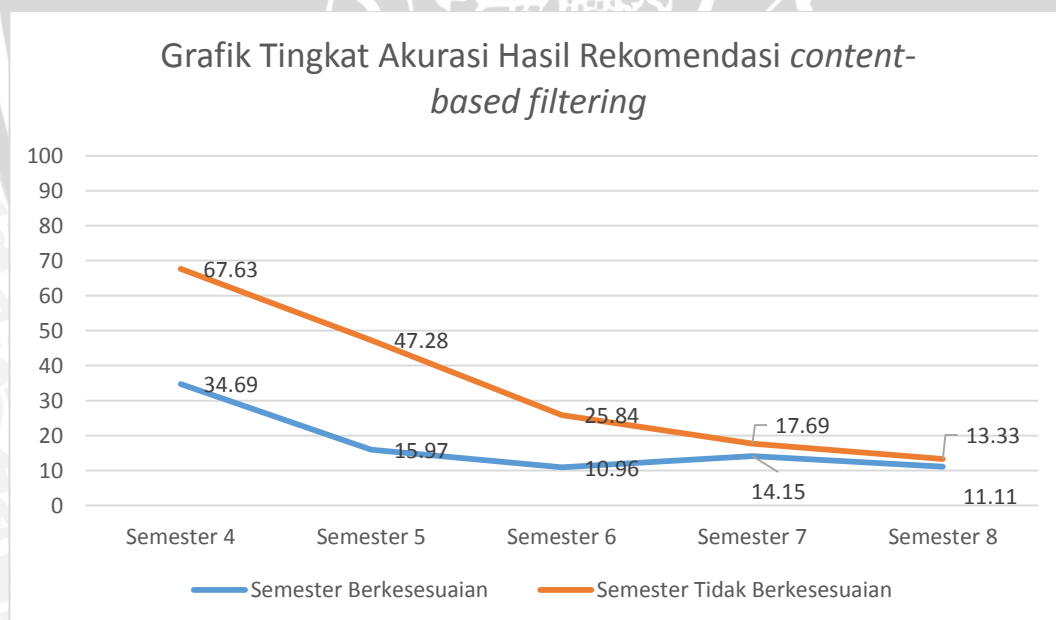
Perhitungan akurasi kedua dilakukan dengan membandingkan kecocokan antara hasil rekomendasi dan mata kuliah yang benar-benar diprogram oleh mahasiswa tertentu namun tidak harus pada semester yang berkesesuaian. Hal ini dilakukan dengan asumsi bahwa mata kuliah tertentu akan diprogram oleh

mahasiswa tertentu tidak selalu pada semester selanjutnya karena bergantung pada ketersediaan kelas dan beberapa faktor lainnya. Penghitungan jumlah kecocokan mata kuliah dilakukan dengan melakukan *query* ke basis data. Tabel 5.3 menampilkan statistik kecocokan hasil rekomendasi dan mata kuliah aktual pada semester yang tidak saling berkesesuaian.

Tabel 5.3 Statistik kecocokan hasil rekomendasi *content-based filtering* dengan data aktual pada semester yang tidak saling berkesesuaian

Semester	Jumlah MK dalam KHS	Jumlah Rekomendasi Cocok	Akurasi
4	343	232	67,63 %
5	313	148	47,28 %
6	383	99	25,84 %
7	113	20	17,69 %
8	45	6	13,33 %

Hasil pengujian pengaruh jumlah semester terhadap tingkat akurasi hasil rekomendasi *content-based filtering* juga disajikan dalam bentuk grafik pada Gambar 5.1.



Gambar 5.1 Grafik tingkat akurasi hasil rekomendasi *content-based filtering*

Dari hasil pengujian yang tersaji dalam Gambar 5.1 ditemukan bahwa nilai akurasi berkisar antara 10,98% hingga 34,69% untuk pengujian terhadap semester

yang berkesesuaian dan 13,33% hingga 67,63% untuk pengujian terhadap semester yang tidak saling berkesesuaian. Hal ini menunjukkan bahwa mata kuliah yang direkomendasikan memiliki kecenderungan untuk diprogram tidak pada semester yang aktif, melainkan pada semester-semester selanjutnya. Temuan kedua, juga ditemukan kecenderungan bahwa tingkat akurasi semakin menurun pada semester lanjut. Semestinya, akurasi akan semakin meningkat pada semester lanjut karena data latih yang digunakan semakin banyak. Analisa pertama, hal ini disebabkan oleh berkurangnya pilihan mata kuliah yang ditawarkan secara signifikan. Tabel 5.4 menampilkan rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester yang diujikan. Analisa kedua, penurunan tingkat akurasi disebabkan karena nilai mata kuliah yang ditempuh mahasiswa yang merupakan data latih yang digunakan pada metode ini semakin beragam pada semester lanjut. Tabel 5.5 menunjukkan keberagaman nilai mata kuliah yang didapatkan mahasiswa pada masing-masing semester.

Tabel 5.4 Rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester uji

Semester	Rata-rata Banyak MK
4	33,92
5	29,2
6	31,76
7	27,85
8	25,09

Tabel 5.5 Rata-rata standar deviasi data latih pada setiap semester uji

Semester	Rata-rata Standar Deviasi Data Latih
4	0,502
5	0,611
6	0,499
7	0,543
8	0,558

5.2 Pengujian Pengaruh Jumlah Semester terhadap terhadap Tingkat Akurasi Hasil Prediksi *Item-Based Collaborative Filtering*

Statistik data uji yang digunakan pada masing-masing skenario pengujian pengaruh jumlah semester terhadap terhadap tingkat akurasi hasil prediksi *item-based collaborative filtering* ditampilkan dalam Tabel 5.6.

Tabel 5.6 Statistik data latih dan data uji pada pengujian pengaruh jumlah semester terhadap hasil rekomendasi *item-based collaborative filtering*

Semester	Mahasiswa	KHS Data Uji	KHS Data Latih
4	50	292	375
5	50	328	4
6	50	377	332
7	50	250	709
8	50	111	970

Perhitungan akurasi yang pertama dilakukan dengan membandingkan kecocokan antara hasil rekomendasi dan mata kuliah yang benar-benar diprogram oleh mahasiswa tertentu pada semester yang berkesesuaian. Penghitungan jumlah kecocokan mata kuliah dilakukan dengan melakukan *query* ke basis data. Selanjutnya dilakukan perhitungan dengan menggunakan persamaan (2.10). Berikut adalah contoh perhitungan akurasi hasil rekomendasi *item-based collaborative filtering* dengan persamaan dari *herlocker* pada semester keempat dengan data aktual pada semester yang sama:

$$\text{Akurasi} = \frac{119}{292}$$

$$\text{Akurasi} = 0.4075 = 40,75 \%$$

Tabel 5.7 menampilkan statistik kecocokan hasil rekomendasi dan mata kuliah aktual pada semester yang berkesesuaian.

Tabel 5.7 Statistik kecocokan hasil rekomendasi *item-based collaborative filtering* dengan data aktual pada semester yang berkesesuaian

Semester	Weighted Sum	Herlocker
----------	--------------	-----------

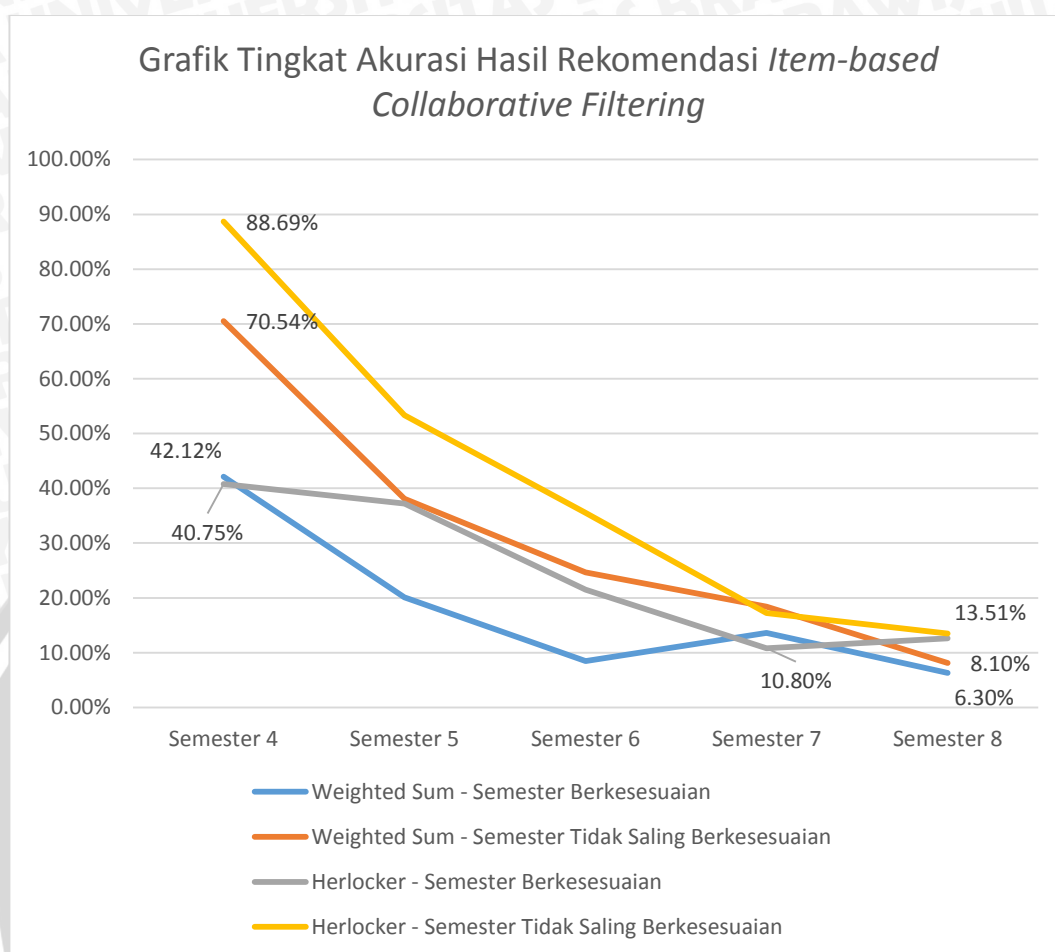
	Jumlah MK dalam KHS	Jumlah Rekomendasi Cocok	Akurasi	Jumlah Rekomendasi Cocok	Akurasi
4	292	123	42,12 %	119	40,75 %
5	328	66	20,12 %	122	37,19 %
6	377	32	8,48 %	81	21,48 %
7	250	34	13,60 %	27	10,80 %
8	111	7	6,30 %	14	12,61 %

Perhitungan akurasi kedua dilakukan dengan membandingkan kecocokan antara hasil rekomendasi dan mata kuliah yang benar-benar diprogram oleh mahasiswa tertentu namun tidak harus pada semester yang berkesesuaian. Hal ini dilakukan dengan asumsi bahwa mata kuliah tertentu akan diprogram oleh mahasiswa tertentu tidak selalu pada semester selanjutnya karena bergantung pada ketersediaan kelas dan beberapa faktor lainnya. Penghitungan jumlah kecocokan mata kuliah dilakukan dengan melakukan *query* ke basis data. Tabel 5.8 menampilkan statistik kecocokan hasil rekomendasi dan mata kuliah aktual pada semester yang tidak saling berkesesuaian.

Tabel 5.8 Statistik kecocokan hasil rekomendasi *item-based collaborative filtering* dengan data aktual pada semester yang tidak saling berkesesuaian

Semester	Jumlah MK dalam KHS	Weighted Sum		Herlocker	
		Jumlah Rekomendasi Cocok	Akurasi	Jumlah Rekomendasi Cocok	Akurasi
4	292	206	70,54 %	259	88,69 %
5	328	125	38,10 %	175	53,35 %
6	377	93	24,66 %	134	35,54 %
7	250	46	18,40 %	43	17,20 %
8	111	9	8,10 %	15	13,51 %

Hasil pengujian pengaruh jumlah semester terhadap tingkat akurasi hasil rekomendasi *item-based collaborative filtering* juga disajikan dalam bentuk grafik pada Gambar 5.2.



Gambar 5.2 Grafik pengaruh jumlah semester terhadap tingkat akurasi hasil rekomendasi *item-based collaborative filtering*

Dari hasil pengujian yang tersaji dalam Gambar 5.2 ditemukan bahwa nilai akurasi dari prediksi yang dihasilkan dengan persamaan *weighted sum* berkisar antara 6,3% hingga 42,12% untuk semester yang berkesesuaian dan 8,1% sampai 70,54% untuk semester yang tidak saling berkesesuaian. Sedangkan nilai akurasi dari prediksi yang dihasilkan oleh persamaan dari *herlocker* berkisar antara 10,80% hingga 40,75% untuk semester yang berkesesuaian dan 13,51% hingga 88,69% untuk semester yang tidak saling berkesesuaian. Temuan kedua, juga ditemukan kecenderungan bahwa tingkat akurasi semakin menurun pada semester lanjut. Semestinya, akurasi akan semakin meningkat pada semester lanjut karena data latih yang digunakan semakin banyak. Analisa pertama, hal ini disebabkan oleh berkurangnya pilihan mata kuliah yang ditawarkan secara signifikan. Tabel 5.9 menampilkan rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester

yang diujikan. Analisa kedua, pada semester lanjut, mata kuliah yang ditempuh oleh para mahasiswa semakin beragam, hal ini menyebabkan data latih yang digunakan untuk memberikan rekomendasi juga semakin beragam.

Tabel 5.9 Rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester uji

Semester	Rata-rata Banyak MK
4	34,46
5	37
6	31,84
7	28,14
8	25,48

5.3 Pengujian Pengaruh Jumlah Semester terhadap terhadap Tingkat Akurasi Hasil Prediksi *User-Based Collaborative Filtering*

Statistik data uji yang digunakan pada masing-masing skenario pengujian pengaruh jumlah semester terhadap terhadap tingkat akurasi hasil prediksi *user-based filtering* ditampilkan dalam Tabel 5.10.

Tabel 5.10 Statistik data latih dan data uji pada pengujian pengaruh jumlah semester terhadap hasil rekomendasi *user-based filtering*

Semester	Mahasiswa	KHS Data Uji	KHS Data Latih
4	20	91	109
5	20	130	3
6	20	146	133
7	20	102	279
8	20	42	393

Perhitungan akurasi yang pertama dilakukan dengan membandingkan kecocokan antara hasil rekomendasi dan mata kuliah yang benar-benar diprogram oleh mahasiswa tertentu pada semester yang berkesesuaian. Penghitungan jumlah kecocokan mata kuliah dilakukan dengan melakukan *query* ke basis data. Selanjutnya dilakukan perhitungan dengan menggunakan persamaan (2.10). Berikut adalah contoh perhitungan akurasi hasil rekomendasi *user-based collaborative filtering* dengan menggunakan persamaan *weighted sum* pada semester keempat dengan data aktual pada semester yang sama:

$$Akurasi = \frac{51}{91}$$

$$Akurasi = 0.5604 = 56,04 \%$$

Tabel 5.11 menampilkan statistik kecocokan hasil rekomendasi dan mata kuliah aktual pada semester yang berkesesuaian.

Tabel 5.11 Statistik kecocokan hasil rekomendasi user-based collaborative filtering dengan data aktual pada semester yang berkesesuaian

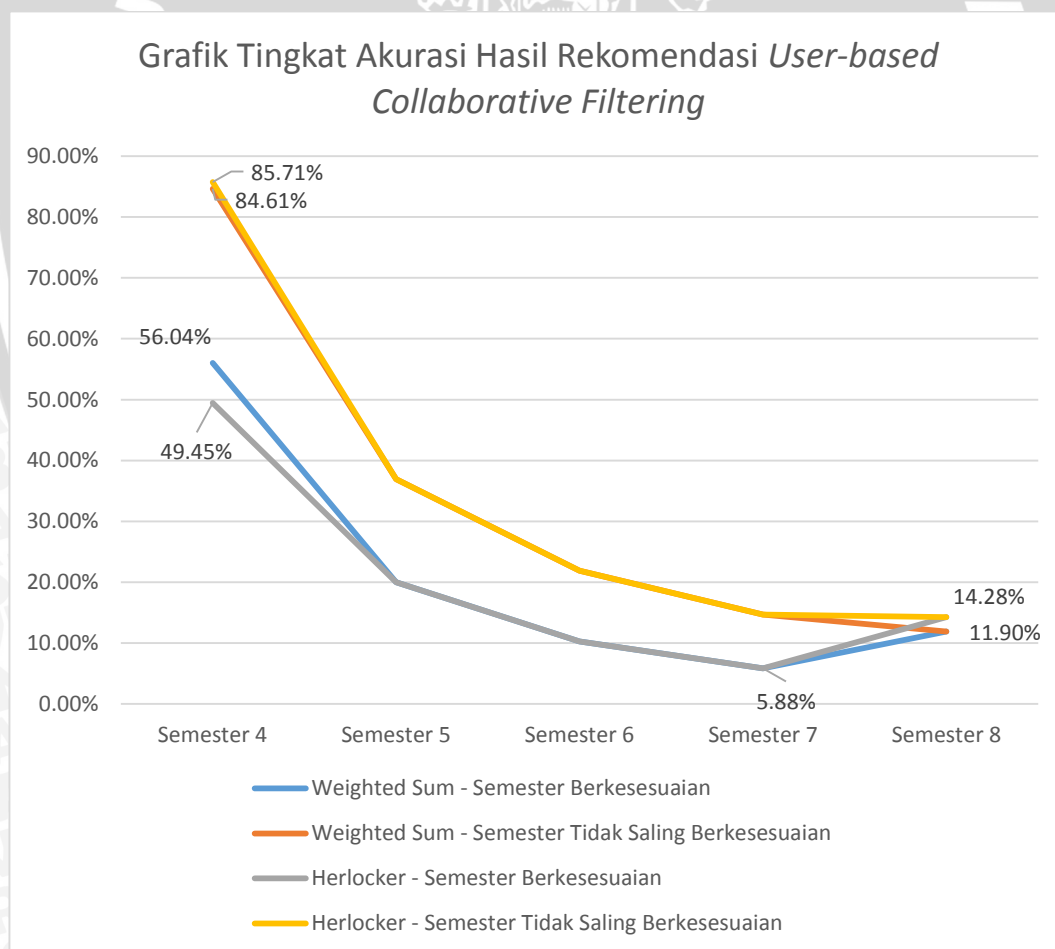
Semester	Jumlah MK dalam KHS	Weighted Sum		Herlocker	
		Jumlah Rekomendasi Cocok	Akurasi	Jumlah Rekomendasi Cocok	Akurasi
4	91	51	56,04 %	45	49,45 %
5	130	26	20,00 %	26	20,00 %
6	146	15	10,27 %	15	10,27 %
7	102	6	5,88 %	6	05,88 %
8	42	5	11,90 %	6	14,28 %

Perhitungan akurasi kedua dilakukan dengan membandingkan kecocokan antara hasil rekomendasi dan mata kuliah yang benar-benar diprogram oleh mahasiswa tertentu namun tidak harus pada semester yang berkesesuaian. Hal ini dilakukan dengan asumsi bahwa mata kuliah tertentu akan diprogram oleh mahasiswa tertentu tidak selalu pada semester selanjutnya karena bergantung pada ketersediaan kelas dan beberapa faktor lainnya. Penghitungan jumlah kecocokan mata kuliah dilakukan dengan melakukan *query* ke basis data. Tabel 5.12 menampilkan statistik kecocokan hasil rekomendasi dan mata kuliah aktual pada semester yang tidak saling berkesesuaian.

Tabel 5.12 Statistik kecocokan hasil rekomendasi *user-based collaborative filtering* dengan data aktual pada semester yang tidak saling berkesesuaian

Semester	Jumlah MK dalam KHS	Weighted Sum		Herlocker	
		Jumlah Rekomendasi Cocok	Akurasi	Jumlah Rekomendasi Cocok	Akurasi
4	91	77	84,61 %	78	85,71 %
5	130	48	36,92 %	48	36,92 %
6	146	32	21,91 %	32	21,91 %
7	102	15	14,70 %	15	14,70 %
8	42	5	11,90 %	6	14,28 %

Hasil pengujian pengaruh jumlah semester terhadap tingkat error hasil rekomendasi *user-based collaborative filtering* juga disajikan dalam bentuk grafik pada Gambar 5.3.



Gambar 5.3 Grafik pengaruh jumlah semester terhadap tingkat akurasi hasil rekomendasi *user-based collaborative filtering*

Dari hasil pengujian yang tersaji dalam Gambar 5.3 ditemukan bahwa nilai akurasi dari prediksi yang dihasilkan dengan persamaan *weighted sum* berkisar antara 5,88% hingga 56,04% untuk semester yang berkesesuaian dan 11,9% sampai 84,61% untuk semester yang tidak saling berkesesuaian. Sedangkan nilai akurasi dari prediksi yang dihasilkan oleh persamaan dari *herlocker* berkisar antara 5,88% hingga 49,45% untuk semester yang berkesesuaian dan 14,28% hingga 85,71% untuk semester yang tidak saling berkesesuaian. Temuan kedua, juga ditemukan kecenderungan bahwa tingkat akurasi semakin menurun pada semester lanjut. Semestinya, akurasi akan semakin meningkat pada semester lanjut karena data latih yang digunakan semakin banyak. Analisa pertama, hal ini disebabkan oleh berkurangnya pilihan mata kuliah yang ditawarkan secara signifikan. Tabel 5.13 menampilkan rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester yang diujikan. Analisa kedua, pada semester lanjut, mata kuliah yang ditempuh oleh para mahasiswa semakin beragam, hal ini menyebabkan data latih yang digunakan untuk memberikan rekomendasi juga semakin beragam.

Tabel 5.13 Rata-rata jumlah mata kuliah yang ditawarkan pada setiap semester uji

Semester	Rata-rata Banyak MK
4	35,2
5	37
6	31,95
7	28,3
8	25,3

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Terdapat beberapa kesimpulan yang dapat diambil setelah hasil penelitian ini selesai dilakukan, yaitu sebagai berikut:

1. Untuk dapat mengimplementasikan sistem rekomendasi diperlukan beberapa tahapan, hal yang pertama dilakukan yaitu mempersiapkan data. Pada penelitian ini data mahasiswa diperoleh dari Universitas Brawijaya berdasarkan batasan masalah yang telah ditentukan. Kemudian melakukan perhitungan rekomendasi menggunakan metode *content-based filtering* dan *collaborative filtering*, dimana *collaborative filtering* dibagi menjadi dua metode lagi yakni *item-based* dan *user-based*. Metode *content-based filtering* dapat merekomendasikan mata kuliah baru atau yang belum pernah diambil sebelumnya. Sedangkan metode *collaborative filtering* dapat merekomendasikan mata kuliah lama atau yang sudah pernah diambil sebelumnya.
2. Untuk mengetahui kualitas hasil rekomendasi, dilakukan pengujian terhadap masing-masing metode. Berdasarkan hasil pengujian, didapatkan hasil akurasi berkisar antara 10,98% hingga 34,69% untuk pengujian terhadap semester yang berkesesuaian dan 13,33% hingga 67,63% untuk pengujian terhadap semester yang tidak saling berkesesuaian dengan metode *content-based filtering*; 6,3% hingga 42,12% untuk semester yang berkesesuaian dan 8,1% sampai 70,54% untuk semester yang tidak saling berkesesuaian dengan metode *item-based collaborative filtering* dengan persamaan *weighted sum* dan 10,80% hingga 40,75% untuk semester yang berkesesuaian dan 13,51% hingga 88,69% untuk semester yang tidak saling berkesesuaian dengan metode *item-based collaborative filtering* dengan persamaan *herlocker*; 5,88% hingga 56,04% untuk semester yang berkesesuaian dan 11,9% sampai 84,61% untuk semester yang tidak saling berkesesuaian dengan metode *user-based collaborative filtering* dengan persamaan *weighted sum* dan 5,88% hingga 49,45% untuk

semester yang berkesesuaian dan 14,28% hingga 85,71% untuk semester yang tidak saling berkesesuaian dengan metode *user-based collaborative filtering* dengan persamaan *herlocker*. Hasil pengujian ini menunjukkan bahwa sistem belum dapat diimplementasikan dengan baik ke Universitas Brawijaya.

3. Berdasarkan pengujian yang dilakukan, ditemukan kecenderungan bahwa tingkat error semakin meningkat pada semester lanjut. Hal ini menunjukkan bahwa sistem rekomendasi menghasilkan hasil rekomendasi yang lebih baik ketika diimplementasikan pada semester-semester awal perkuliahan.

6.2 Saran

Terdapat beberapa saran yang dapat diberikan setelah hasil penelitian ini selesai dilakukan, yaitu sebagai berikut:

1. Perlu dilakukan penelitian lebih lanjut untuk menghasilkan sistem rekomendasi pemilihan mata kuliah yang lebih baik karena pada penelitian ini ditemukan kecenderungan tingkat akurasi yang semakin menurun pada semester lanjut.
2. Perlu dilakukan penelitian lebih lanjut dengan metode hibrid, *content-based filtering* dengan TF-IDF, *decision tree* dengan teknik selain ID3, serta berbagai teknik *collaborative filtering* untuk dibandingkan dengan kinerja sistem rekomendasi pada penelitian ini.
3. Perlu dilakukan penelitian lebih lanjut dengan *content-based filtering* dengan memperhatikan keminatan studi dan laboratorium dari mata kuliah yang ditempuh oleh mahasiswa.

Daftar Pustaka

- [RIC-11] F. Ricci, L. Rokach, dan B. Shapira. 2011, "Introduction to Recommender Systems Handbook", 1st edition, Springer Science+Business Media, London.
- [ADO-05] G. Adomavicius dan A. Tuzhilin. 2005. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering, vol 17.
- [SAR-01] B. Sarwar, G. Karypis, J. Konstan, dan J. Riedl. 2001. "Item-based Collaborative Filtering Recommendation Algorithms", WWW10, May 1-5.
- [JAN-11] D. Jannach, M. Zanker, A. Felfernig, dan G. Friedrich. 2011, "Recommender Systems: An Introduction", 1st edition, Cambridge University Press, Cambridge.
- [RAJ-11] A. Rajaraman, J. D. Ullman. 2011. "Mining of Massive Datasets", 1st edition, Cambridge University Press, Cambridge.
- [BRE-98] J. S. Breese, D. Heckerman, dan C. Kadie. 1998. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", Microsoft Research, Redmond.
- [PAZ-07] M. J. Pazzani, D. Billsus. "Content-based Recommendation Systems", Springer Berlin Heidelberg, Berlin.
- [OMA-07] O'Mahony, Michael P., and Barry Smyth. 2007. "A recommender system for on-line course enrolment: an initial study." Proceedings of the 2007 ACM conference on Recommender systems. ACM.
- [NAU-13] Naufal, Hafizh Herdi, Ade Romadhony, and Ema Rahmawati. 2013. "Perancangan dan Implementasi Sistem Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Metode Hybrid."
- [SAF-91] S. Rasoul Safavian and David Landgrebe. 1991. "A Survey of Decision Tree Classifier Methodology." IEEE Trans. Systems, Man, & Cybernetics.

[KWO-09] Kwon, Hyeong-Joon, Tae-Hoon Lee, and Kwang-Seok Hong. 2009. "Improved memory-based Collaborative filtering using entropi-based similarity measures." Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA09).

[MOO-00] R. J. Mooney dan L. Roy. 2000. "Content-Based Book Recommending Using Learning for Text Categorization". Proceedings of the Fifth ACM Conference on Digital Libraries. ACM.

[GER-10] A. Gershman, dkk. 2010. "A Decision Tree Based Recommender System". Ben-Gurion University of the Negev. Beer-Sheva.

