

APLIKASI PREDIKSI NILAI DAN REKOMENDASI MATAKULIAH MENGGUNAKAN METODE COLLABORATIVE FILTERING DAN ALGORITMA C4.5 PADA PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS X

Filbert ¹⁾ Bagus Mulyawan ²⁾ Tri Sutrisno ³⁾

¹⁾ Teknik Informatika, FTI, Universitas Tarumanagara
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia

¹⁾ email : filbert.535180020@stu.untar.ac.id, ²⁾ email : bagus@fti.untar.ac.id, ³⁾ email : tris@fti.untar.ac.id

ABSTRACT

A common problem with lectures and colleges is that students have difficulty choosing a course due to the large number of choices, or the student credits are limited due to poor course results. We hope that by designing this value prediction application and course recommendations, we will help students get the best course recommendations according to their abilities and grades. The program uses the C45 algorithm to display the selection of courses with the lowest score, and the scores generated by collaborative filtering calculations are advanced by determining the value of similarity between students and predicting grades. The system will identify each course with the highest average score / highest pass rate and it will be used as a student course recommendation. Test results show that the accuracy in determining a student's recommended course using the C45 algorithm is 70%, and the calculation error in determining a student's value prediction is 40-56%.

Key words

Student Credits, Score Prediction, Course Recommendations, C45 Algorithm, Collaborative Filtering.

1. Pendahuluan

Mahasiswa merupakan status tertinggi dan dianggap seorang yang intelek. Secara umum, mahasiswa adalah sebutan untuk seseorang yang sedang menempuh pendidikan di sebuah universitas, sekolah tinggi, ataupun akademi. Mahasiswa identik dengan kekritisan dalam berpikir dan setiap melakukan suatu tindakan. Bukan hanya sekedar akademik yang diunggulkan oleh seorang mahasiswa, akan tetapi *soft skill* termasuk dalam hal bersosialisasi dan berkomunikasi serta kontribusi nyata harus ada dalam diri mahasiswa.[1]

Masalah yang terjadi umumnya di perkuliahan atau perguruan tinggi adalah mahasiswa seringkali kebingungan ketika mendekati proses pemilihan matakuliah yang akan diambil semester selanjutnya. Mahasiswa memerlukan sebuah pedoman atau rekomendasi yang dapat digunakan sebagai acuan apabila ingin mengambil sebuah matakuliah sesuai dengan nilai semester dan minat penjurusan mahasiswa. Terkadang mahasiswa juga kesulitan mengambil matakuliah dikarenakan keterbatasan jumlah sks yang didapat setiap semester. Melalui masalah yang timbul ini, peneliti ingin merancang aplikasi berbasis web yang nantinya akan memberikan rekomendasi matakuliah kepada mahasiswa melalui hasil nilai semester mahasiswa.

Tujuan Pembuatan rancangan aplikasi prediksi dan rekomendasi matakuliah ini dibuat adalah agar melalui penggunaan Collaborative Filtering dan Algoritma C4.5 dapat memberikan hasil rekomendasi yang baik dan akurat sehingga pengambilan matakuliah mahasiswa sesuai dengan prediksi nilai dan minat penjurusan. Juga mahasiswa dapat mengetahui hasil rekomendasi matakuliah sehingga tidak bingung saat akan memilih matakuliah selanjutnya, serta membantu mahasiswa yang memiliki keterbatasan sks sehingga dapat mendapat rekomendasi matakuliah yang paling sesuai dan terakhir adalah agar mahasiswa dapat tetap menjaga kualitas nilai belajar.

Beberapa penelitian yang telah dilakukan banyak menggunakan metode Collaborative Filtering (CF) dalam memberikan rekomendasi. Hal ini dikarenakan metode CF mampu memberikan rekomendasi berdasarkan kemiripan preferensi kebiasaan user terhadap suatu *item* tertentu. Penelitian yang pernah dilakukan sebelumnya menggunakan metode CF untuk memfilter *email* yang diterima *user*. *User* akan memberikan reaksi terhadap *email* yang sudah dibaca dan selanjutnya digunakan untuk melakukan *filtering*. Kelemahan pada penelitian ini terletak pada *user experience*

yang masih sedikit sehingga data yang dihasilkan belum akurat. Penelitian lain menggunakan metode CF dalam pemberian rekomendasi untuk menganalisis hubungan antara keterampilan belajar siswa dengan hasil belajar. Peneliti melakukan perbandingan antara penggunaan metode CF tradisional (*user-based CF*) dengan metode *Hybrid CF* yang diusulkan. Hasil yang didapatkan dari penelitian tersebut adalah rekomendasi menggunakan metode yang diusulkan mampu mengurangi kesulitan siswa dalam memilih materi. Terdapat beberapa kelemahan dalam penelitian diantaranya, dibutuhkan waktu yang lama untuk melakukan prediksi materi yang direkomendasikan, belum adanya validasi terhadap keefektifan sistem rekomendasi yang dihasilkan, serta adanya pengaruh perbedaan jenis kelamin yang didominasi oleh perempuan dalam pemberian rekomendasi.

2. Dasar Teori

2.1 Sistem yang Dirancang

Aplikasi yang dirancang merupakan aplikasi yang dapat menampilkan hasil prediksi nilai mahasiswa Universitas X dengan data yang diambil yaitu data nilai mahasiswa tahun 2018 sampai dengan 2020. Aplikasi ini juga dibuat untuk memberikan rekomendasi kepada mahasiswa yang akan mengambil matakuliah pada semester yang akan datang.

Nilai mahasiswa diupload ke dalam database kemudian dilakukan perhitungan. Diawali perhitungan dengan Collaborative Filtering untuk mencari matakuliah yang unggul dengan jumlah lulus mahasiswa terbanyak, kemudian program akan memberikan rekomendasi matakuliah berdasarkan hasil algoritma C45 dan memberikan pilihan matakuliah dari hasil Collaborative Filtering.

Data yang digunakan pada aplikasi ini adalah data nilai mahasiswa yang didapat menggunakan Teknik dokumentasi yaitu Teknik mengumpulkan data melalui fakultas Teknik Informatika Universitas X. Aplikasi ini terdiri dari beberapa modul yaitu Menu *Login*, *Register*, *Home* yang terdiri dari cek daftar nilai, daftar matakuliah, kriteria nilai dan cek hasil rekomendasi.

2.2 Sistem Rekomendasi

Sistem Rekomendasi adalah sebuah bahasa yang menyediakan saran untuk *item* yang akan sesuai untuk pengguna. Saran yang diberikan berkaitan dengan berbagai pembuatan keputusan oleh pengguna (*decision making*), seperti barang yang

akan dibeli, bahasa 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.

2.3 Collaborative Filtering

Collaborative Filtering bekerja dengan membuat suatu *database* yang berisi *item preferences* dari pengguna. 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.[2]

2.3.1 User-Based Collaborative Filtering

User-based Collaborative Filtering merupakan salah satu pendekatan yang pertama kali digunakan dalam pembuatan sistem rekomendasi, dengan mengidentifikasi data-data nilai yang memiliki kesamaan atau *similar preferences*.

Kemudian dari setiap nilai yang memiliki kesamaan, didapatkan *items* yang dapat ditawarkan kepada pengguna aktif.

$$sim(a, b) = \frac{\sum_{i \in I} (R_{a,i} - \bar{R}_a)(R_{b,i} - \bar{R}_b)}{\sqrt{\sum_{i \in I} (R_{a,i} - \bar{R}_a)^2} \sqrt{\sum_{i \in I} (R_{b,i} - \bar{R}_b)^2}} \quad (1)$$

Keterangan :

$sim(a, b)$: similaritas preference pengguna a dan b

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

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

2.4 Algoritma C45

Algoritma C4.5 merupakan algoritma yang digunakan untuk membentuk pohon keputusan (*Decision Tree*). Pohon keputusan merupakan metode klasifikasi dan prediksi yang terkenal. Pohon keputusan berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel *input* dengan sebuah variabel target.

$Entropy(S)$ merupakan jumlah bit yang diperkirakan dibutuhkan untuk dapat mengekstrak suatu kelas (+ atau -) dari sejumlah data acak pada ruang sampel S . Entropy dapat dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. Entropy digunakan untuk mengukur ketidakpastian S .

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (2)$$

Keterangan :

S = Himpunan Kasus

A = Fitur

n = Jumlah partisi S

p_i = proporsi dari S_i terhadap S

$Gain(S, A)$ merupakan perolehan informasi dari atribut A *relative* terhadap output data S . Perolehan informasi didapat dari *output* data atau *variable dependent* S yang dikelompokkan berdasarkan atribut A , dinotasikan dengan $Gain(S, A)$. Untuk memilih atribut sebagai akar, didasarkan pada nilai gain tertinggi dari atribut yang ada.

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (6)$$

Keterangan:

S = Himpunan Kasus

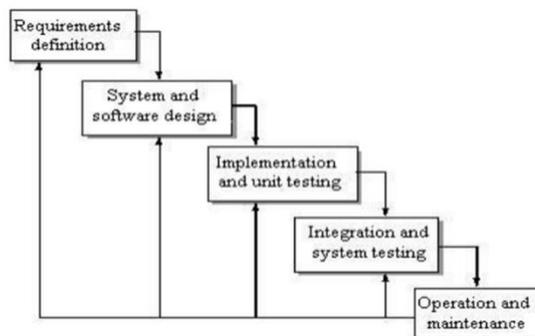
A = Atribut

n = Jumlah partisi atribut A

$|S_i|$ = Jumlah kasus pada partisi ke- i

$|S|$ = Jumlah kasus dalam S

2.5 Metode Pengembangan Aplikasi



Gambar 1. Waterfall Model

Metode perancangan software berdasarkan teori model *Waterfall* menurut Sommerville "Tahapan utama yang langsung mencerminkan dasar pembangunan kegiatan", berikut ini tahapannya:

1. Requirements Definition

Layanan, batasan, dan tujuan dari sistem ditetapkan melalui konsultasi dengan pengguna sistem. Semua itu didefinisikan secara detail dan bertindak sebagai spesifikasi sistem.

2. System and software Design

Proses desain sistem membagi kebutuhan menjadi *hardware* atau *software*. Ini menetapkan hampir seluruh perancangan sistem. Desain *software* melibatkan pengidentifikasian dan penggambaran

mengenai pemisahan dasar sistem *software* dan hubungannya.

3. Implementation and unit testing

Dalam tahap ini, *design software* adalah menyadari sebagai kumpulan program atau satuan program. Unit testing melibatkan verifikasi bahwa setiap unit telah mencapai spesifikasinya.

4. Integration and system testing

Satuan program atau kumpulan program diintegrasikan dan di tes sebagai sistem yang telah selesai, untuk menjamin bahwa kebutuhan *software* telah terpenuhi. Setelah pengujian, sistem *software* dikirimkan kepada pelanggan.

5. Operation and maintenance

Biasanya, ini adalah bagian siklus hidup *software* yang paling lama. Sistem di-install dan dimasukkan kedalam penggunaan. Pemeliharaan melibatkan pembenaran kesalahan yang tidak ditemui dalam tahap awal siklus, meningkatkan implementasi satuan sistem, dan meningkatkan layanan sistem sehubungan ditemukannya kebutuhan baru.

2.5.1 HTML

Hypertext Markup Language atau HTML adalah bahasa markup yang digunakan untuk membuat struktur halaman *website*. HTML terdiri dari kombinasi teks dan simbol yang disimpan dalam sebuah file. Dalam membuat file HTML, terdapat standar atau format khusus yang harus diikuti. Format tersebut telah tertuang dalam standar kode internasional atau ASCII (*American Standard Code for Information Interchange*).[3]

2.5.2 JavaScript

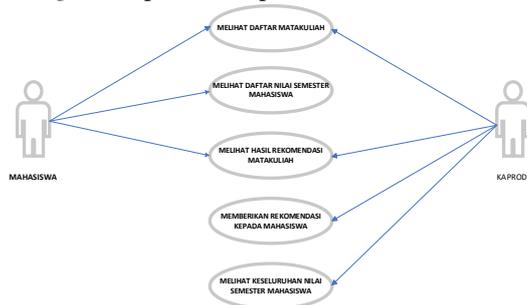
JS merupakan bahasa pemrograman jenis interpreter, sehingga tidak memerlukan compiler untuk menjalankannya. JavaScript memiliki fitur-fitur seperti berorientasi objek, *client-side*, *high-level programming*, dan *loosely typed*. Pada tahun 1994 JavaScript mulai dikenal, pada saat itu web dan internet sudah mulai berkembang. JavaScript didesain oleh *Brendan Eich* yang merupakan karyawan Netscape. Transformasi nama JavaScript, dimulai dari Mocha, Mona, LiveScript, hingga akhirnya resmi bernama JavaScript. [4]

3. Pembuatan

3.1 Use Case Diagram

Use Case Diagram untuk mengetahui akses apa saja yang diberikan kepada mahasiswa. Aplikasi prediksi predikat mahasiswa memiliki 2 *use case*

yaitu *use case mahasiswa* dan *use case admin*. *Use Case Diagram* dapat dilihat pada **Gambar 2**.



Gambar 2. Use Case Diagram

Penjelasan mengenai masing-masing use case sebagai berikut:

1. *Login* ke Aplikasi

Login ke dalam website agar dapat akses fitur. Tampilan mahasiswa diawali dengan memasukkan nama dan password. Apabila belum ada, dapat mendaftar terlebih dahulu. Setelah login, terdapat pilihan untuk cek nilai mahasiswa itu sendiri kemudian terdapat pilihan untuk cek hasil rekomendasi mahasiswa.

2. Melihat Daftar Matakuliah

Fitur menampilkan daftar matakuliah terhadap matakuliah yang diambil mahasiswa tersebut. Dapat diakses mahasiswa.

3. Cek Hasil Rekomendasi

Bagian ini merupakan bagian utama yang ditujukan kepada mahasiswa agar tidak salah mengambil matakuliah selanjutnya. System dibuat dengan menyesuaikan kemampuan mahasiswa dalam matakuliah yang sedang diambil.

3.2 Data Flow Diagram

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data diantara komponen-komponen tersebut, asal, tujuan dan penyimpanan dari data tersebut.

DFD yang dirancang adalah Diagram Konteks, *DFD Level 1*, *DFD Login* dan *DFD Aplikasi Prediksi Nilai*.



Gambar 3. Diagram Konteks

Diagram konteks berfungsi untuk menetapkan konteks serta batasan-batasan sistem pada sebuah pemodelan. Salah satu tugasnya yaitu untuk melakukan penyimpanan data eksternal lain.

Proses masing-masing tiap alur program dijelaskan sebagai berikut:

1. Proses Memasukan Nama Dan Password

Agar dapat menggunakan aplikasi tersebut, user diwajibkan mendaftar terlebih dahulu dengan memasukan Nama dan Password. Apabila sudah ada dapat melakukan login.

2. Proses Cek Nilai

Mahasiswa dapat melakukan pengecekan nilai. Hanya saja, Login sebagai mahasiswa hanya menampilkan nilai mahasiswa itu sendiri.

3. Proses Hasil Rekomendasi

Sistem rekomendasi akan memberikan pilihan matakuliah selanjutnya yang dapat diambil mahasiswa dan juga dengan menyesuaikan hasil nilai mahasiswa.

4. Hasil Pengujian

4.1 Cara Pengujian

Proses pengujian aplikasi dilakukan 3 (tiga) kali tahap pengujian, yaitu tahap pengetesan (testing) oleh pembuat program sendiri, setelah dilakukan pengetesan oleh pembuat program, pengujian berikutnya akan dilakukan oleh beberapa mahasiswa yang penulis pilih secara acak untuk mengetahui umpan balik, dan masalah-masalah yang terjadi mengenai aplikasi ini, kemudian yang terakhir adalah pengujian yang akan dilakukan oleh dosen universitas. Pengujian dilakukan untuk memastikan bahwa semua fungsi dapat berjalan dengan baik dan perhitungan yang didapat merupakan hasil final.

4.1.1 Unit Testing

Unit Testing merupakan metode pengujian untuk melakukan pengujian terhadap fungsi-fungsi tombol aksi yang dimiliki oleh aplikasi yang dirancang. Pengujian dilakukan oleh *programmer* yang bertanggung jawab atas perancangan aplikasi ini.

4.1.2 Internal Testing

Internal Testing merupakan pengujian yang dilakukan untuk melihat proses alur komponen aplikasi yang dirancang.

4.1.3 UAT (User Acceptance Test)

UAT merupakan pengujian yang dilakukan oleh *user* langsung yang menggunakan aplikasi tersebut. Aplikasi ini dilakukan pengujian oleh beberapa teman sebagai *user* yang akan menggunakan aplikasi rekomendasi matakuliah ini.

4.2 Hasil Pengujian Unit Testing

Setelah dilakukan pengujian unit testing menghasilkan hasil sebagai berikut:

Tabel 1 Hasil Pengujian Unit Testing

Modul	Aksi	Hasil
Get Started	Tombol Halaman Login	OK
Login	Tombol Login	OK
	Tombol Register	OK
Register	Tombol Register	OK
Center Button : Daftar Matakuliah	Tombol Menampilkan Matakuliah	OK
Center Button : Daftar Rekomendasi Matakuliah	Tombol Menampilkan Rekomendasi Matakuliah	OK
Center Button : Cek Hasil Rekomendasi	Tombol Menampilkan Hasil Rekomendasi	OK
	Tombol Kembali	OK
Center Button : Logout Button	Tombol Logout / Keluar	OK

4.3 Hasil Pengujian Unit Testing

Tabel 2 Hasil Pengujian Unit Testing

Modul	Alur	Hasil
Home	Halaman Utama	OK
Login	Login	OK
Register	Registrasi	OK
Tampilan Mahasiswa	Melihat Pilihan Menu	OK
Center Button : Daftar Rekomendasi Matakuliah	Menampilkan Daftar Rekomendasi Matakuliah	OK
Center Button : Cek Hasil Rekomendasi	Menampilkan Nilai Mahasiswa beserta Hasil Rekomendasi	OK
Kembali	Kembali Ke Halaman Sebelumnya	OK
Center Button : Logout Button	Kembali ke Halaman Utama	OK

4.4 Pengujian Metode C45

Pengujian metode prediksi nilai dan rekomendasi matakuliah dilakukan dengan *confusion matrix*. Untuk rumus perhitungan dapat dilihat pada dibawah ini:

$$Accuracy = \frac{TP+TN}{P+N} \quad (3)$$

Keterangan :

TP = Jumlah data *true* positif

TN = Jumlah data *true* negative

P = Jumlah data Positif yang sebenarnya

N = Jumlah data negative yang sebenarnya

Untuk metode *confusion matrix* dilakukan dengan mengambil 20% data asli untuk diuji. Kemudian data diklasifikasi menjadi data uji *true*, data uji *false*, data hasil *true* dan data hasil *false*. Berikut adalah *table* yang sudah diklasifikasi:

Tabel 1 Hasil Pengujian Akurasi

	Hasil True	Hasil False
Direkomendasikan	7	7
Tidak Direkomendasikan	3	3

$$Accuracy = \frac{7+7}{10+10} = \frac{14}{20} \times 100\% = 70\%$$

4.2 Hasil Pengujian Collaborative Filtering

Perhitungan prediksi nilai dengan menggunakan Collaborative Filtering menghasilkan nilai MAE (*Mean Absolute Error*). MAE merupakan tingkat *error* kesalahan perhitungan. Tingkat *error* perhitungan ditampilkan sebagai berikut:

Tabel 2 Hasil Mean Absolute Error

MAE (2)	TOTAL
MHS 1	64.83
MHS 2	46.00
MHS 3	58.67
MHS 4	72.67
MHS 5	60.17
MHS 6	56.00
MHS 7	41.00
MHS 8	71.33
MHS 9	59.33
MHS 10	56.33
MHS 11	79.59
MHS 12	86.32
MHS 13	79.59
MHS 14	44.59

5. Kesimpulan dan Saran

5.1 Kesimpulan

Pembuatan aplikasi prediksi nilai dan rekomendasi matakuliah bertujuan untuk membantu mahasiswa supaya mendapatkan rekomendasi matakuliah yang terbaik yang dapat diambil mahasiswa dengan mempertimbangkan nilai setiap semester mahasiswa. Melalui pembuatan aplikasi ini, dapat disimpulkan bahwa:

1. Aplikasi ini masih mempunyai kekurangan sehingga perlu diperbaiki dan dikembangkan seiring berjalan waktu.
2. Aplikasi ini diharapkan mempermudah mahasiswa mendapatkan rekomendasi matakuliah, terlebih terhadap mahasiswa dengan keterbatasan sks.

5.2 Saran

Berdasarkan hasil pembuatan dan pengetesan aplikasi Prediksi Nilai dan Rekomendasi Matakuliah Mahasiswa dapat diberikan saran sebagai berikut :

1. Perancangan aplikasi dengan topik ini dapat dilakukan dengan metode yang lainnya.
2. Pembuatan aplikasi ini dapat lebih baik lagi apabila terdapat sumber maupun contoh perhitungan.

REFERENSI

- [1] Hayati, Ika Umi., 15 April 2022, “Perguruan Tinggi: Pengertian, Tujuan, Jenis, dan Bentuknya”, <https://calonmahasiswa.com/perguruan-tinggi/>.
- [2] Putra, Hananta, Fitri, Aditya., Mahmudy, Firdaus, Wayan., Setiawan, Darma, Budi., “Sistem Rekomendasi Mata Kuliah Pilihan Mahasiswa Dengan Content-Based Filtering Dan Collaborative Filtering (Studi Kasus: Universitas Brawijaya)”, Jurnal Universitas Brawijaya, Vol. V, Nomor 17, (Desember, 2015), h. 3.
- [3] Aprilia, Putri., 12 April 2022, “Pengertian HTML, Fungsi dan Cara Kerjanya”, <https://www.niagahoster.co.id/blog/html-adalah/>.
- [4] Dicoding Intern, 11 April 2022, “Apa itu JavaScript? Fungsi dan Contohnya”, <https://www.dicoding.com/blog/apa-itu-javascript-fungsi-dan-contohnya/>.

Filbert, Mahasiswa S1, Program Studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara.

Bagus Mulyawan S.Kom., M.M, Dosen Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara.

Tri Sutrisno , S.Si., M.Sc Dosen Program Studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara.