

PENGEMBANGAN APLIKASI PENGUKURAN CAPAIAN PEMBELAJARAN MAHASISWA UNTUK REKOMENDASI MATA KULIAH PEMINATAN MENGGUNAKAN ALGORITMA DECISION TREE

Syahrul Ramadhan¹, Bagus Mulyawan², Tri Sutrisno³

^{1,2,3} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara
Jln. Letjen S Parman No. 1, Jakarta 11440 Indonesia
E-mail : ¹syahrul.535180022@stu.untar.ac.id, ²bagus@fti.untar.ac.id, ³tris@fti.untar.ac.id

Abstrak

Pengukuran capaian pembelajaran mahasiswa sangat diperlukan untuk proses penilaian yang sesuai dan juga untuk keperluan lain seperti akreditasi. Sistem pemberian rekomendasi matakuliah peminatan juga sangat diperlukan untuk mahasiswa agar tidak terhambat dan lulus tepat pada waktunya. Untuk itu diperlukan sebuah sistem yang secara khusus dapat memberikan informasi-informasi tersebut. Perancangan aplikasi ini dilakukan dengan tujuan memberikan informasi-informasi yang dibutuhkan untuk melihat capaian mahasiswa dan merekomendasikan matakuliah peminatan yang sesuai berdasarkan data mahasiswa tersebut. Proses rekomendasi yang dilakukan menggunakan algoritma decision tree. Rancangan aplikasi ini dibuat berbasis website dengan menggunakan JavaScript sebagai backend dan MongoDB sebagai pengaturan sistem basis data yang digunakan. Untuk mengukur akurasi metode digunakan confusion matrix dengan mengambil dua puluh persen sampel data untuk dibandingkan dengan hasil dari algoritma yang digunakan dan nilai akurasi yang didapatkan cukup baik yaitu sekitar 75% dan 87,5% untuk 40 persen data. Hasil akhir untuk aplikasi ini adalah bahwa aplikasi pengukuran capaian pembelajaran dan rekomendasi matakuliah sudah cukup baik dalam memberikan informasi capaian mahasiswa dan rekomendasi matakuliah peminatan secara otomatis.

Kata Kunci — *Capaian Pembelajaran, Rekomendasi, Decision Tree, Confusion Matrix, backend*

Abstract

Measurement of student learning achievement is very necessary for the appropriate assessment process and also for other purposes such as accreditation. The system for providing recommendations for specialization courses is also very necessary for students to graduate on time. For that we need a system that can specifically provide this information. The recommendation process is carried out using a decision tree algorithm. The design of this application is made based on a website using JavaScript as the backend and MongoDB as the database system settings used. To measure the accuracy of the method, a confusion matrix is used by taking twenty percent of the data sample to be compared with the results of the algorithm used and the accuracy value obtained is quite good, which is around 75% and 87.5% for 40 percent of the data. The final result for this application is that the application of measuring learning achievement and course recommendations is good enough to provide information on student achievements and recommendations for specialization courses automatically.

Keywords — *Learning Achievement, recommendations, Decision Tree, Confusion Matrix, backend*

1. PENDAHULUAN

Mahasiswa adalah sebutan bagi orang yang menempuh pendidikan tinggi disebuah sekolah tinggi akademi, dan yang paling umum adalah universitas. Dalam sistem pendidikan, mahasiswa adalah aset penting bagi sebuah institusi pendidikan, untuk itu perlu diperhatikan faktor-faktor yang menjadi penyebab kelulusan mahasiswa tidak tepat pada waktunya. Periode tahun pertama perkuliahan sering dianggap sebagai masa yang paling menantang bagi mahasiswa, di mana pada masa ini mahasiswa akan menemui berbagai situasi yang baru seperti sistem perkuliahan, dan materi perkuliahan yang lebih sulit. [1] Oleh sebab itu kesalahan dalam memilih mata kuliah dapat menjadi salah satu faktor terhambatnya kelulusan mahasiswa, yang mana hal ini semestinya bisa diatasi dengan baik melalui sistem yang dapat mengukur capaian pembelajaran mahasiswa dan dapat secara otomatis merekomendasikan mata kuliah yang sesuai berdasarkan data capaian pembelajaran tersebut. Oleh sebab itu perlu adanya klasifikasi khusus yang dapat mengukur capaian pembelajaran mahasiswa secara tepat dan akurat.

Kriteria akreditasi fakultas saat ini juga membutuhkan prosedur penilaian capaian pembelajaran yang efektif dengan hasil yang terdokumentasi dengan baik agar mempermudah proses penilaian akreditasi dan juga untuk keperluan lainnya, salah satunya untuk merekomendasikan mata kuliah peminatan. Untuk mencapai hal tersebut, diperlukan metode yang efektif untuk memperkirakan kinerja siswa. Dengan demikian, metode penilaian langsung telah diidentifikasi dan dianggap cocok untuk dinilai secara langsung pengetahuan khusus siswa selama ujian atau melalui pengamatan keterampilan. Melalui penilaian langsung yang dilakukan tersebut, penilaian yang dilakukan akan menjadi landasan untuk mengukur capaian pembelajaran yang dilakukan pada perancangan ini.

Perancangan aplikasi ini dilakukan dengan tujuan memberikan informasi-informasi yang dibutuhkan untuk melihat capaian pembelajaran mahasiswa dan merekomendasikan matakuliah peminatan yang sesuai berdasarkan data mahasiswa tersebut. Untuk mempermudah pengukuran capaian pembelajaran tersebut, diperlukan sebuah sistem yang secara otomatis dapat mengukur capaian pembelajaran pada mahasiswa. Sistem ini akan dijalankan melalui program yang akan dibangun. Program ini nantinya dapat mengukur capaian pembelajaran mahasiswa secara efisien karena penilaian dilakukan secara otomatis sehingga lebih menghemat waktu dan juga tenaga. Melalui data capaian pembelajaran ini, akan dilakukan klasifikasi untuk merekomendasikan mata kuliah peminatan yang cocok untuk mahasiswa tersebut. Metode klasifikasi akan dilakukan menggunakan algoritma *decision tree*. Algoritma *decision tree* ini dipilih karena dapat mengklasifikasikan data secara akurat dibandingkan dengan algoritma lain.

Decision tree ini adalah salah satu metode klasifikasi yang cukup sering digunakan pada beberapa jurnal perancangan untuk rekomendasi. Algoritma ini sering digunakan karena mudah untuk diinterpretasi. Konsep dari pohon keputusan adalah mengubah data menjadi *decision tree* dan aturan-aturan keputusan. Manfaat utama dari penggunaan *decision tree* adalah kemampuannya untuk memecah proses pengambilan keputusan yang kompleks menjadi lebih sederhana, sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan yang sedang terjadi. Kelebihan lain dari metode ini adalah mampu meminimalisir data-data yang tidak diperlukan dengan cara mengeleminasi data yang tidak dibutuhkan. Beberapa penelitian menunjukkan tingkat akurasi yang lebih tinggi dibandingkan dengan algoritma yang lain.[2] melalui jurnal ilmiah ini perancang memutuskan untuk menggunakan metode *decision tree* untuk klasifikasi data. Untuk mengimplementasikan proses klasifikasi tersebut, akan dirancang sebuah program yang dapat menampung data nilai mahasiswa dan mengklasifikasikannya menjadi capaian pembelajaran mahasiswa. Selain itu aplikasi ini juga dapat merekomendasikan mata kuliah peminatan yang memiliki kemungkinan lulus lebih tinggi

dibandingkan dengan matakuliah lain, untuk diambil oleh mahasiswa yang bersangkutan berdasarkan nilai yang dianalisa.

2. METODE PENELITIAN

2.1 Sistem Yang Dirancang

Aplikasi yang dibuat merupakan aplikasi dengan sistem pengukuran capaian pembelajaran serta Analisa capaian belajar dengan beberapa kriteria penilaian untuk rekomendasi mata kuliah peminatan yang sesuai. Hal ini bertujuan agar memudahkan pihak kampus untuk memberikan penilaian capaian mahasiswa dan merekomendasikan mata kuliah peminatan yang sesuai dari hasil Analisa algoritma *decision tree*. Aplikasi ini sendiri merupakan aplikasi berbasis *web* dengan metode pengembangan SDLC *waterfall*. Aplikasi ini berfokus pada pemberian informasi kepada mahasiswa disertai dengan analisa-analisa yang dilakukan terhadap beberapa kriteria data yang ditampilkan. Proses pertama yang berjalan pada sistem yang dirancang ini adalah Analisa data yang dimasukkan. Data tersebut dimasukkan oleh admin yang berisikan beberapa informasi yang diperlukan untuk proses Analisa. Data tersebut adalah data mahasiswa, data mata kuliah, data dosen dan data nilai mahasiswa. Data nilai tersebut dibagi menjadi beberapa kriteria penilaian untuk keperluan Analisa yaitu, nilai presentasi, nilai ujian praktek, nilai ujian akhir dan kriteria nilai lainnya. Proses selanjutnya adalah menampilkan hasil Analisa pada data yang dilakukan oleh sistem. Hasil Analisa untuk capaian pembelajaran akan ditampilkan dalam bentuk tabel, kemudian untuk hasil Analisa untuk rekomendasi mata kuliah peminatan akan ditampilkan melalui tabel dan juga diagram data yang menampilkan persentase peluang kelulusan pada mata kuliah peminatan yang dapat diambil.

2.2 JavaScript

JavaScript membuat program mudah diakses. Tidak hanya setiap orang yang memiliki *browser*, tetapi hanya dengan sedikit usaha, mereka dapat melihat JavaScript yang mengaktifkan situs *web*. [3] JavaScript adalah Bahasa pemrograman yang biasanya digunakan untuk menambah nilai estetika tampilan pada situs yang dibuat. Konten yang ditambahkan juga bisa bersifat dinamis atau dapat bergerak tanpa *reload* halaman secara langsung, hal tersebut berguna agar situs menjadi lebih hidup saat digunakan. Selain untuk nilai estetika, penulis juga akan menggunakan Bahasa pemrograman ini untuk menjalankan *web server* dengan dipadukan NodeJS. Cara kerja javascript pada sisi *client* tidak terlalu rumit. Ketika pengguna membuka halaman *web*, *script* akan otomatis terunduh dan dapat dijalankan langsung oleh *browser*. Karena proses berjalannya *script* ini tidak melalui *server*, ini membuat pengguna membutuhkan *browser* yang mendukung JavaScript. Sedangkan dalam sisi *server*, JavaScript umumnya digunakan untuk kebutuhan basis data. Proses yang terjadi Ketika program berjalan disisi *server*, program tersebut tidak akan berhenti pada satu proses saja. Sistem akan menyiapkan proses berikutnya untuk dijalankan, keunggulan ini yang membuat perancangan *web* menggunakan JavaScript menjadi lebih interaktif.

2.3 Metode Decision Tree

Decision Tree adalah teknik sederhana namun memiliki tingkat keberhasilan yang tinggi untuk memprediksi dan menjelaskan hubungan antara beberapa pengukuran tentang *item* dan target nilainya. [4] Perancangan ini menggunakan algoritma *decision tree* untuk memberikan rekomendasi dengan menganalisa data kriteria yang ada. Algoritma ini memiliki dua jenis keputusan. Pertama *decision tree* variabel kategori yang adalah sebuah pohon keputusan yang variabel target berupa sebuah kategori untuk dicari kategori yang tepat untuk dipilih. Selanjutnya adalah *decision tree* variabel kontinu yaitu, sebuah pohon keputusan dengan variabel target kontinu seperti usia, nilai dan sebagainya. Pada perancangan ini dipilih algoritma C4.5. Algoritma

ini dipilih karena dapat melakukan rekomendasi dengan memberikan tingkat nilai akurasi yang ideal untuk merekomendasikan matakuliah peminatan kepada mahasiswa.

2.4 Algoritma C4.5

Algoritma C4.5 merupakan algoritma yang biasanya digunakan untuk klasifikasi data. Algoritma ini memiliki atribut-atribut numerik dan bisa juga berupa kategori. Banyak yang menganggap C4.5 identik dengan J48, perlu dicatat bahwa ada beberapa perbedaan. Prosesnya sendiri pertama menentukan atribut yang menjadi akar. Setelah itu buat cabang untuk setiap nilai, kemudian bagi kasus untuk setiap cabang yang telah dibuat. Terakhir ulangi proses sampai kasus pada setiap cabang memiliki kelas yang sama. [5] Perincian algoritma tersebut akan dijabarkan dibawah ini:

1. Menghitung jumlah kasus
Langkah pertama adalah menghitung jumlah kasus, hal ini dilakukan untuk mempermudah proses pembuatan pohon keputusan dengan membagi atau mengkategorikan data menjadi beberapa kasus.
2. Menghitung semua *entropy* dari semua kasus
Entropy adalah ukuran dari teori informasi yang dapat mengetahui karakteristik dari kumpulan data. Dari nilai *entropy* tersebut kemudian dihitung nilai *information gain* masing-masing atribut setelah menentukan jumlah kasus. Untuk menghitungnya dapat menggunakan rumus sebagai berikut:

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

S: Himpunan Kasus

N: Jumlah partisi S

p_i : Proporsi dari S_i terhadap S

3. Lakukan perhitungan *gain* untuk semua atribut
information gain adalah informasi yang didapatkan dari perubahan *entropy* pada suatu kumpulan data. Setelah didapat entropi dilanjutkan dengan menghitung *gain* untuk setiap atribut. Untuk menghitungnya dapat menggunakan rumus sebagai berikut:

$$Gain (S,A) = Entropy (S) - \sum_{i=1}^n * Entropy (S_i) \quad (2)$$

S: Himpunan

A: Atribut

N: Jumlah Partisi Atribut A

$|S_i|$: Jumlah Kasus pada partisi ke 1

$|S|$: Jumlah Kasus dalam S

4. Ubah menjadi pohon keputusan
Setelah menghitung, nilai *Gain* tertinggi akan menjadi akar pohon keputusan dan *node* akarnya menjadi cabang dari akar tersebut. Proses ini terus dilakukan berulang sampai seluruh kasus dari setiap cabang memiliki kelas yang sama.

2.5 Pengukuran Capaian Pembelajaran

Pengukuran capaian pembelajaran ini menjadi salah satu fitur utama pada rancangan yang dibuat. Pengukuran ini dilakukan dengan menganalisa beberapa variabel data pada tabel data nilai mahasiswa. Variabel tersebut digunakan sebagai kriteria penilaian dikarenakan sesuai dan mudah untuk diimplementasikan kedalam perancangan ini. Proses ini dimulai dengan admin atau dosen melakukan *input* data nilai pada aplikasi ini. Data nilai tersebut terdiri dari beberapa kriteria penilaian. Setelah itu data tersebut akan diakumulasi dan dijadikan satu bentuk nilai poin. Nilai poin hasil dari akumulasi nilai tadi akan dikategorikan Kembali menjadi sebuah indikator pencapaian pembelajaran mahasiswa. Pada perancangan ini penulis akan membuat lima indikator

untuk capaian belajar mahasiswa yaitu, A, B, C, D, dan E. Adapun sampel data yang akan digunakan akan dijelaskan lebih lanjut pada subbab berikutnya.

2.6 Rekomendasi Mata Kuliah

Fitur ini merupakan salah satu fitur utama pada rancangan ini. Proses rekomendasi dilakukan dengan cara menganalisa variabel data nilai mata kuliah tertentu yang memiliki hubungan dengan mata kuliah peminatan pada semester atas. Proses ini dimulai dengan admin atau dosen memasukan data nilai pada aplikasi. Setelah data dimasukan selanjutnya data akan disimpan dan diproses secara otomatis untuk dilakukan Analisa terhadap data nilai mata kuliah tersebut. Setelah proses analisa selesai, selanjutnya aplikasi akan menampilkan daftar mata kuliah peminatan yang memiliki peluang lulus lebih tinggi yang dapat diambil oleh mahasiswa.

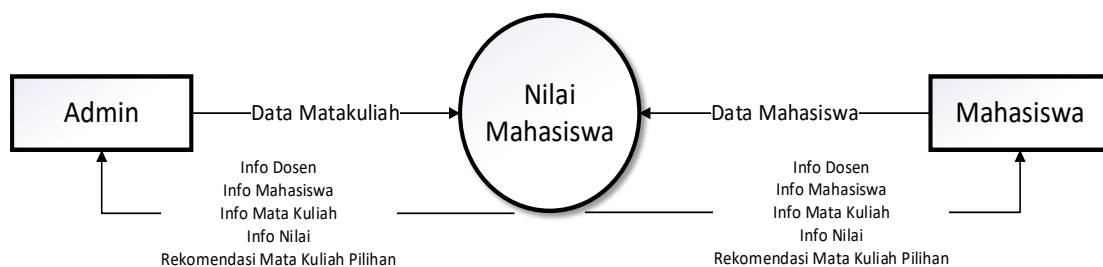
3. HASIL DAN PEMBAHASAN

3.1 Perancangan Sistem Aplikasi

Dalam merancang sebuah sistem, diperlukan gambaran baik dari segi desain maupun struktur berjalannya sebuah sistem agar sistem yang dirancang dapat sesuai dengan kriteria yang dibutuhkan. Desain dan struktur yang dibuat memiliki tujuan agar mempermudah proses perancangan yang dibuat. Proses perancangan ini bermulai dari proses pembuatan DFD (*Data Flow Diagram*) untuk mengetahui aliran data yang terjadi pada sistem yang dibuat. DFD ini akan mempermudah proses pembuatan *Use case* dan desain pada aplikasi yang dibuat. Proses selanjutnya adalah pembuatan *use case* dan juga ERD (*Entity Relationship Diagram*). *Use case* ini dibuat untuk memberikan gambaran proses yang dapat dilakukan oleh setiap entitas, yang mana proses tersebut juga digambarkan pada hubungan setiap entitas yang dapat dilihat melalui ERD yang dibuat. Selanjutnya adalah proses pembuatan *mockup* dan deskripsi alur penggunaan aplikasi. Kedua hal tersebut dibuat dengan tujuan memberikan gambaran singkat mengenai desain aplikasi yang akan dirancang dan juga tata cara menggunakan aplikasi tersebut. Terakhir adalah perancangan tabel data agar dapat digunakan sebagai acuan dalam membuat basis data. Tabel data ini dibuat berdasarkan hasil Analisa kebutuhan data pada tahap pertama perancangan ini. Desain dan struktur diatas akan dijelaskan secara lengkap pada subbab berikutnya.

3.2 Diagram Konteks

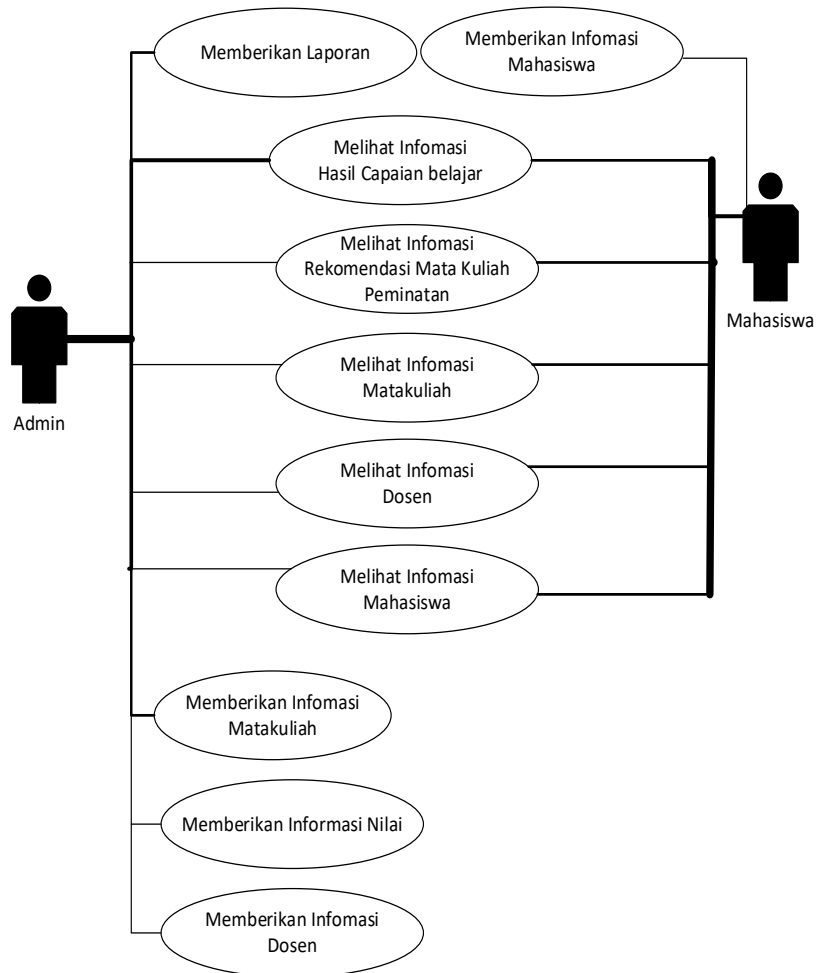
Pada subbab ini dijelaskan bagaimana proses rancangan dan juga aliran data pada aplikasi pengukuran capaian pembelajaran. Pada diagram konteks yang dirancang terdapat beberapa beberapa *user* yang dapat melihat atau menyunting data yang ada. Pertama mahasiswa, pada rancangan ini mahasiswa dapat melihat informasi nilai, capaian pembelajarannya dan juga dapat melihat rekomendasi mata kuliah peminatan yang sesuai dengan hasil Analisa data nilai mahasiswa tersebut. Selanjutnya Dosen, pada aplikasi ini dosen dapat memasukan nilai mata kuliah. Terakhir Admin, pada aplikasi ini admin dapat menyunting dan memasukan data mahasiswa maupun dosen. Berikut diagram konteks proses tersebut:



Gambar 1 Diagram Konteks

3.3 Use Case Diagram

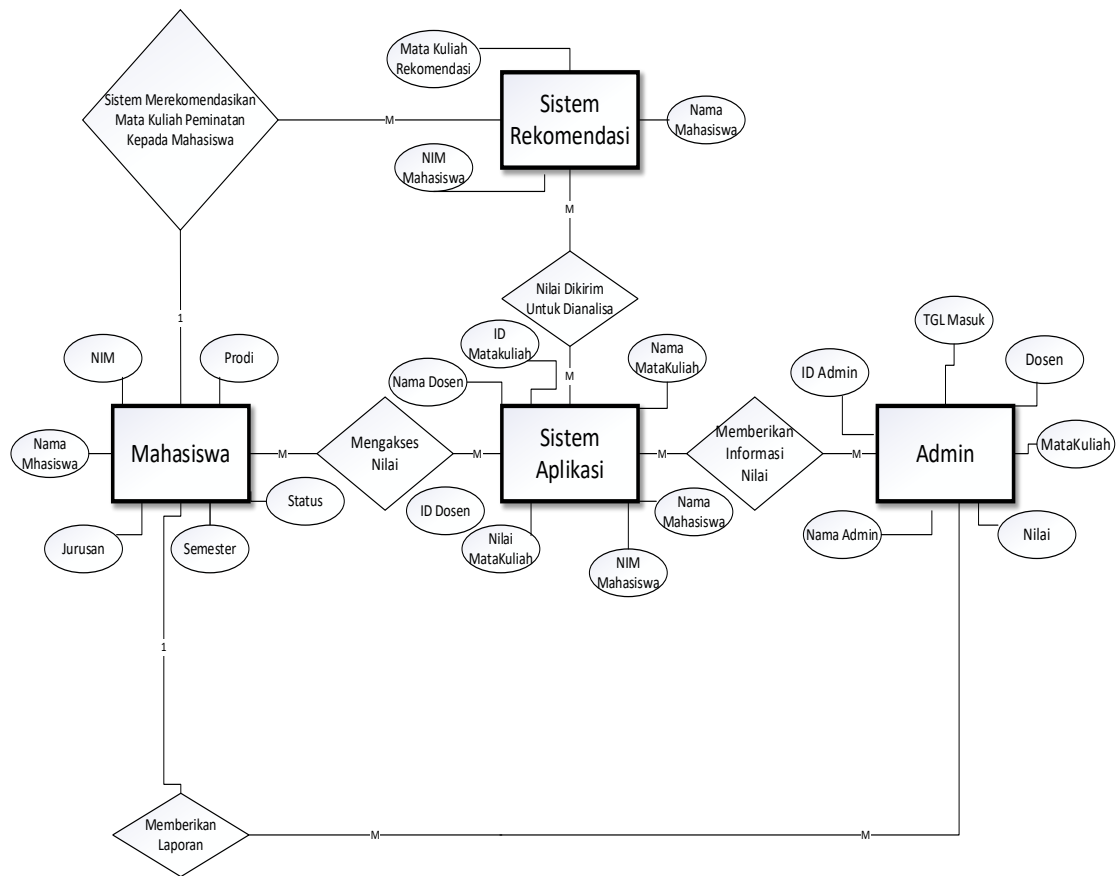
Aplikasi pengukuran capaian belajar dan Rekomendasi mata kuliah peminatan ini memiliki tiga buah *use case*. Yaitu *use case* admin, mahasiswa dan dosen. *Use case* tersebut dapat dilihat pada gambar berikut:



Gambar 2 Use case Diagram

3.4 Entity Relationship Diagram

Untuk perancangan aplikasi ini, diperlukan suatu *Entity Relationship Diagram* agar dapat memberikan gambaran hubungan antar entitas yang terjadi pada rancangan ini. Berikut adalah gambar ERD yang dibuat:



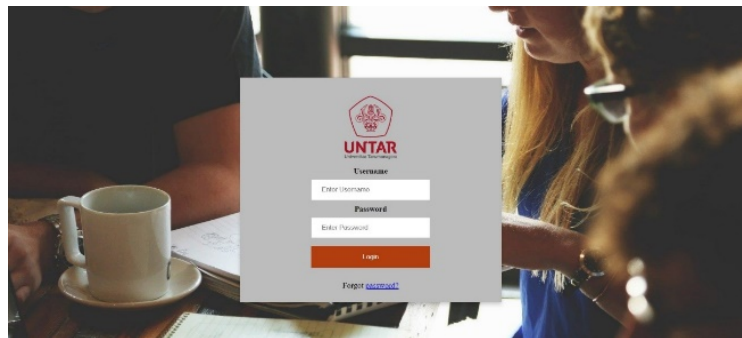
Gambar 3 ERD

3.5 Alur Penggunaan

Screen desain merupakan salah satu komponen penting yang mesti dibuat pada saat merancang sebuah program. Setiap *screen* mewakili beberapa modul atau fungsi yang dirancang. Melalui modul dan *screen* tersebut juga dijelaskan alur penggunaannya. Berikut modul, *screen* desain dan tata cara penggunaan aplikasi yang sudah dibuat:

1. Modul *Login*

Modul *login* adalah modul dasar yang umumnya digunakan pada setiap aplikasi agar data yang diproses dapat disesuaikan dengan setiap pengguna yang mengakses aplikasi ini. Modul ini juga berfungsi untuk menjaga keamanan dalam mengakses data pada aplikasi ini. Pada modul ini *user* diminta memasukkan *Username* dan *Password* untuk mengakses data nilai yang diinginkan. Berikut adalah UI untuk *login*:



Gambar 4 Login

2. Modul Home

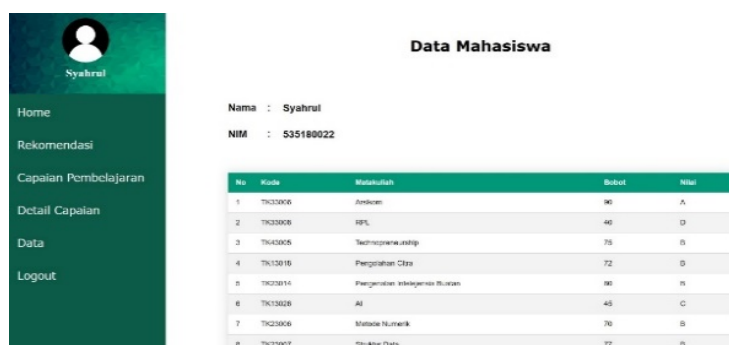
Modul Home ini adalah modul yang berisikan tampilan awal dari aplikasi setelah user login. Modul ini berisikan dua menu yang dapat dipilih oleh user. Menu pertama adalah daftar nilai, capaian belajar dan rekomendasi mata kuliah peminatan yang ingin diambil. Berikut adalah UI untuk home:



Gambar 5 Home

3. Modul Daftar Nilai (User)

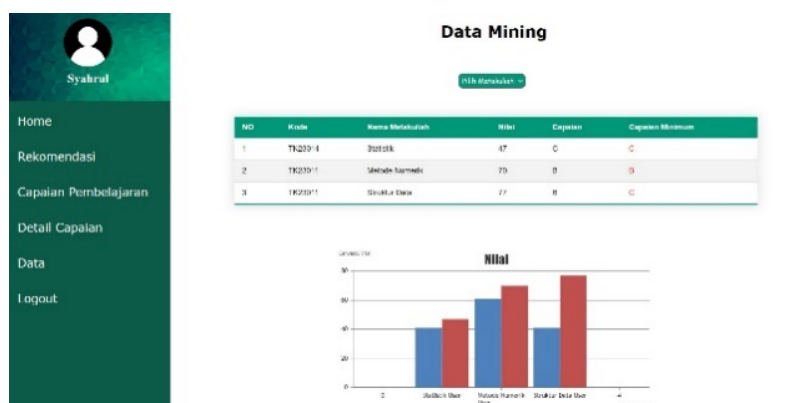
Untuk user modul nilai ini berisikan daftar nilai tanpa dapat disunting atau diubah. Berikut adalah UI untuk daftar nilai User:



Gambar 6 Data Mahasiswa

4. Modul Rekomendasi dan Capaian

Pada modul ini user dapat melihat rekomendasi mata kuliah pilihan yang dapat diambil dengan kemungkinan lulus terbesar, berdasarkan dari analisa nilai pada mata kuliah tertentu yang sudah dilakukan oleh aplikasi. Berikut adalah gambar menu rekomendasi dan capaian:



Gambar 7 Hasil Rekomendasi

Syahrul Ramadhan: Pengembangan Aplikasi Pengukuran Capaian Pembelajaran Mahasiswa Untuk Rekomendasi Mata Kuliah Peminatan Menggunakan Algoritma Decision Tree



Gambar 8 Hasil Capaian Mahasiswa

3.6 Pembuatan Sistem

Pada proses selanjutnya adalah proses pembuatan aplikasi. Aplikasi ini merupakan aplikasi berbasis *web*. Aplikasi yang digunakan untuk proses pengkodean adalah Visual studio. Untuk Bahasa pemrogramannya sendiri, digunakan JavaScript untuk membuat *frontend* maupun *backend*. Untuk sisi *server* nya sendiri disini menggunakan Node JS untuk mengeksekusi proses data yang berjalan. Kemudian untuk *software* sistem basis datanya menggunakan mongoDB.

Untuk metode pengembangan yang digunakan adalah SDLC *Waterfall*. Teknik model pengembangan ini dilakukan karena dinilai memiliki cara kerja yang efisien. Karena dapat meminimalisir kemungkinan yang tidak diinginkan dan proses pengembangan yang cukup cepat. Selain itu metode ini juga dapat meningkatkan hubungan dengan *client*, yang dalam hal ini mahasiswa. Dikarenakan mahasiswa ikut andil dalam menilai kinerja aplikasi ini.

Proses pertama yang dilakukan pada pembuatan sistem ini adalah desain *mockup*. Desain ini dibuat dengan tujuan untuk mempermudah penulis dalam membuat antarmuka. Setelah *mockup* dibuat selanjutnya penulis menimplementasikan desain tersebut kedalam aplikasi pengkodean yang digunakan. Setelah *frontend* selesai dibuat, proses selanjutnya yang dilakukan adalah pembuatan *backend* aplikasi. Proses ini pada saat ini masih dalam tahap pengerjaan.

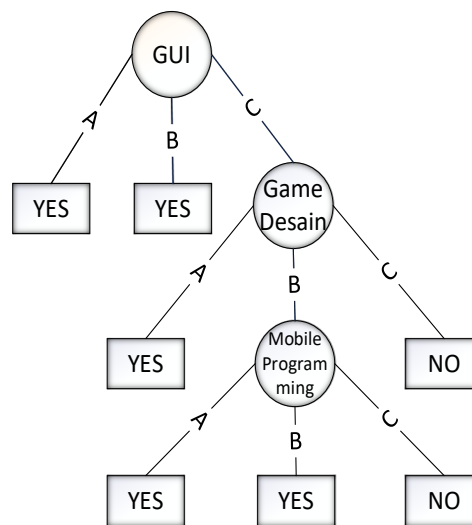
Proses selanjutnya adalah pembuatan NoSQL *database* menggunakan mongoDB. Setelah proses ini selesai selanjutnya dilakukan proses pengujian aplikasi dengan mengolah data yang sudah berhasil dikumpulkan. Setelah proses pengujian selesai, dilanjutkan dengan proses evaluasi untuk dilakukan pengembangan agar aplikasi dapat berjalan dengan sesuai dan efisien.

3.7 Pengujian

Pengujian dilakukan oleh *user* yang akan menggunakan aplikasi ini, agar dapat berjalan sesuai dengan kebutuhan yang diinginkan. Pengujian dimulai dengan melakukan *internal testing* yang dilakukan langsung oleh mahasiswa aktif Fakultas Teknologi informasi Universitas Tarumanagara. *User testing* ini dilakukan dengan membuat aplikasi kedalam beberapa modul untuk dijalankan. Modul tersebut dilakukan pengetesan sesuai dengan skenario yang sudah disiapkan.

3.8 Hasil Pohon Keputusan

Tahap pertama pada algoritma ini adalah mencari nilai *entropy* seluruh atribut penilaian. Tahap selanjutnya nilai *entropy* tersebut digunakan untuk menghitung nilai *Gain* pada setiap matakuliah peminatan. Berikut adalah hasil pohon keputusan pada perhitungan yang dilakukan:



Gambar 9 Hasil pohon keputusan

3.9 Pengujian Metode

Pengujian metode rekomendasi matakuliah dilakukan untuk metode yang tertera pada perancangan aplikasi ini yaitu *confusion matrix* [6]. Untuk rumus perhitungan dapat dilihat sebagai berikut:

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

Keterangan:

TP = Jumlah data *true* positif

TN = Jumlah data *true* negatif

P = Jumlah data Positif yang sebenarnya

N = Jumlah data negatif yang sebenarnya

Untuk metode *confusion matrix* perhitungan dilakukan dengan mengambil 20 persen data untuk diuji. Kemudian data tersebut diklasifikasi menjadi data uji *true*, data uji *false*, data hasil *true* dan data hasil *false*. Berikut adalah tabel data yang sudah diklasifikasi:

Tabel 1 Klasifikasi Data Uji 20%

	Hasil <i>True</i>	Hasil <i>False</i>
Uji Lulus	6	2
Uji Tidak	1	3

$$Accuracy = \frac{6+3}{8+4} = \frac{9}{12} \times 100\% = 75\%$$

Error yang terjadi masih cukup besar, oleh sebab itu perancang menambah data pada rancangan agar tingkat akurasi yang dihasilkan sesuai. Setelah data ditambah selanjutnya tingkat akurasi dihitung Kembali dengan 40% data pada rancangan berikut adalah hasil akurasinya:

Tabel 2 Klasifikasi Data Uji 40%

	Hasil <i>True</i>	Hasil <i>False</i>
Uji Lulus	25	2
Uji Tidak	2	3

$$Accuracy = \frac{25+3}{27+5} = \frac{28}{32} \times 100\% = 87,5\%$$

3.10 Pembahasan

Dari hasil pengujian yang dilakukan aplikasi rekomendasi matakuliah dan pengukuran capaian pembelajaran terdapat beberapa fitur yang ditambahkan seperti diagram pada hasil capaian agar mudah dipahami. Selama proses pengujian aplikasi tidak ada penemuan bug atau pemrosesan data yang salah. Untuk pengujian yang dilakukan pada aplikasi menunjukkan tidak terdapat *bug* dalam aplikasi Pengukuran Capaian Pembelajaran dan Rekomendasi Matakuliah dan dapat berjalan dengan lancar. Hasil akurasi menunjukkan tingkat *error* yang lebih sedikit setelah menambah data.

4. KESIMPULAN

Pembuatan aplikasi pengukuran capaian pembelajaran dan rekomendasi matakuliah ini bertujuan agar mempermudah mahasiswa dan juga pihak kampus dalam memberikan dan menerima informasi capaian serta rekomendasi secara otomatis. Aplikasi ini membantu pengguna untuk bekerja lebih efektif. Secara garis besar pembuatan aplikasi pengukuran capaian pembelajaran dan rekomendasi matakuliah yang telah dilakukan dapat disimpulkan sebagai berikut:

1. Perlu dilakukannya migrasi dari cara manual dalam memberikan rekomendasi matakuliah dan pengukuran capaian pembelajaran menjadi otomatis agar mempermudah pihak kampus.
2. Dengan menggunakan aplikasi ini, maka proses pengolahan data yang dibutuhkan kampus akan lebih mudah.
3. Rekomendasi yang dilakukan pada aplikasi ini diharapkan dapat memberikan saran yang sesuai dengan apa yang *user* butuhkan berdasarkan hasil analisa pada nilai matakuliah *user* yang bersangkutan.
4. Hasil akurasi menunjukkan tingkat *error* yang lebih sedikit setelah menambah data. Dari 20% data menghasilkan akurasi sebesar 75%. Setelah ditambah menjadi 40% data menghasilkan tingkat akurasi 87,5%. Untuk mendapatkan hasil akurasi yang maksimal, maka diperlukan data yang lebih banyak untuk pengujiannya.

DAFTAR PUSTAKA

- [1] Rahayu, Maria, Nugraheni, Mardi., dan Arianti, Rudangta., 2020, Penyesuaian Mahasiswa Tahun Pertama di Perguruan Tinggi: Studi Pada Mahasiswa Fakultas Psikologi UKSW, *Jurnal Psikologi Sains dan Profesi*, No. 2, Vol. 4.
- [2] Puspita, Rani., dan Widodo, Agus., 2020, Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS, *Jurnal Informatika Universitas Pamulang*, No. 4, Vol. 5.
- [3] Brown, Ethan., 2016, *Learning JavaScript*, Ed. 3, O'Reilly Media, Sabastopol.
- [4] Rokach, Lior., dan Maimon, Oded., 2015, *Data Mining with Decision Trees: Theory and Applications*, Ed. 2, World Scientific, Jurong East.
- [5] Arifin, Muhamad, Fauzal., dan Fitriana, Devi., 2018, Penerapan Algoritma Klasifikasi C4.5 dalam Rekomendasi Penerimaan Mitra Penjualan Studi Kasus: PT Atria Artha Persada, *Jurnal Telekomunikasi dan Komputer*, No. 2, Vol. 8.
- [6] Barito, Eugenius, Edsel., dan Beng, Jap, Tji., dan Arisandi, Desi., 2022, Penerapan Algoritma C4.5 Untuk Klasifikasi Mahasiswa Penerima Bantuan Social Covid-19, *Jurnal Ilmu Komputer dan Sistem Informasi*, No. 1, Vol. 9.