

## **BAB 3**

### **METODOLOGI PENELITIAN**

Untuk membangun aplikasi Sistem Rekomendasi pilihan mata kuliah elektif, digunakan metode SDLC (*Software Development Life Cycle*). *Software Development Life Cycle* adalah sebuah kerangka kerja yang digunakan untuk menyusun, merencanakan, dan mengendalikan proses pengembangan sistem informasi [19]. Model kerangka kerja yang digunakan adalah *Waterfall*. Model *Waterfall* dilakukan dengan pendekatan yang sistematis, dimulai dari tahap analisis, desain, implementasi, *testing*, dan evaluasi [20].

#### **3.1 Analisis Kebutuhan**

Tahap analisis kebutuhan merupakan tahap awal dalam metode *waterfall*. Pada tahap analisis kebutuhan, hal yang dilakukan berhubungan dengan studi literatur untuk mencari teori-teori yang mendasari pembangunan dan pengembangan sistem. Teori yang dicari berasal dari jurnal ilmiah sebelumnya dan pendapat para ahli. Selain mencari teori-teori yang mendasari, dilakukan juga wawancara yang digunakan untuk mengumpulkan informasi yang dibutuhkan dari pakar. Setelah melakukan studi literatur, dilanjutkan dengan melakukan analisis requirements yang dibutuhkan secara spesifikasi *hardware* dan *software* yang akan digunakan dalam pembangunan sistem.

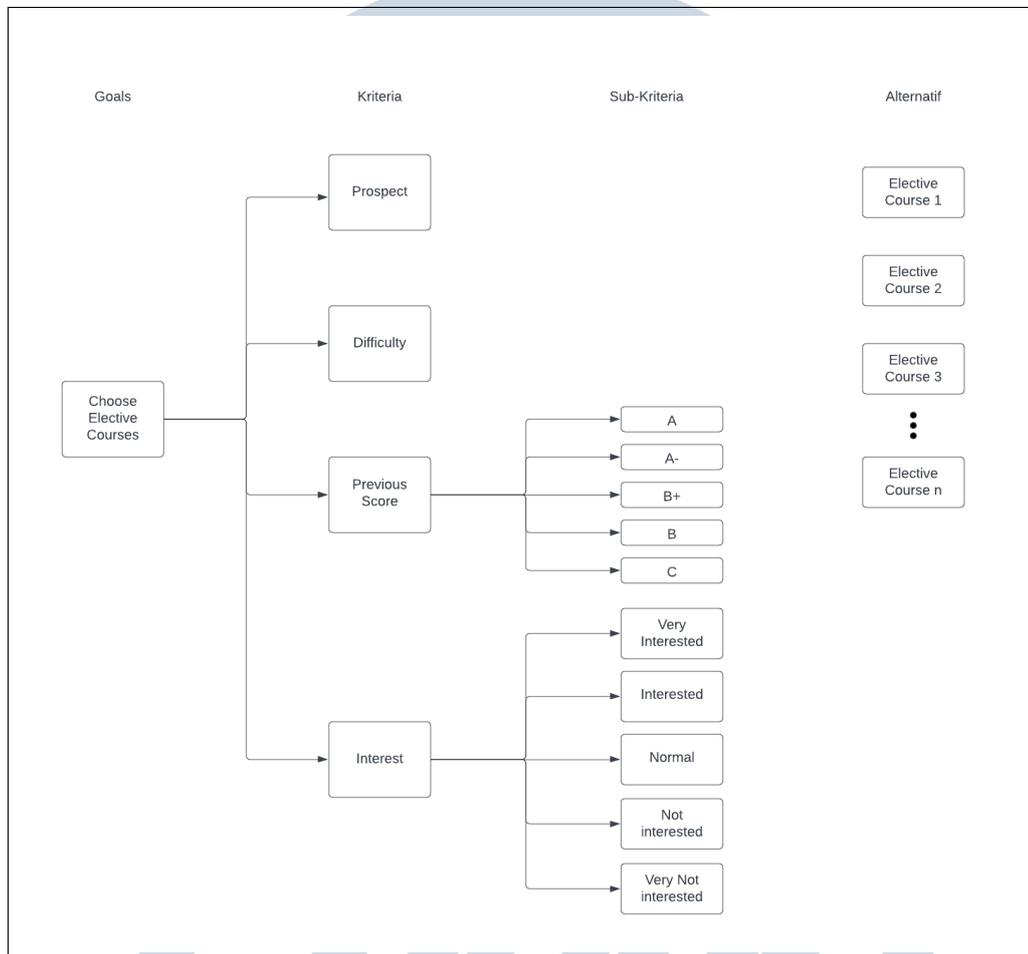
#### **3.2 Perancangan Sistem**

Perancangan sistem merupakan tahap kedua, dengan tujuan untuk menggambarkan struktur sistem dan menjelaskannya secara keseluruhan bagaimana sistem akan dibangun dan beroperasi berdasarkan kebutuhan yang telah diidentifikasi. Perancangan sistem yang dibuat berisikan penjelasan mengenai *data flow diagram*, *flowchart*, dan struktur *database* dari sistem yang dibangun.

##### **3.2.1 Hierarki AHP**

Telah dibuat tabel hierarki AHP yang digunakan dalam sistem yang dibuat. Tabel Hierarki AHP terdiri dari empat bagian yaitu: *Goals*, Kriteria, Sub-Kriteria,

dan Alternatif dari sistem yang dibuat. Berikut akan dilampirkan gambar tabel hierarki yang telah dibuat.



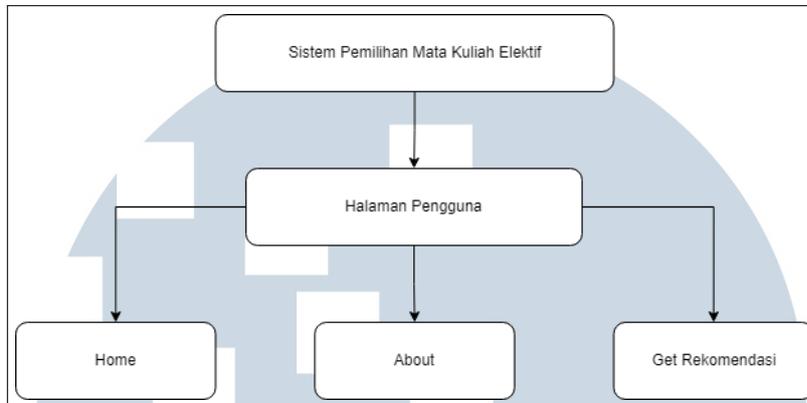
Gambar 3.1. Hierarki AHP Mata Kuliah Elektif

Pada gambar 3.1 dapat dilihat bahwa tujuan dari AHP adalah Pemilihan Mata Kuliah Elektif, dimana kriteria dari tujuan tersebut ada empat yaitu Prospek, Tingkat Kesulitan, Nilai Mata Kuliah Sebelum, dan Minat. Pada kriteria Nilai Mata Kuliah Sebelum, terdapat sub-kriteria yaitu A, A-, B+, B, dan C. Sedangkan pada kriteria Minat terdapat sub-kriteria Sangat tertarik, Tertarik, Normal, Tidak Tertarik, Sangat Tidak Tertarik. Pada bagian kanan gambar 3.1 terdapat alternatif yang merupakan hasil dari rekomendasi Pemilihan Mata Kuliah Elektif.

### 3.2.2 Sitemap

Sitemap adalah gambaran dari navigasi sistem yang akan dibuat. Sitemap akan menggambarkan navigasi yang terdapat pada sistem yang dibuat. Berikut

adalah gambaran dari sitemap yang telah dibuat.

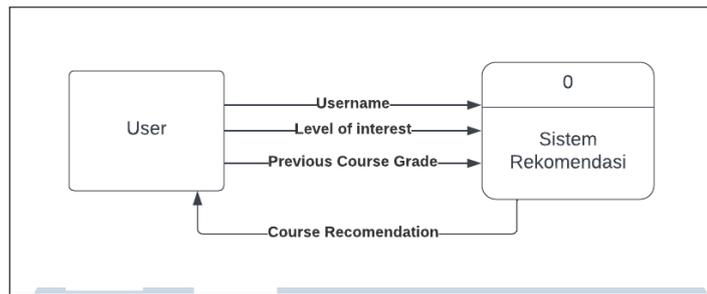


Gambar 3.2. Sitemap

Seperti yang dapat dilihat pada gambar diatas, Sitemap berisikan navigasi halaman-halaman yang tersedia pada sistem. Halaman yang terdapat dalam sistem yang dirancang ada 4 yaitu: Home, About, Information, dan Get Rekomendasi. Halaman Home merupakan halaman awal yang akan ditampilkan kepada pengguna ketika mengakses sistem yang telah dirancang. Pada halaman home terdapat *button* yang dapat digunakan oleh pengguna untuk mengarahkan pengguna ke halaman lain yang terdapat dalam sistem yakni halaman About, Details, dan halaman Get Rekomendasi. Apabila pengguna menekan tombol halaman About, maka pengguna akan segera diarahkan ke halaman About yang berisikan biodata mengenai perancang sistem. Pada halaman tersebut, pengguna juga dapat berpindah ke halaman Information, Get Rekomendasi. Apabila pengguna memilih untuk menekan *button* yang mengarahkan ke halaman *Information*, maka pengguna akan diarahkan ke halaman tersebut. Halaman *Information* halaman tersebut berisikan informasi mengenai rekomendasi mata kuliah peminatan yang ada dalam sistem. Dalam halaman tersebut akan dicantumkan deskripsi mengenai mata kuliah yang ada. Selanjutnya terdapat halaman terakhir yaitu halaman Get Rekomendasi. Halaman tersebut akan mengarahkan pengguna ke dalam tahap untuk mendapatkan rekomendasi yang sesuai dengan kebutuhan pengguna.

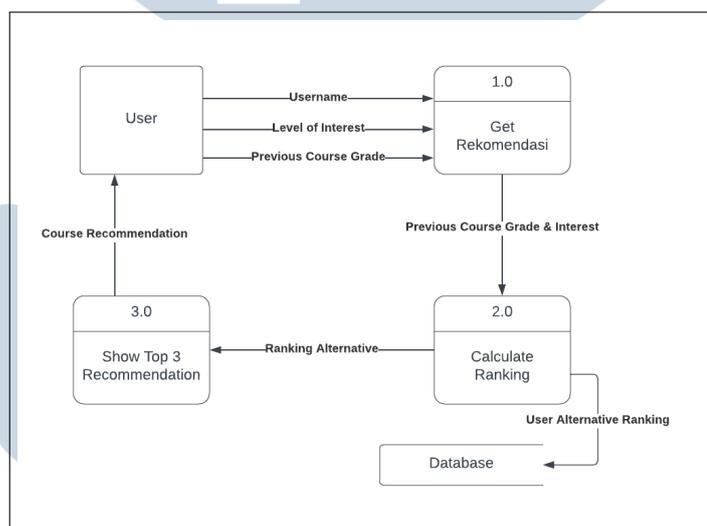
### 3.2.3 Data Flow Diagram

Pada bagian ini akan dijelaskan *data flow diagram* (DFD) yang dibuat untuk menjelaskan alur proses dari informasi yang dimiliki oleh sistem. Terdapat dua data yaitu *input*, dan *output*. Terdapat dua level *data flow diagram* (DFD) yang dibangun yaitu diagram konteks, dan *data flow diagram* level 1.



Gambar 3.3. Data Flow Diagram Level 0 sistem rekomendasi *my-course-rec*

Gambar 3.3 merupakan gambar diagram konteks yang dimiliki oleh sistem rekomendasi *my-course-rec*. Terdapat satu aktor yaitu *aktor* yang menerima dan memberikan informasi yang dibutuhkan dan dihasilkan oleh sistem. DFD level 0 merupakan gambaran besar dari sistem rekomendasi *my-course-rec* yang dibuangun. Dapat dilihat alur informasi yang terjadi pada entitas *user* terdapat tiga informasi yang diberikan kepada sistem untuk mendapatkan rekomendasi. informasi yang diberikan adalah *username*, *level of interest*, dan *previous course grade*.



Gambar 3.4. Data Flow Diagram Level 1 sistem rekomendasi *my-course-rec*

Gambar 3.4 merupakan *data flow diagram level 1*. Digambarkan tiga proses yang menjelaskan secara lebih detail alur informasi terhadap proses yang dimiliki oleh sistem rekomendasi yang dibangun. Tiga proses tersebut adalah *Get Rekomendasi*, *Calculate Ranking*, *Show Top 3 Recommendation*. Proses *Get Rekomendasi* adalah proses pengguna mengisi pertanyaan-pertanyaan yang telah disediakan untuk mendapatkan rekomendasi yang diinginkan. Proses *Calculate Ranking* adalah proses dimana bobot dari masing-masing alternatif akan dihitung

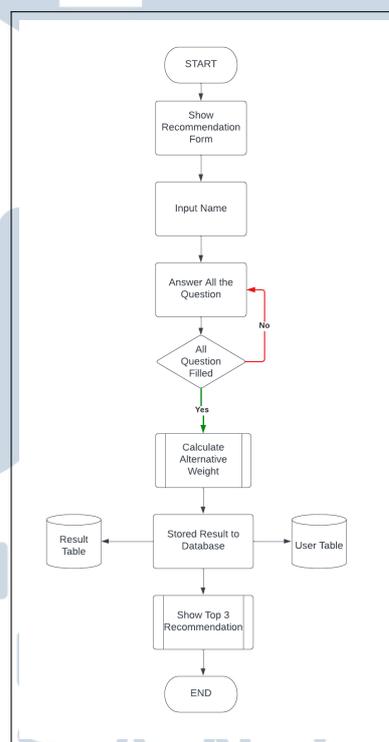
sesuai dengan jawaban yang telah diberikan oleh pengguna sebelumnya. Setelah menghitung, maka dilakukan perankingan sesuai dengan bobot tertinggi. Setelah dilakukan perankingan sesuai dengan bobot tertinggi, maka tiga rekomendasi dengan bobot tertinggi akan ditampilkan kepada pengguna.

### 3.2.4 Diagram Alir

Diagram alir atau *flowchart* yang dibuat berfungsi untuk menggambarkan alur kerja dari sistem yang akan dibuat. Terdapat 3 *flowchart* dalam program yang akan dibangun. Berikut akan dilampirkan *flowchart* yang telah dibuat untuk alur pengguna ketika menjalankan program.

#### 1. Flowchart Pengguna Get Rekomendasi

Berikut akan dilampirkan gambar serta penjelasan *flowchart* dari proses pengguna mendapatkan rekomendasi dari sistem rekomendasi yang telah dibangun.

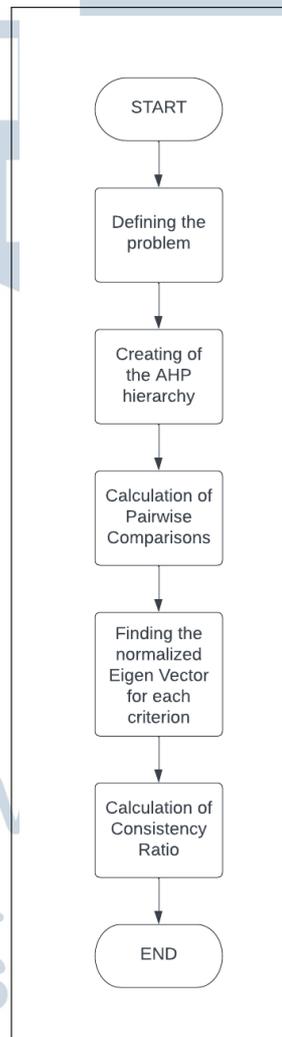


Gambar 3.5. Flow Get Rekomendasi

Seperti yang dapat dilihat pada gambar 3.5 ketika user memasuki halaman Get Rekomendasi, program akan menampilkan sebuah *form* yang harus diisi

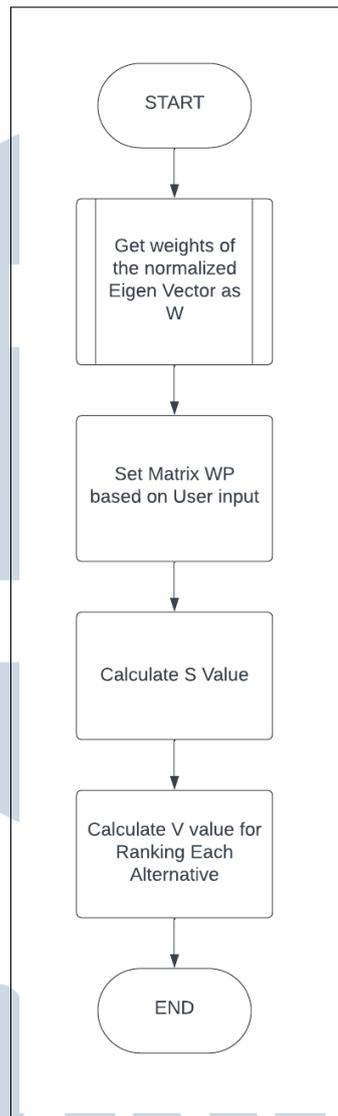
oleh pengguna. Pada *form* yang disediakan, pengguna akan diarahkan untuk memasukkan nama terlebih dahulu. Setelah pengguna selesai memasukkan nama, terdapat pertanyaan-pertanyaan yang harus diisi oleh pengguna untuk mendapatkan rekomendasi. Apabila pengguna belum menjawab seluruh pertanyaan, maka pengguna tidak dapat lanjut ke tahap selanjutnya sampai seluruh pertanyaan telah diisi. Apabila pengguna telah selesai menjawab seluruh pertanyaan, bobot alternatif akan dikalkulasikan dan hasil akan disimpan kedalam *database*. Selanjutnya, tiga rekomendasi teratas akan ditampilkan kepada pengguna.

#### 1. Flowchart Kalkulasi nilai Eigen Vektor Normalisasi



Gambar 3.6. Flow Kalkulasi nilai Eigen Vektor Normalisasi

#### 2. Flowchart Kalkulasi Bobot Alternatif

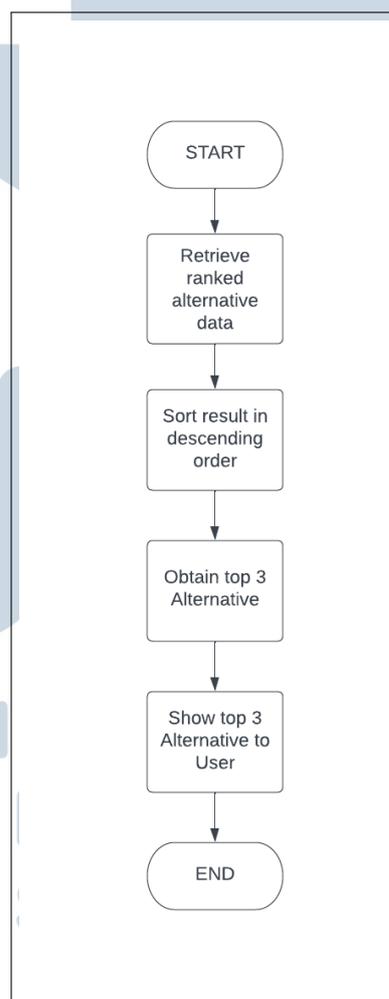


Gambar 3.7. Flow Kalkulasi Bobot Alternatif

Seperti yang telah disinggung sebelumnya, ketika pengguna sedang melakukan proses Get Rekomendasi terdapat sebuah tahap yang dilalui yaitu tahap menghitung bobot alternatif. Proses yang terjadi pada tahap menghitung bobot alternatif dapat dilihat pada gambar 3.7. Proses pertama yang dilakukan adalah mengambil bobot Eigen Vektor Normalisasi dari setiap kriteria yang didapatkan dari perhitungan yang dilakukan sebelumnya oleh metode AHP. Perhitungan yang dilakukan oleh AHP untuk mendapatkan bobot nilai Eigen Vektor Normalisasi dapat dilihat pada gambar 3.6. Seperti yang dapat dilihat pada gambar 3.6 tahap yang dilakukan adalah mendefinisikan masalah yang ingin diselesaikan. Setelah selesai mendefinisikannya, selanjutnya adalah melakukan hierarki

AHP yang digunakan untuk menentukan *goals, criteria, sub-criteria* dan alternatif. Setelah membuat hierarki AHP, selanjutnya adalah perhitungan *pairwise comparison* antar kriteria yang digunakan. Setelah mendapatkan bobot dari *pairwise comparison*, bobot tersebut akan digunakan untuk mengkalkulasikan nilai Eigen Vektor Normalisasi. Setelah mendapatkan bobot Eigen Vektor Normalisasi, selanjutnya mengatur matrix WP berdasarkan *input* dari pengguna. Setelah melakukan pengaturan matriks, selanjutnya melakukan kalkulasi nilai S yang kemudian digunakan untuk menghitung ranking dari masing-masing alternatif yang ada. Setelah menyelesaikan perhitungan bobot alternatif, hasil akan disimpan ke dalam database yang kemudian hasil alternatif akan ditampilkan kepada pengguna.

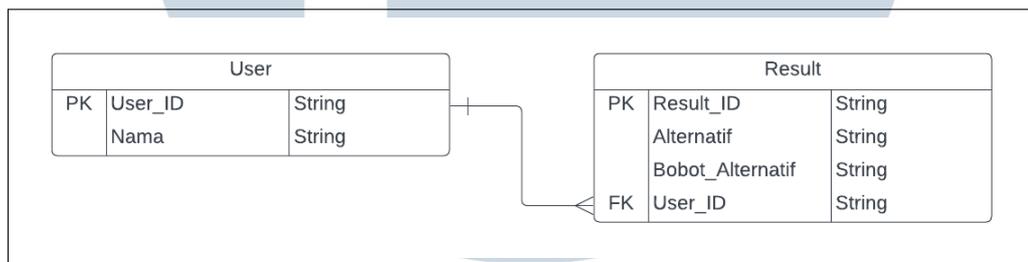
### 3. Flowchart Result



Gambar 3.8. Flow Tampilkan Result

Setelah selesai melakukan perhitungan bobot alternatif, tahap terakhir yang akan dilakukan dalam program adalah menampilkan hasil rekomendasi kepada pengguna. Seperti yang dapat dilihat pada gambar 3.8 proses yang dilakukan ketika hasil alternatif akan ditampilkan kepada pengguna. Proses pertama yang dilakukan adalah mengambil data alternatif peringkat yang telah dihitung sebelumnya. Setelah data tersebut diambil akan dilakukan sorting terhadap hasil secara urutan *descending*. Setelah melakukan *sorting*, program akan mengambil tiga alternatif terbaik. Ketiga alternatif yang diambil akan ditampilkan kepada pengguna sebagai rekomendasi.

### 3.2.5 Struktur Basis Data (ERD)



Gambar 3.9. Struktur basis data sistem rekomendasi pemilihan mata kuliah elektif

Gambar 3.9 merupakan struktur dan relasi tabel yang dimiliki oleh sistem rekomendasi pemilihan mata kuliah elektif. Terdapat 2 tabel yang saling berkaitan untuk menyimpan data-data yang dimiliki oleh sistem. Tabel tersebut antara lain, User tabel, dan Result tabel yang merupakan tabel penghubung dari relasi *one to many*. Berikut adalah penjelasan dari masing-masing tabel.

#### A Tabel User

Tabel User merupakan tabel pengguna yang menyimpan data-data dari pengguna yang menggunakan sistem rekomendasi yang dibangun. Berikut akan dilampirkan tabel user yang telah dibuat beserta dengan penjelasannya.

Tabel 3.1. User

Key	Nama Kolom	Tipe Data	Keterangan
PK	user_id	String (25)	Primary key untuk tabel <i>user</i>

	nama	String (50)	Nama pengguna
--	------	-------------	---------------

Tabel 3.1 merupakan tabel user yang telah dibuat. Tabel User menyimpan dua data dari pengguna yaitu USER\_ID dan nama dari pengguna, kedua data yang terdapat pada tabel user memiliki tipe data string. Pada tabel user, terdapat sebuah *primary key* yaitu USER\_ID. Tabel user memiliki relasi *one to many* dengan tabel *result*.

## B Tabel Result

Tabel Result merupakan tabel yang menyimpan data alternatif pada sistem yang akan ditampilkan kepada pengguna. Berikut akan dilampirkan tabel user yang telah dibuat beserta dengan penjelasannya.

Tabel 3.2. Result

Key	Nama Kolom	Tipe Data	Keterangan
PK	result_id	String (25)	<i>Primary key</i> untuk tabel <i>hr_admin_group</i>
	Alternatif	String (15)	Nama alternatif rekomendasi
	Bobot_Alternatif	String (20)	Bobot alternatif rekomendasi
FK	user_id	String(25)	ID pengguna dari tabel user

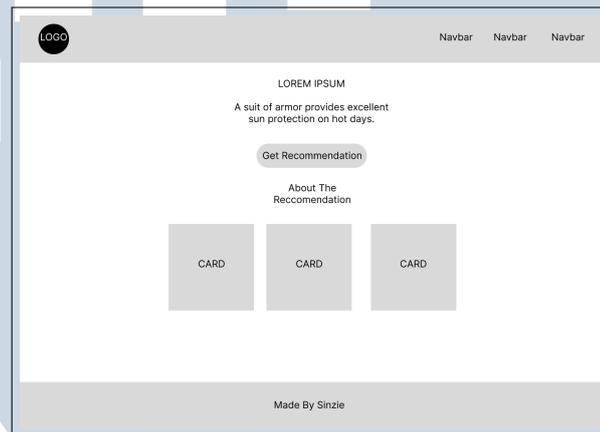
Pada tabel 3.2 *result* terdapat sebuah *primary key* yaitu Result\_ID. Tabel *result* juga memiliki sebuah *foreign key* yaitu User\_ID. data lainnya yang disimpan pada tabel *result* adalah data alternatif, dan juga bobot dari masing-masing alternatif yang ada.

## 3.3 Design

Tahap desain, akan dilakukan perancangan *wireframe* yang akan dijadikan sebagai acuan tampilan dari sistem yang akan dibuat. Terdapat 6 halaman *wireframe* yang dibuat yakni, halaman *home page*, *about page*, *information page*, *details page*, *get recommendation page*, dan *result page*. Berikut akan dilampirkan, dan

dijabarkan penjelasan mengenai *wireframe* untuk tampilan pengguna yang telah dibuat.

Tampilan pertama yang akan ditampilkan kepada pengguna ketika mengakses website yang dibuat adalah halaman *homepage*. Berikut adalah tampilan halaman homepage yang akan dibuat.

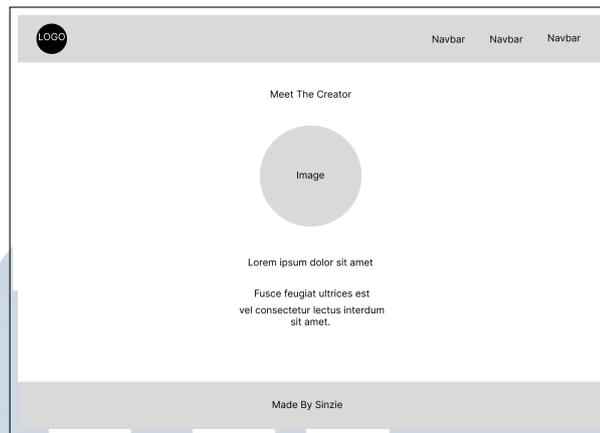


Gambar 3.10. Tampilan Homepage

Dapat dilihat pada gambar 3.10 terdapat logo aplikasi pada bagian atas kiri, dan button navigasi pada atas kanan yang dapat digunakan oleh pengguna untuk berpindah-pindah halaman. Pada bagian konten halaman, terdapat tulisan, dan disertai *button* yang bertulisan get rekomendasi. Pengguna dapat menekan tombol tersebut dan akan diarahkan ke halaman pengambilan rekomendasi. Pada bagian akhir, terdapat *cards* yang berisi gambar dari rekomendasi yang ada. Pengguna dapat menekan gambar tersebut untuk diarahkan ke halaman *information* untuk melihat informasi mengenai rekomendasi yang ada. Terakhir, terdapat *footer* pada bagian bawah halaman.

Apabila pengguna menekan tombol navigasi *About*, maka pengguna akan diarahkan ke halaman *About*. Berikut akan dilampirkan tampilan halaman *About*.

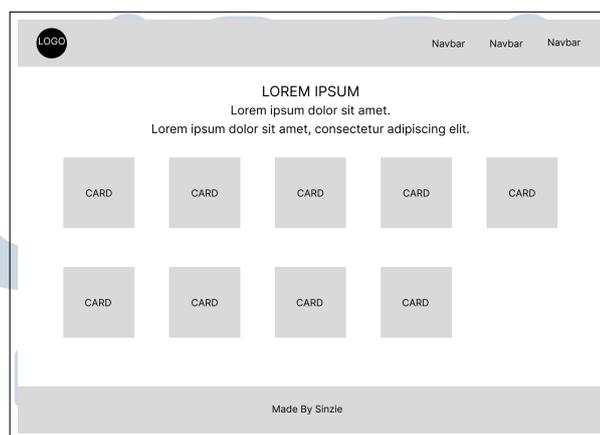
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.11. Halaman About

Seperti yang dapat dilihat pada gambar 3.11 pada halaman *about* terdapat *header*, dan pada konten dari halaman tersebut, pengguna akan melihat biodata dari perancang website. Pada halaman *about* akan ada gambar dari perancang, nama, jurusan, dan angkatan dari perancang website. Pada bagian bawah halaman terdapat *footer*.

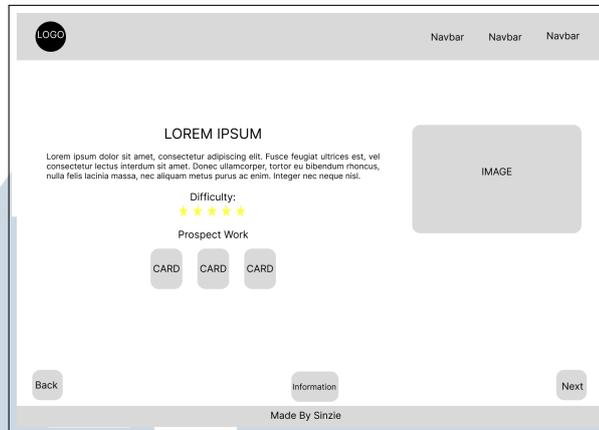
Selanjutnya terdapat halaman *Information* yang dapat diakses oleh pengguna. Halaman *Information* berisikan informasi-informasi yang dapat dilihat dan dibaca oleh pengguna mengenai data rekomendasi yang ada pada website. Berikut adalah tampilan dari halaman *Information*.



Gambar 3.12. Halaman Information

Gambar 3.12 merupakan tampilan awal yang akan dilihat oleh pengguna ketika mengakses halaman *Information*. Pengguna akan melihat *cards* yang berisikan gambar dari masing-masing rekomendasi. Pengguna dapat menekan *cards* yang tersedia. Apabila pengguna menekannya, maka pengguna akan diarahkan ke halaman *Information Details*. Berikut adalah tampilan dari halaman

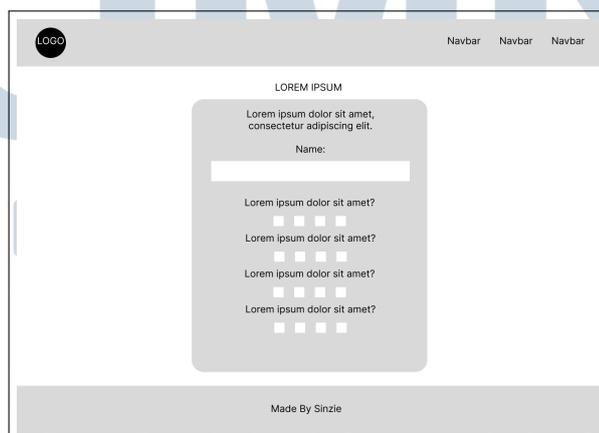
### Information Details.



Gambar 3.13. Halaman Information Details

Dapat dilihat pada gambar 3.13 pada halaman *Information Details* bagian konten terdapat nama dari mata kuliah peminatan yang ada, dan deskripsi mengenai mata kuliah peminatan tersebut hingga tingkat kesulitannya. Selanjutnya, terdapat *cards* yang akan menggambarkan prospek kerja dari peminatan mata kuliah tersebut. Pada bagian bawah terdapat *button* yang dapat digunakan oleh pengguna untuk berpindah halaman menuju halaman *Information Details* lainnya yang telah disediakan, dan *button* yang dapat mengarahkan pengguna kembali ke halaman *information*.

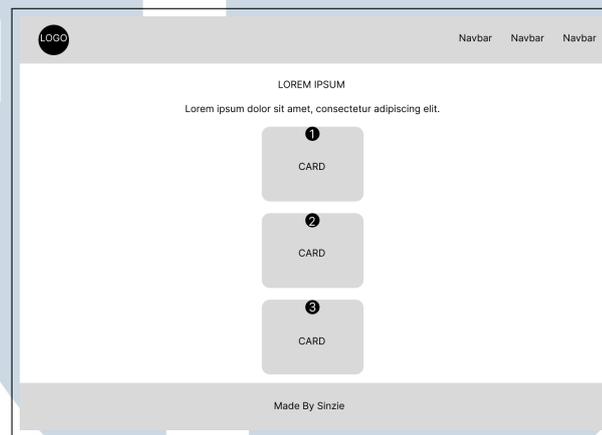
Selanjutnya merupakan halaman terakhir yang dapat diakses oleh pengguna, yaitu halaman *Get Rekomendasi*. Berikut adalah tampilan gambar dari halaman *Get Rekomendasi*.



Gambar 3.14. Halaman Get Rekomendasi

Dapat dilihat pada gambar 3.14 halaman *Get Rekomendasi* akan ada sebuah *form* yang harus diisi oleh pengguna. *form* terdiri dari pertanyaan nama, hingga

pilihan ganda yang nantinya akan diisi oleh pengguna. Setelah pengguna selesai mengisi *form* yang telah disediakan, dan yakin akan jawaban tersebut, pengguna dapat menekan tombol *submit*. Setelah pengguna mengumpulkan jawabannya, pengguna akan diarahkan ke halaman hasil rekomendasi yang akan menampilkan hasil rekomendasi mata kuliah peminatan. Berikut adalah gambaran halaman Hasil Rekomendasi.



Gambar 3.15. Halaman Hasil Rekomendasi

Pada halaman Hasil Rekomendasi yang dapat dilihat pada gambar 3.15, terdapat tiga buah *card* yang berisikan gambar dan angka pada bagian atas *card* dari hasil rekomendasi yang diberikan kepada pengguna. Angka pada bagian atas *card* digunakan untuk menandakan ranking dari masing-masing *card* tersebut. Tiga buah *card* akan dijabarkan kebawah sesuai dengan hasil ranking yang telah dilakukan sesuai dengan jawaban pengguna. Pengguna dapat menekan *card* pada halaman *result*. Apabila pengguna menekan salah satu *card*, pengguna akan diarahkan ke halaman *Information Details* untuk melihat informasi lebih lanjut mengenai mata kuliah peminatan tersebut.

### 3.4 Implementasi

Tahap implementasi dilakukan dengan membangun *source code* sistem menggunakan teknologi yang telah ditentukan pada tahap sebelumnya. Struktur Hierarki AHP akan digunakan untuk menghasilkan nilai bobot kriteria. Hasil nilai kemudian akan digunakan oleh metode WP untuk menghasilkan alternatif yang sesuai dan akan menghasilkan mata kuliah elektif yang akan direkomendasikan. Metode akan diimplementasikan menggunakan MongoDB untuk database, Laravel untuk frontend dan backend yang akan dirancang,

### **3.5 Testing**

Tahap pengujian adalah tahap mencari kesalahan pada tiap fungsi yang berfokus pada masukan sistem data, tampilan sistem, pemakaian memori, dan kecepatan eksekusi data sehingga jika masukan data tidak sesuai dengan apa yang diharapkan maka sistem gagal [21]. Pada penelitian ini, pengujian dilakukan dengan memberikan sistem yang telah berhasil dibangun kepada mahasiswa Informatika Universitas Multimedia Nusantara untuk dilakukan pengujian sistem. Sampel mahasiswa yang diambil minimal sebanyak 30 orang. Mahasiswa akan mencoba menggunakan sistem dengan memasukkan nilai akhir mata kuliah wajib yang pernah ditempuh dan bidang mata kuliah elektif yang diminati. Kemudian sistem akan memberikan hasil berupa mata kuliah elektif yang diminati. Kemudian sistem akan memberikan hasil berupa mata kuliah elektif yang direkomendasikan. Sehingga dengan adanya pengujian diharapkan kesalahan yang terdapat dalam sistem dapat diketahui dan dimimalisir.

### **3.6 Maintenance**

Tahap evaluasi dilakukan dengan memberikan kuisisioner kepada mahasiswa Informatika Universitas Multimedia Nusantara yang telah menggunakan sistem. Kuisisioner berisikan pertanyaan mengenai kesesuaian hasil yang diberikan oleh sistem dan kepuasan mahasiswa dengan hasil yang diberikan. Proses penilaian yang diberikan oleh mahasiswa berada dalam rentan skor 5 (sangat sesuai) sampai 1 (sangat tidak sesuai). Hasil dari kuisisioner akan dilihat dan dihitung menggunakan skala likert untuk melihat kesesuaian dan kepuasan mahasiswa dalam hasil yang diberikan oleh sistem.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A