

BAB 3

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Pengembangan sistem dan penyusunan kuesioner dalam penelitian ini dilakukan melalui beberapa tahapan sebagai berikut:

1. Analisis Kebutuhan

Tahap ini bertujuan untuk mengidentifikasi kebutuhan awal sistem, seperti menentukan *framework*, *software*, dan *library* yang diperlukan untuk mengimplementasikan model dan fungsi yang akan digunakan dalam pengembangan *website e-learning* berbasis *Intelligent Tutoring Systems (ITS)*.

2. Studi Literatur

Pada tahap ini dilakukan kajian terhadap teori-teori dan rumus matematika yang relevan, seperti konsep dasar *ITS* dan model *Bayesian Network*, sebagai dasar dalam merancang dan mengembangkan sistem yang adaptif terhadap kemampuan mahasiswa.

3. Perancangan Sistem

Tahap ini mencakup perancangan sistem dalam tiga aspek utama, yaitu:

- Merancang sistem agar sesuai kebutuhan pengguna.
- Merancang antarmuka pengguna (*user interface*) yang menarik dan mudah digunakan.
- Melakukan uji coba awal untuk memastikan setiap fitur berjalan sesuai dengan rancangan.

4. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini berasal dari buku berjudul *Teknik Pemuliaan Tanaman* karya Dr. Muhamad Syukur, SP., M.Si. Materi dari buku tersebut diolah dan diekstraksi menggunakan bantuan platform *AI* seperti *Grok.com* dan *AI Studio* untuk dijadikan sebagai bahan pembelajaran dalam sistem *e-learning*.

5. Implementasi

Tahap implementasi dilakukan dengan merealisasikan hasil perancangan menggunakan bahasa pemrograman *PHP* dengan bantuan *framework Laravel* untuk membangun sistem secara terstruktur dan efisien.

6. Uji Coba dan Analisis Hasil

Website e-learning yang telah dibangun akan diuji melalui perbandingan antara hasil perhitungan sistem dan hasil perhitungan manual. Sistem ini dirancang untuk menghitung distribusi tingkat kesulitan soal pada tugas tambahan yang terdiri dari 20 soal, dan dikelompokkan ke dalam tiga kategori, yaitu: mudah (*easy*), sedang (*medium*), dan sulit (*hard*). Tujuannya adalah untuk memverifikasi keakuratan dan ketepatan hasil perhitungan sistem terhadap analisis manual.

7. Penyebaran Kuesioner

Setelah sistem digunakan oleh mahasiswa untuk mengerjakan empat tugas dalam satu semester pada satu mata kuliah, pengguna akan memperoleh umpan balik secara otomatis dari sistem. Apabila terdapat tugas yang tidak lulus, sistem akan secara otomatis menghasilkan tugas tambahan dengan tingkat kesulitan soal yang disesuaikan berdasarkan performa pengguna sebelumnya. Kuesioner kemudian disebarakan kepada pengguna untuk mengevaluasi kepuasan mereka terhadap sistem berdasarkan model *End-User Computing Satisfaction (EUCS)*.

3.2 Perancangan Website

Dalam perancangan *website*, dilakukan perancangan sistem dalam bentuk program berbasis *web*. Tahap ini bertujuan untuk memastikan sistem yang dikembangkan sesuai dengan tujuan utama penelitian.

Beberapa hal yang dirancang dalam tahap ini meliputi:

- Rancangan *database*, untuk menyimpan data mahasiswa/dosen, soal, nilai, materi dan hasil analisis.
- *Flowchart* program, untuk menggambarkan alur logika sistem dan *user interface*.
- Rancangan *user interface*, untuk menentukan tampilan antarmuka pengguna agar mudah digunakan dan sesuai kebutuhan (*mockup*).

Tahap perancangan ini sangat penting agar proses pengembangan sistem, berikut penjelasan lebih lanjut mengenai rancangan sistem :

3.2.1 Rancangan Flowchart

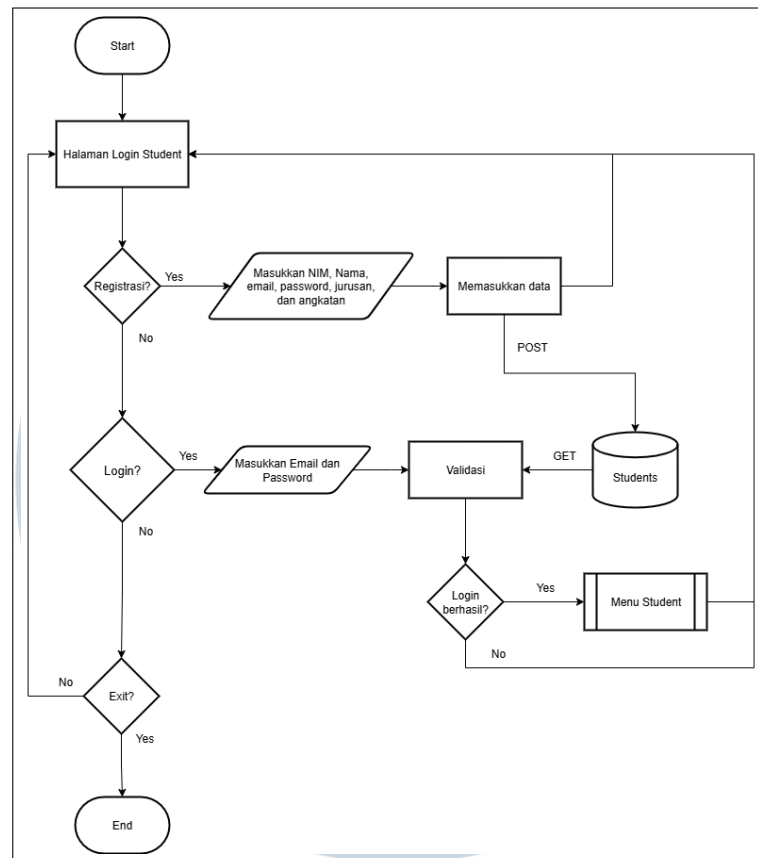
Dalam rancangan sistem ini, terdapat tiga level pengguna (*user level*), yaitu *Student*, *Lecturer*, dan *Admin*.

- ***Student*** merupakan pengguna dari kalangan mahasiswa yang dapat mengakses materi pembelajaran yang telah disediakan oleh dosen, serta mengerjakan tugas-tugas yang diberikan melalui platform e-learning.
- ***Lecturer*** adalah pengguna dari kalangan dosen yang memiliki akses untuk mengunggah materi pembelajaran dan membuat tugas yang akan dikerjakan oleh mahasiswa.
- ***Admin*** berperan sebagai pengelola sistem atau tim *IT*, yang bertanggung jawab untuk membuat akun dosen dan mengatur akses pengguna (baik dosen maupun mahasiswa) terhadap mata kuliah tertentu.

A Flowchart Student

Pada Gambar 3.1, ditampilkan alur awal bagi mahasiswa. Halaman pertama yang diakses adalah *login page*. Jika mahasiswa sudah memiliki akun, maka dapat langsung melakukan *login* ke dalam *e-learning*. Namun, apabila belum memiliki akun, mahasiswa harus terlebih dahulu melakukan proses *registrasi* untuk dapat mengakses e-learning.

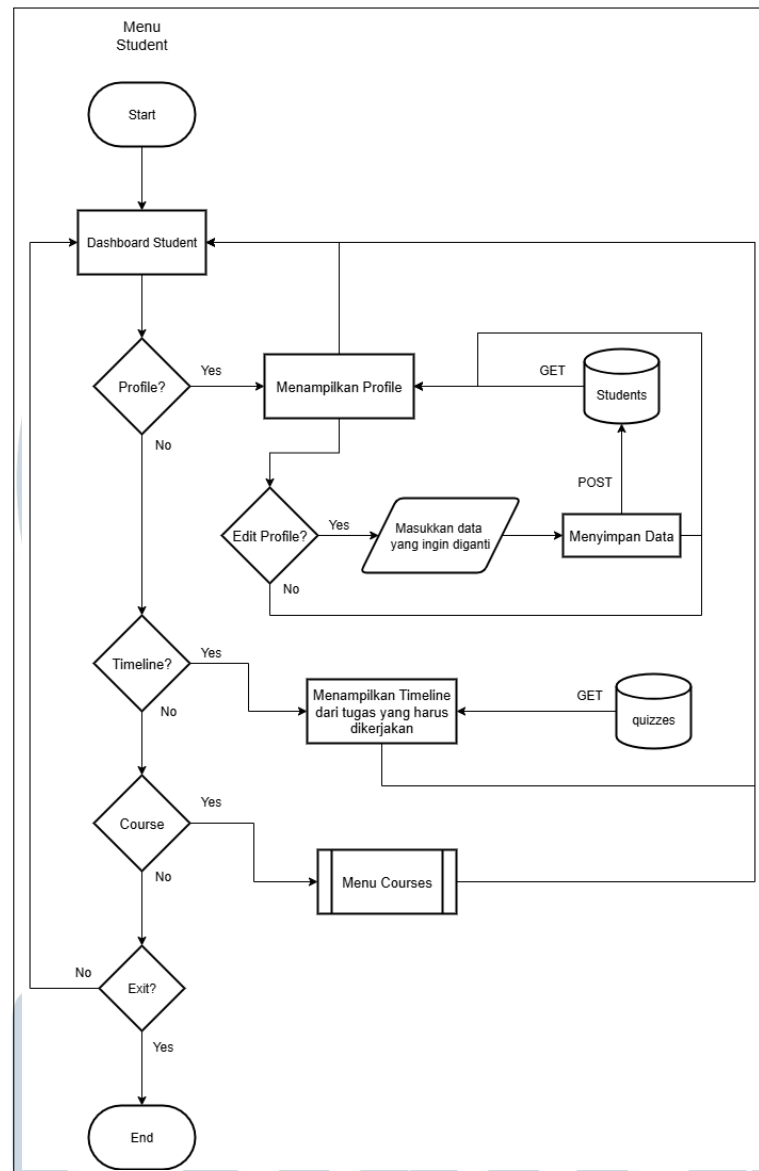
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.1. Flowchart Student *Login*

Pada Gambar 3.2, ditampilkan alur mahasiswa setelah berhasil melakukan *login*, yaitu masuk ke halaman *dashboard*. Halaman ini menampilkan informasi seperti *profile*, *timeline*, dan daftar *courses* atau mata kuliah yang diambil. Jika mahasiswa ingin mengubah data pribadi, maka dapat mengakses halaman *profile* untuk melakukan *edit*. Sementara itu, bagian *timeline* akan menampilkan urutan tugas berdasarkan *deadline* terdekat agar mahasiswa dapat mengetahui prioritas tugas yang harus dikerjakan terlebih dahulu.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

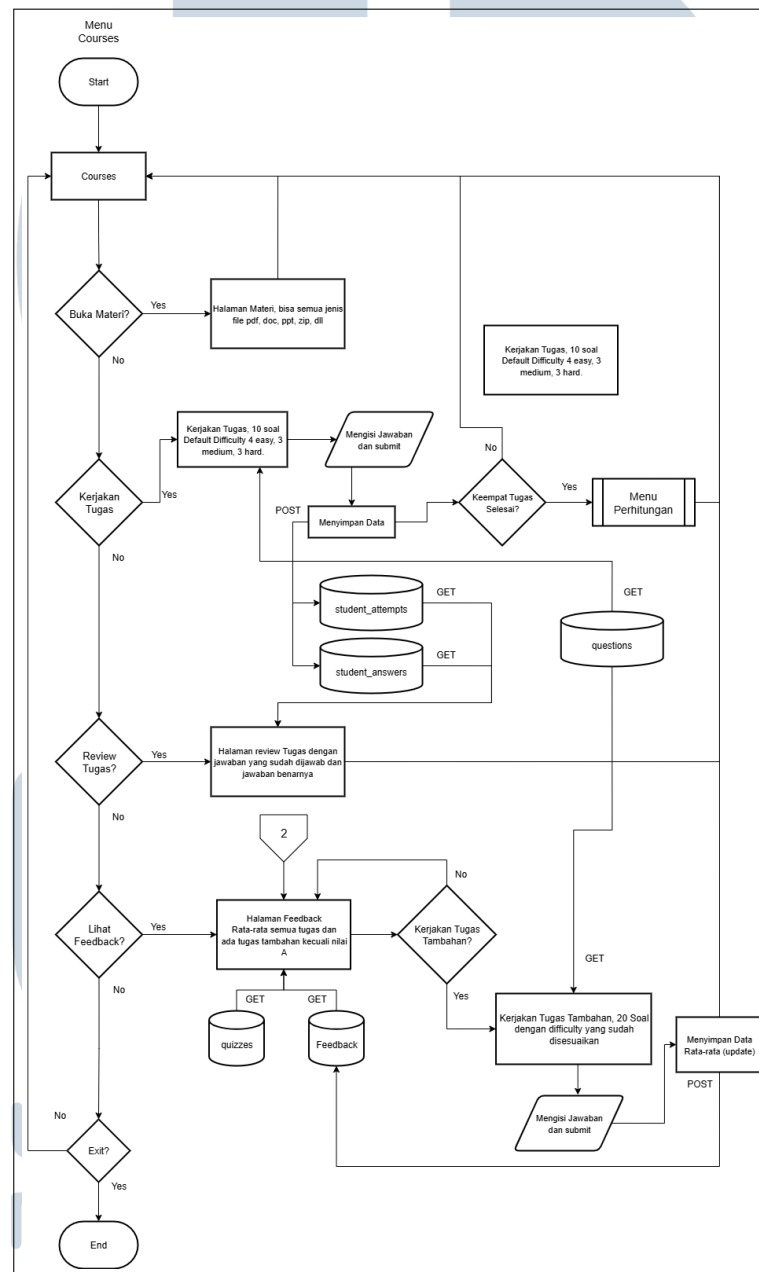


Gambar 3.2. Flowchart Dashboard Student

Pada Gambar 3.3, ditampilkan alur ketika mahasiswa mengakses halaman *course* atau mata kuliah tertentu. Di dalam halaman ini tersedia beberapa menu, antara lain: materi pembelajaran yang diberikan oleh dosen, daftar tugas, nilai dari tugas yang telah dikerjakan, serta halaman *feedback* yang dapat diakses setelah seluruh tugas selesai dikerjakan.

Halaman *feedback* berfungsi sebagai tempat penyampaian umpan balik sistem terhadap kinerja mahasiswa selama satu semester. Jika terdapat satu atau lebih tugas dengan nilai di bawah C atau 55, maka sistem secara otomatis akan menyediakan *tugas tambahan* dengan materi yang disesuaikan dari tugas-tugas

yang tidak lulus tersebut. Namun, apabila seluruh tugas yang dikerjakan lulus tetapi nilai rata-rata masih berada di bawah nilai A, maka sistem tetap akan menyediakan *tugas tambahan* bersifat opsional (tidak wajib), yang mencakup materi dari keempat tugas sebelumnya.

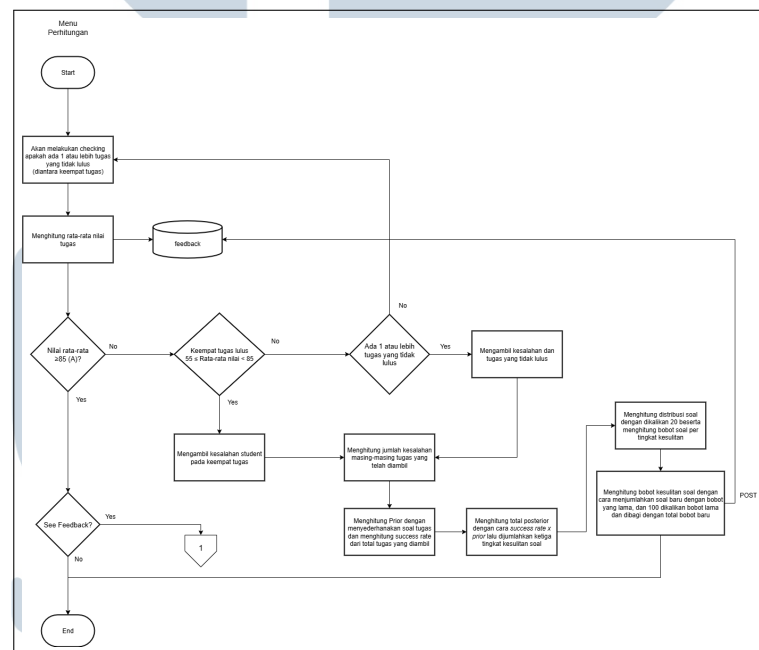


Gambar 3.3. Flowchart Course Student

Pada Gambar 3.4, ditampilkan alur ketika mahasiswa mengerjakan tugas terakhir atau tugas keempat dari suatu mata kuliah pada halaman *course*. Pada tahap

ini, sistem akan melakukan proses perhitungan berdasarkan hasil dari keempat tugas yang telah dikerjakan sebelumnya.

Jika semua tugas dinyatakan lulus dengan nilai sempurna (nilai A), maka sistem hanya akan menghitung rata-rata nilai mahasiswa tanpa memberikan *tugas tambahan*. Namun, apabila seluruh tugas dinyatakan lulus tetapi nilai rata-rata masih berada di bawah A, maka sistem akan tetap membuat *tugas tambahan* yang bersifat opsional. Materi dari tugas tambahan ini diambil secara proporsional dari keempat tugas yang telah dikerjakan. Sementara itu, jika terdapat satu atau lebih tugas yang tidak lulus (nilai di bawah C atau 55), sistem secara otomatis akan menghasilkan *tugas tambahan* sebanyak 20 soal. Soal-soal ini tidak dibagikan secara acak, melainkan disesuaikan dengan tingkat kesalahan dan performa mahasiswa. Dengan menggunakan model *Bayesian Network*, sistem akan menghitung jumlah soal berdasarkan tingkat kesulitan — yaitu mudah (*easy*), sedang (*medium*), dan sulit (*hard*) — sesuai dengan kelemahan mahasiswa pada tugas sebelumnya.

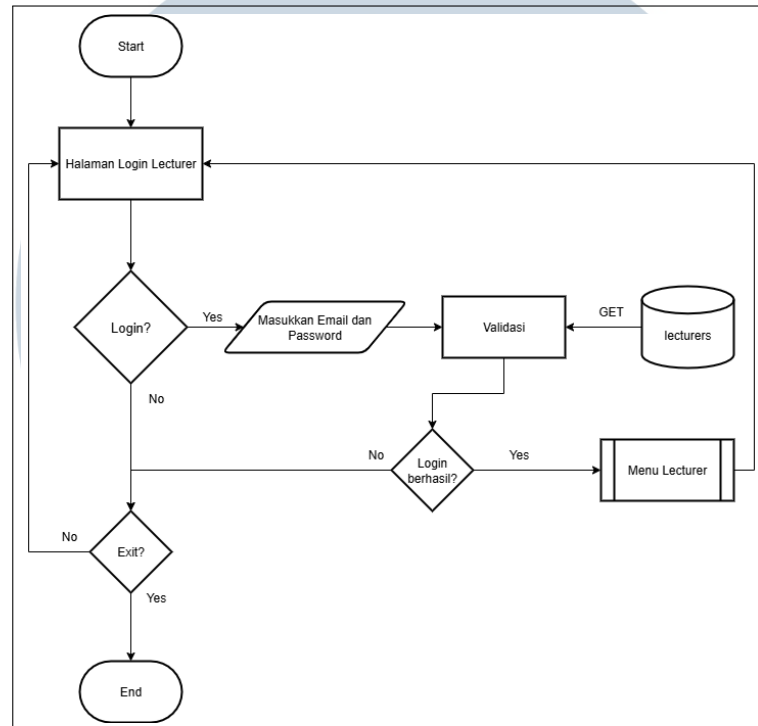


Gambar 3.4. Flowchart Perhitungan Student

B Flowchart Lecturer

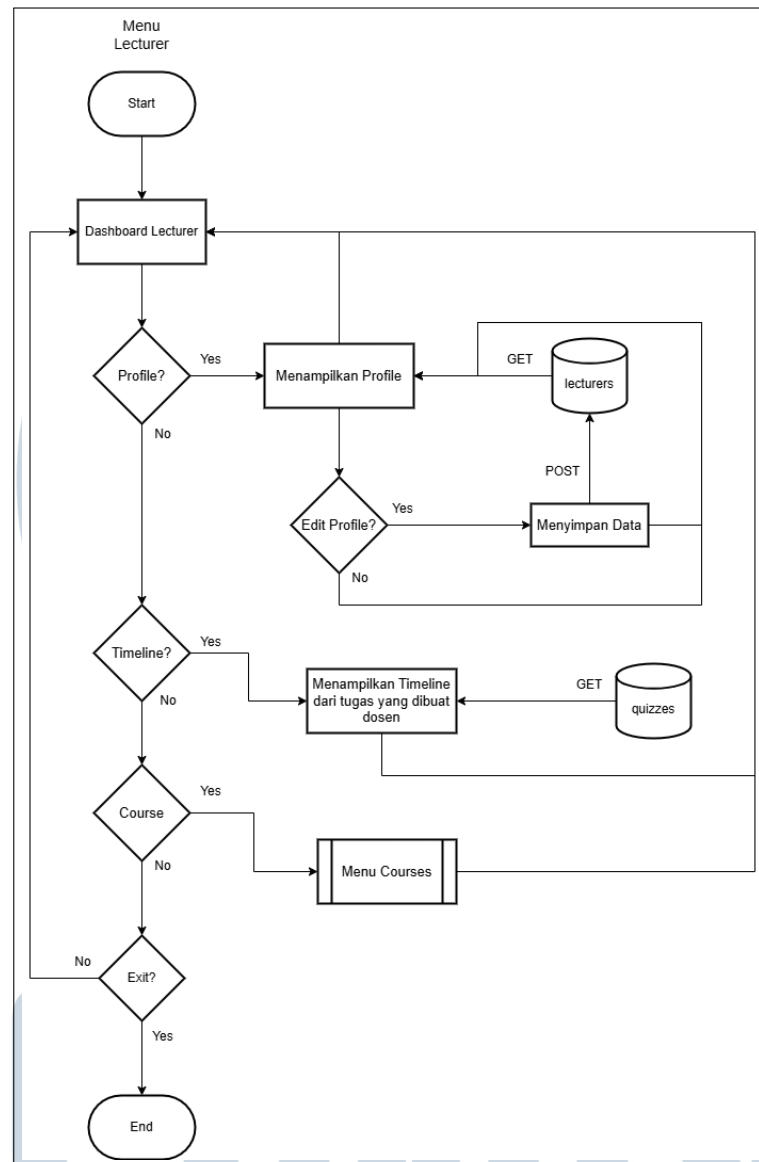
Pada Gambar 3.5, ditampilkan alur awal bagi dosen atau *lecturer*. Halaman pertama yang diakses adalah halaman *login*. Berbeda dengan mahasiswa, dosen

tidak dapat melakukan proses *registrasi* secara mandiri. Oleh karena itu, akun dosen harus dibuat terlebih dahulu oleh *admin* agar dapat digunakan untuk mengakses *e-learning*.



Gambar 3.5. Flowchart Lecturer *Login*

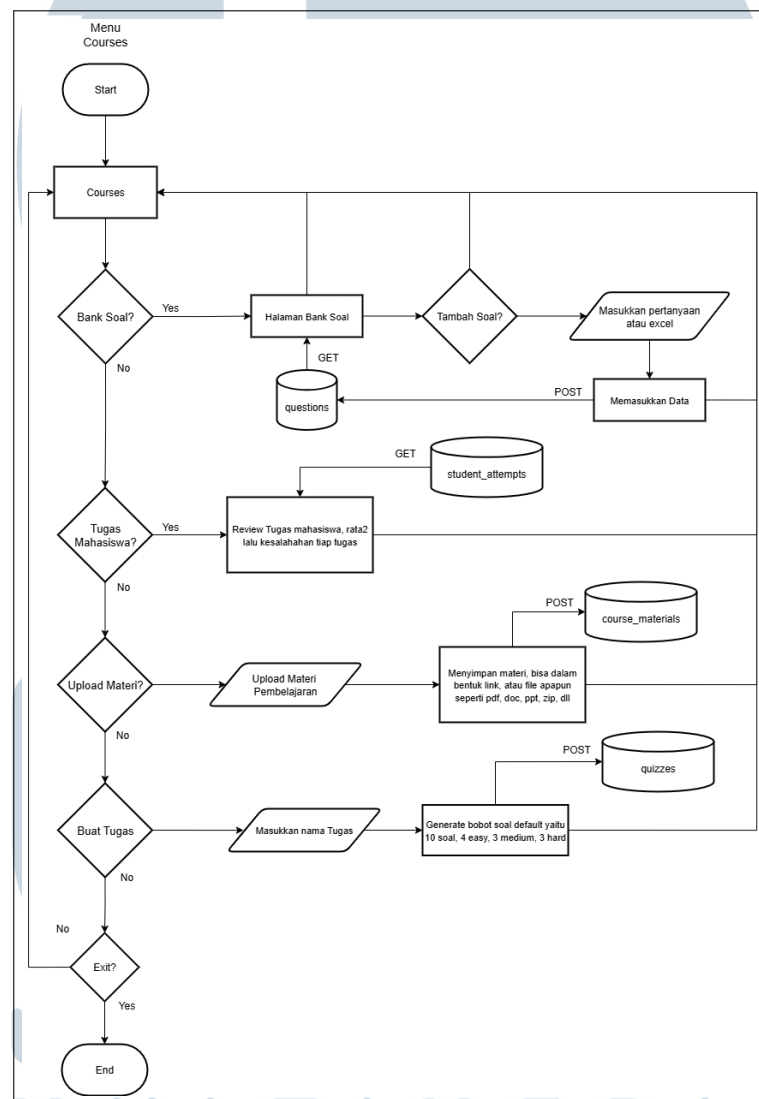
Pada Gambar 3.6, ditampilkan alur ketika dosen berhasil melakukan *login* ke dalam sistem. Setelah masuk, dosen akan diarahkan ke halaman utama yang menampilkan beberapa menu, serupa dengan yang dimiliki oleh mahasiswa. Perbedaan utama terletak pada isi halaman *profile*, karena informasi yang ditampilkan disesuaikan dengan data dosen. Selain itu, tampilan *timeline* pada akun dosen juga berbeda karena yang ditampilkan adalah daftar tugas yang telah dibuat oleh dosen tersebut untuk setiap mata kuliah yang diampu.



Gambar 3.6. Flowchart Menu Lecturer

Pada Gambar 3.7, ditampilkan alur ketika dosen mengakses menu *course* atau mata kuliah. Di dalam menu ini, dosen dapat mengunggah *materi pembelajaran* yang wajib diberikan setiap minggu. Pada minggu ke-4, ke-7, ke-10, dan ke-14, sistem akan menyediakan opsi untuk membuat tugas, sehingga dalam satu semester terdapat empat tugas yang harus disiapkan oleh dosen. Dosen dapat mengunggah materi dalam berbagai format file seperti *.zip*, *.pdf*, *.doc*, *.ppt*, dan lainnya. Sebelum membuat soal, dosen harus terlebih dahulu mengakses *bank soal* untuk menginput pertanyaan beserta tingkat kesulitannya (*difficulty level*), baik secara manual maupun melalui fitur *import* dalam format *Excel*. Setelah soal

tersedia dalam *bank soal*, dosen dapat membuat tugas berdasarkan materi atau topik tertentu. Secara default, sistem akan membangkitkan 10 soal dengan komposisi: 4 soal mudah (*easy*), 3 soal sedang (*medium*), dan 3 soal sulit (*hard*). Setelah mahasiswa menyelesaikan tugas, dosen dapat meninjau hasil pengerjaan mahasiswa secara keseluruhan, termasuk melihat nilai masing-masing tugas dan rata-rata akhir dari setiap mahasiswa.

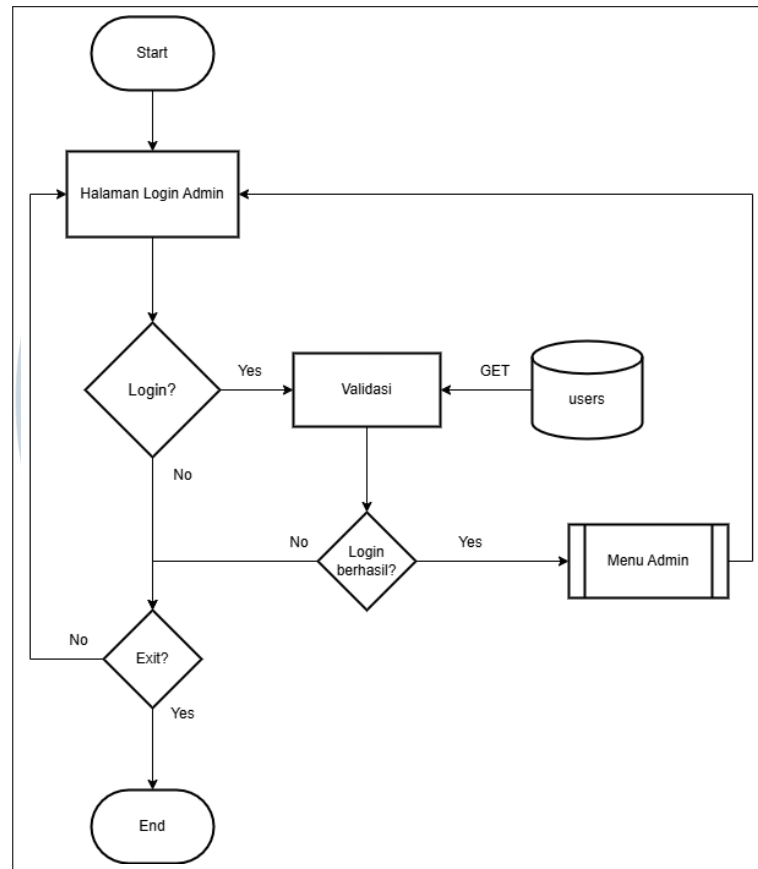


Gambar 3.7. Flowchart Course Lecturer

C Flowchart Admin

Pada Gambar 3.1, ditampilkan alur awal bagi admin. Halaman pertama yang diakses adalah *login page*. Admin bisa langsung melakukan validasi untuk masuk

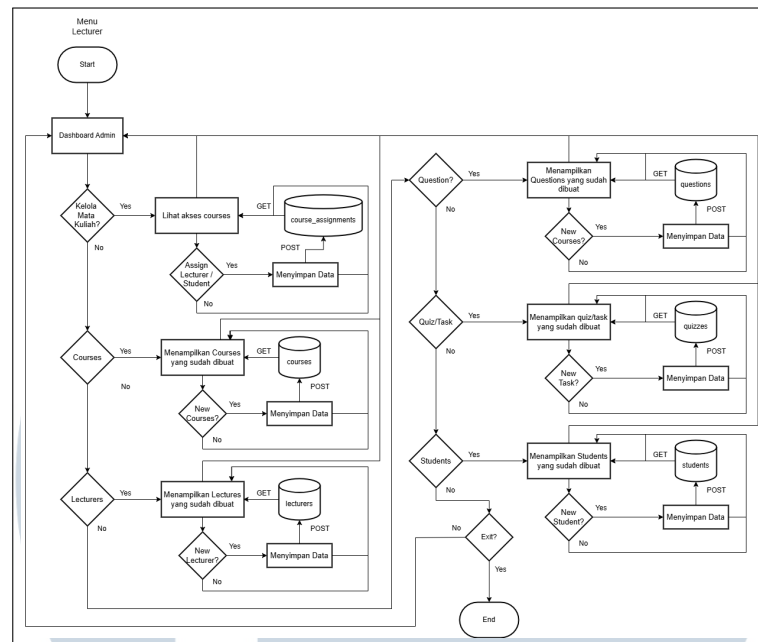
kebagian admin.



Gambar 3.8. Flowchart Admin *Login*

Pada Gambar 3.2, adalah saat admin sudah berhasil masuk yaitu admin dapat melakukan apa yang bisa dilakukan oleh dosen dan admin dapat membuat akun untuk mahasiswa dan dosen, admin dapat mengatur mata kuliah sehingga mahasiswa dan dosen dapat mengakses mata kuliah tersebut.

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



Gambar 3.9. Flowchart Menu Admin

3.2.2 Table Database

Pada Tabel 3.1 ditampilkan struktur tabel *courses* yang berfungsi untuk menyimpan informasi mengenai mata kuliah. Tabel ini mencakup kolom *course_name* sebagai nama mata kuliah, dan *course_code* sebagai kode unik untuk membedakan setiap mata kuliah. Kolom *course_milik* memiliki tujuan untuk menunjukkan program studi sehingga *couses* tersebut dapat diketahui untuk program studinya.

Tabel 3.1. Tabel Struktur Database *courses*

Nama Kolom	Tipe Data	Atribut Key
id	bigint(20)	Primary Key, Auto-increment
course_code	varchar(255)	Unique
course_name	varchar(255)	-
course_milik	enum	-
created_at	Timestamp()	-
updated_at	Timestamp()	-

Pada Tabel 3.2 ditampilkan struktur dari tabel *quizzes* yang bertujuan untuk

menyimpan tugas yang telah dibuat dosen seperti tugas 1 hingga tugas tambahan.

Tabel 3.2. Tabel Struktur Database quizzes

Nama Kolom	Tipe Data	Atribut Key
id	bigint(20)	Primary Key, Auto-increment
course_code	varchar(255)	Foreign Key
task_number	int(11)	-
title	varchar(255)	-
start_time	datetime	Nullable
end_time	datetime	Nullable
created_at	Timestamp()	-
updated_at	Timestamp()	-

Pada Tabel 3.3 ditampilkan struktur dari tabel *students* yang bertujuan untuk menyimpan data mahasiswa dengan nim, nama, email, password, major, angkatan, dan profile_photo dan motto yang memiliki nilai *default*

Tabel 3.3. Tabel Struktur Database students

Nama Kolom	Tipe Data	Atribut Key
id	bigint(20)	Primary Key, Auto-increment
nim	varchar(255)	Unique
name	varchar(255)	-
email	varchar(255)	Unique
password	varchar(255)	-
major	varchar(255)	-
angkatan	year	-
profile_photo	varchar(255)	-
motto	varchar(255)	-
created_at	Timestamp()	-
updated_at	Timestamp()	-

Pada Tabel 3.4 ditampilkan struktur dari tabel *feedback* yang bertujuan untuk menyimpan hasil akhir dari mahasiswa seperti rata-rata tugas, tugas yang

gagal, dan distribusi tugas tambahan.

Tabel 3.4. Tabel Struktur Database feedback

Nama Kolom	Tipe Data	Atribut Key
id	bigint(20)	Primary Key, Auto-increment
student_id	bigint(20)	Foreign Key
course_id	bigint(20)	Foreign Key
average_score	float	Nullable
feedback_text	text	Nullable
failed_tasks	json	Nullable
question_distribution	json	Nullable
question_weights	json	Nullable
task_distribution	json	Nullable
additional_quiz_id	bigint(20)	Foreign Key, Nullable
created_at	Timestamp()	-
updated_at	Timestamp()	-

Pada Tabel 3.5 ditampilkan struktur dari tabel `student_attempts` yang bertujuan untuk menyimpan semua aktivitas tugas dari mahasiswa beserta dengan nilai dan kesalahan (*error*) dari masing-masing tugas yang telah dikerjakan oleh mahasiswa.

Tabel 3.5. Tabel Struktur Database `student_attempts`

Nama Kolom	Tipe Data	Atribut Key
id	bigint(20)	Primary Key, Auto-increment
student_id	bigint(20)	Foreign Key
quiz_id	bigint(20)	Foreign Key
course_id	bigint(20)	Foreign Key
task_number	tinyint	-
score	int(11)	-
errors_easy	int(11)	-
errors_medium	int(11)	-

Tabel 3.5. Lanjutan dari Tabel `student_attempts`

Nama Kolom	Tipe Data	Atribut Key
<code>errors_hard</code>	<code>int(11)</code>	-
<code>created_at</code>	<code>Timestamp()</code>	-
<code>updated_at</code>	<code>Timestamp()</code>	-

Pada Tabel 3.5 ditampilkan struktur dari tabel `student_answers` yang bertujuan untuk menyimpan hasil dari aktivitas mahasiswa dalam mengerjakan tugas secara lebih detail, termasuk informasi mengenai *question* yang telah dikerjakan. Sehingga, sistem dapat mengetahui bahwa mahasiswa telah menjawab soal tertentu.

Tabel 3.6. Tabel Struktur Database `student_answers`

Nama Kolom	Tipe Data	Atribut Key
<code>id</code>	<code>bigint(20)</code>	Primary Key, Auto-increment
<code>attempt_id</code>	<code>bigint(20)</code>	Foreign Key
<code>question_id</code>	<code>bigint(20)</code>	Foreign Key
<code>selected_option</code>	<code>char(1)</code>	-
<code>is_correct</code>	<code>boolean</code>	-
<code>created_at</code>	<code>Timestamp()</code>	-
<code>updated_at</code>	<code>Timestamp()</code>	-

Pada Tabel 3.7 ditampilkan struktur dari tabel `lecturers` yang bertujuan untuk menyimpan data dosen. Struktur ini serupa dengan data mahasiswa, namun dibedakan karena dosen memiliki mata kuliah untuk kegiatan mengajar.

Tabel 3.7. Tabel Struktur Database `lecturers`

Nama Kolom	Tipe Data	Atribut Key
<code>id</code>	<code>bigint(20)</code>	Primary Key, Auto-increment
<code>nidn</code>	<code>varchar(255)</code>	Unique
<code>name</code>	<code>varchar(255)</code>	-

Tabel 3.7. Lanjutan dari Tabel lecturers

Nama Kolom	Tipe Data	Atribut Key
email	varchar(255)	Unique
password	varchar(255)	-
major	varchar(255)	-
mata_kuliah	varchar(255)	-
profile_photo	varchar(255)	-
motto	varchar(255)	-
created_at	Timestamp()	-
updated_at	Timestamp()	-

Pada Tabel 3.8 ditampilkan struktur dari tabel `course_materials` yang bertujuan untuk menyimpan materi pembelajaran dari setiap mata kuliah. Tabel ini mencakup beberapa data penting, seperti `files` yang disimpan dalam format json karena dosen dapat memberikan materi lebih dari satu file dalam berbagai format, seperti PDF, Word, PowerPoint, ZIP, dan sebagainya. Kolom `video` berisi tautan video pembelajaran, seperti YouTube, jika diperlukan. Sedangkan `is_optional` digunakan untuk menandai apakah materi pada minggu tersebut bersifat opsional atau tidak wajib dipelajari oleh mahasiswa.

Tabel 3.8. Tabel Struktur Database `course_materials`

Nama Kolom	Tipe Data	Atribut Key
id	bigint(20)	Primary Key, Auto-increment
course_id	bigint(20)	Foreign Key
week	int(11)	-
files	json	Nullable
video_url	varchar(255)	Nullable
is_optional	boolean	-
created_at	Timestamp()	-
updated_at	Timestamp()	-

Pada Tabel 3.9 ditampilkan struktur dari tabel `course_assignments` yang bertujuan untuk menyimpan data dosen dan mahasiswa untuk akses pada mata

kuliah tertentu yang dapat dilakukan oleh admin, jika mahasiswa melakukan pendaftaran akun maka yang akan dilakukan sistem adalah membuat akses untuk mahasiswa tersebut dapat mengakses mata kuliah pada jurusan atau program studi tersebut.

Tabel 3.9. Tabel Struktur Database *course_assignments*

Nama Kolom	Tipe Data	Atribut Key
id	bigint(20)	Primary Key, Auto-increment
course_code	varchar(255)	Foreign Key
lecturer_id	bigint(20)	Foreign Key, Nullable
student_id	bigint(20)	Foreign Key, Nullable
created_at	Timestamp()	-
updated_at	Timestamp()	-

Pada Tabel 3.9 ditampilkan struktur dari tabel *course_assignments* yang bertujuan untuk menyimpan *question* atau pertanyaan yang telah dibuat oleh dosen dengan pilihan ganda beserta dengan tingkat kesulitan masing-masing pertanyaan. Nantinya pada *website* akan terdapat 2 pilihan yaitu memasukkan data secara manual atau menggunakan excel atau csv.

Tabel 3.10. Tabel Struktur Database *questions*

Nama Kolom	Tipe Data	Atribut Key
id	bigint(20)	Primary Key, Auto-increment
question_text	varchar(255)	-
option_a	varchar(255)	-
option_b	varchar(255)	-
option_c	varchar(255)	-
option_d	varchar(255)	-
correct_option	varchar(255)	-
task_number	tinyint	Nullable
course_id	bigint(20)	Foreign Key

Tabel 3.10. Lanjutan dari Tabel *questions*

Nama Kolom	Tipe Data	Atribut Key
difficulty	enum	-
image	varchar(255)	Nullable
created_at	Timestamp()	-
updated_at	Timestamp()	-

3.2.3 Rancangan Database

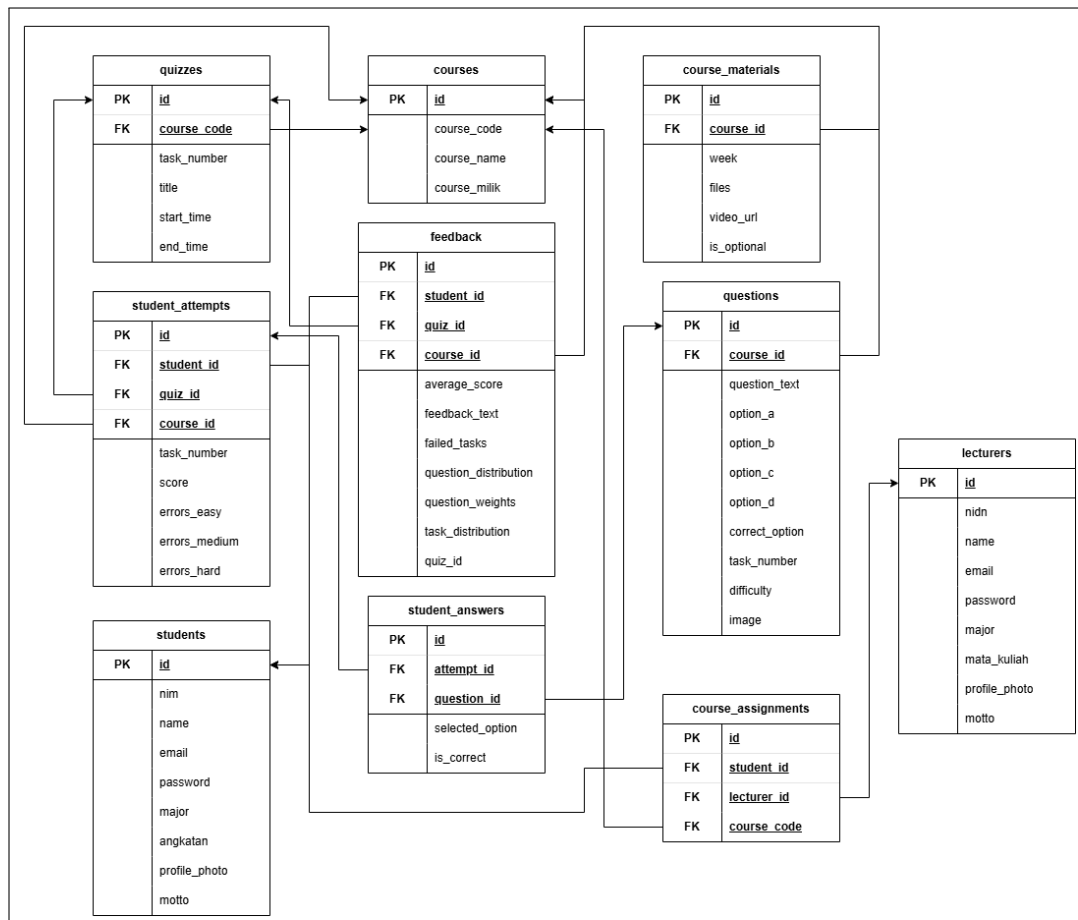
Perancangan *database* pada website e-learning dilakukan menggunakan *MySQL* dan terdiri dari 10 tabel utama. Tabel-tabel ini saling terhubung untuk mendukung seluruh fungsionalitas sistem.

Tabel utama yang digunakan adalah *courses*, yang berfungsi untuk menyimpan data mata kuliah yang akan ditampilkan pada platform e-learning. Selanjutnya, terdapat tabel *students* dan *lecturers* yang digunakan untuk menyimpan data pengguna berupa mahasiswa dan dosen.

Pada Gambar 3.10, skema *database* digambarkan secara lebih rinci. Tabel *quizzes* menyimpan data tugas yang digenerasi oleh dosen. Tabel *course_materials* digunakan untuk menyimpan materi pembelajaran dalam setiap mata kuliah.

Tabel *student_attempts* dan *student_answers* berfungsi untuk mencatat jawaban mahasiswa serta kesalahan yang dilakukan saat mengerjakan tugas. Tabel *course_assignments* digunakan untuk mengelola akses mahasiswa dan dosen terhadap mata kuliah yang tersedia.

Tabel *questions* berperan sebagai *bank soal* yang menyimpan soal-soal yang nantinya akan ditampilkan dalam tugas. Terakhir, tabel *feedback* menyimpan hasil analisis sistem yang merupakan fitur utama dari e-learning ini, termasuk data tugas tambahan, bobot soal, bobot nilai, rata-rata nilai mahasiswa, serta daftar kesalahan pada tiap tugas.

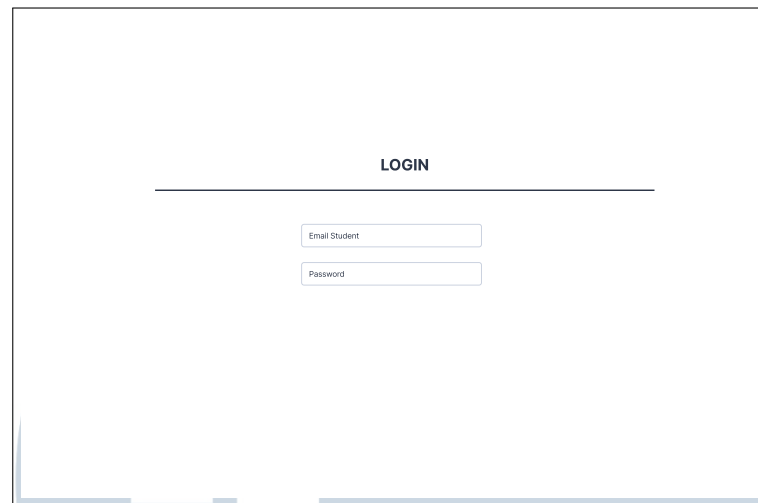


Gambar 3.10. Skema Database

3.2.4 Rancangan Mockup

Dalam perencanaan pembuatan sebuah *website*, diperlukan rancangan tampilan awal sebagai acuan agar antarmuka yang dikembangkan sesuai dengan tujuan sistem dibangun. Rancangan ini dituangkan dalam bentuk *mockup*, yang berfungsi sebagai gambaran awal (*kasar*) dalam proses desain dan pengembangan, sehingga arah pengembangan tidak menyimpang dari kebutuhan utama pengguna.

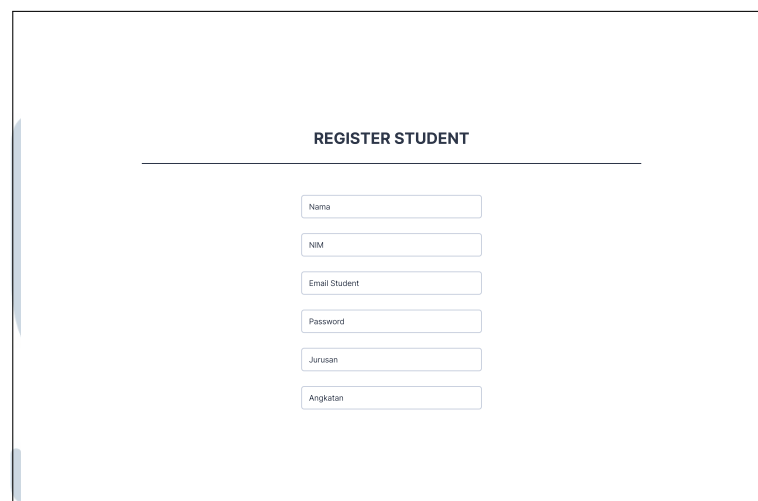
Pada Gambar 3.11 ditampilkan tampilan awal yang digunakan oleh mahasiswa dan dosen. Keduanya memiliki tampilan halaman *login* yang serupa, namun setelah proses validasi berhasil, masing-masing pengguna akan diarahkan ke halaman yang berbeda sesuai dengan peran atau aksesnya dalam sistem.



A mockup of a login page. At the top center, the word "LOGIN" is displayed in a bold, sans-serif font. Below it is a horizontal line. Underneath the line are two input fields: the first is labeled "Email Student" and the second is labeled "Password". Both fields are simple rectangular boxes with a thin border.

Gambar 3.11. Mockup Login Page

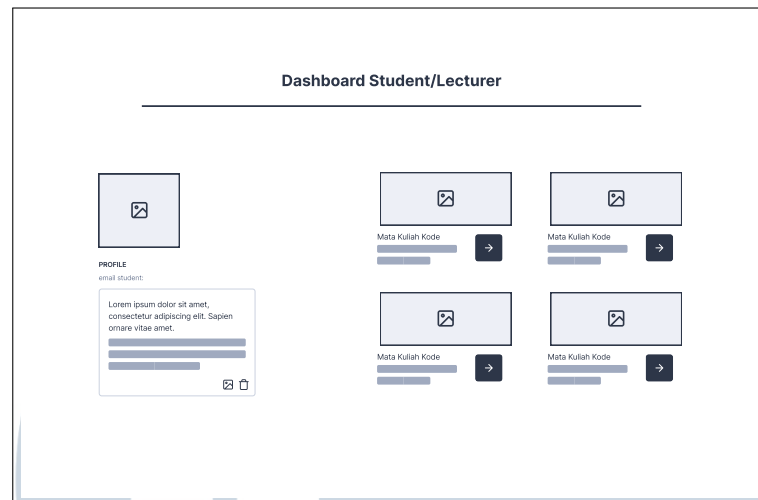
Pada Gambar 3.12 ditampilkan halaman *register* khusus untuk mahasiswa. Dalam sistem *e-learning* ini, fitur registrasi hanya disediakan bagi *student* yang belum memiliki akun. Sedangkan untuk *lecturer*, proses pembuatan akun dilakukan oleh *admin*, sehingga dosen tidak dapat melakukan registrasi secara mandiri melalui sistem.



A mockup of a student registration page. At the top center, the words "REGISTER STUDENT" are displayed in a bold, sans-serif font. Below it is a horizontal line. Underneath the line are seven input fields stacked vertically: "Nama", "NIM", "Email Student", "Password", "Jurusan", and "Angkatan". Each field is a simple rectangular box with a thin border.

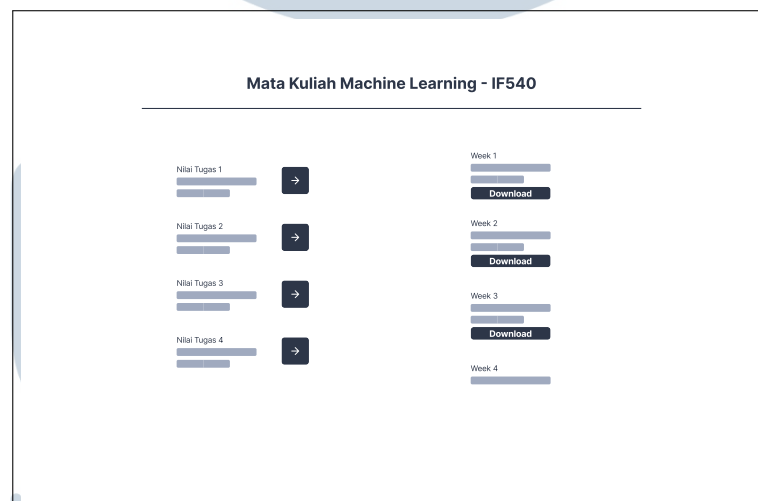
Gambar 3.12. Mockup Register Page

Pada Gambar 3.13 ditampilkan halaman *dashboard* awal yang diakses oleh mahasiswa maupun dosen setelah berhasil melakukan *login*. Tampilan *dashboard* secara umum memiliki struktur antarmuka yang serupa, namun isi konten di dalamnya akan disesuaikan berdasarkan level pengguna (*user level*) masing-masing, yaitu apakah pengguna merupakan *student* atau *lecturer*.



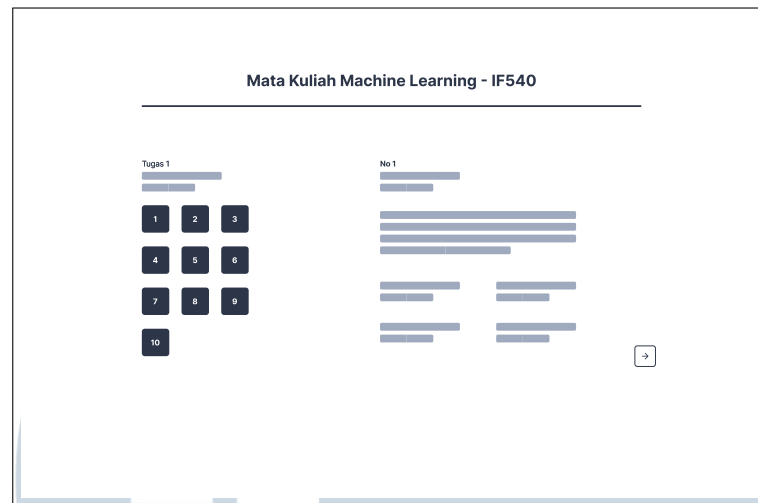
Gambar 3.13. Mockup Register Page

Pada Gambar 3.14 ditampilkan tampilan saat mahasiswa mengakses halaman mata kuliah atau *course*. Pada halaman ini, *student* dapat melihat dan mengunduh materi pembelajaran, serta mengerjakan tugas yang telah dibuat oleh *lecturer*. Setiap mata kuliah akan memiliki 4 tugas selama 1 semester.



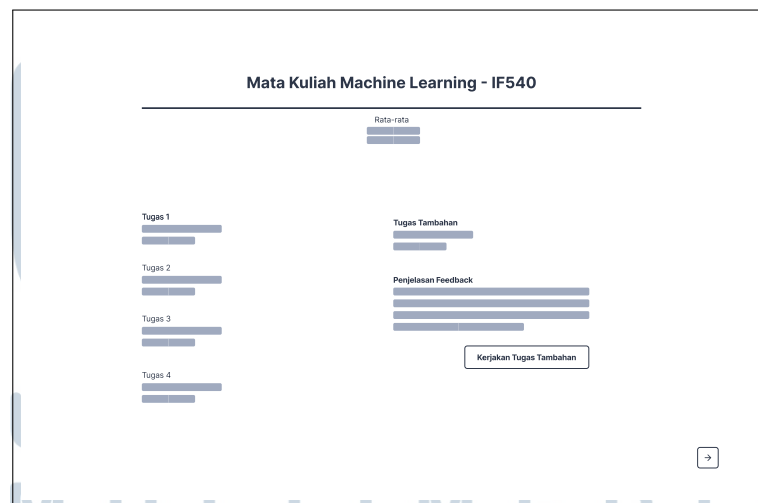
Gambar 3.14. Mockup Course Student

Pada Gambar 3.15 ditampilkan tampilan ketika mahasiswa mengerjakan tugas yang diberikan oleh *lecturer*. Setiap soal ditampilkan dengan nomor yang jelas, sehingga *student* dapat melakukan *skip* atau melewati pertanyaan tertentu terlebih dahulu. Namun, saat akan menekan tombol *submit*, sistem akan menampilkan pesan kesalahan jika masih terdapat soal yang belum dijawab secara lengkap.



Gambar 3.15. Mockup Kerjakan Tugas Student

Pada Gambar 3.16 ditampilkan tampilan halaman *feedback* yang muncul setelah mahasiswa menyelesaikan keempat tugas. Sistem akan memberikan umpan balik secara otomatis berdasarkan hasil pengerjaan tugas-tugas tersebut. Jika terdapat satu atau lebih tugas yang tidak lulus, atau jika seluruh tugas dinyatakan lulus namun rata-rata nilainya berada di bawah A (85), maka sistem akan memberikan *Tugas Tambahan* secara otomatis sebagai bentuk remedial adaptif.

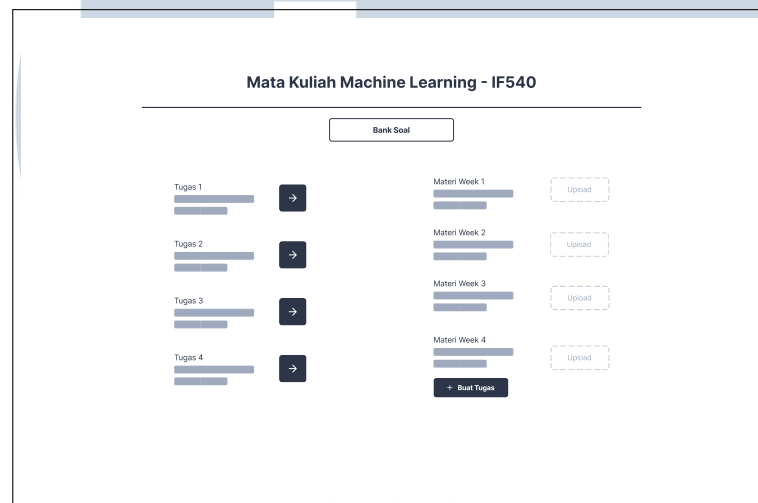


Gambar 3.16. Mockup Feedback Student

Pada Gambar 3.17 ditampilkan halaman *course* yang diakses oleh *lecturer* atau dosen. Dosen memiliki akses untuk mengunggah soal ke dalam sistem baik secara manual maupun melalui berkas berformat *Excel*, serta dapat mengunggah

materi pembelajaran dalam berbagai format file seperti *ZIP*, *PDF*, *DOC*, *PPT*, dan lainnya.

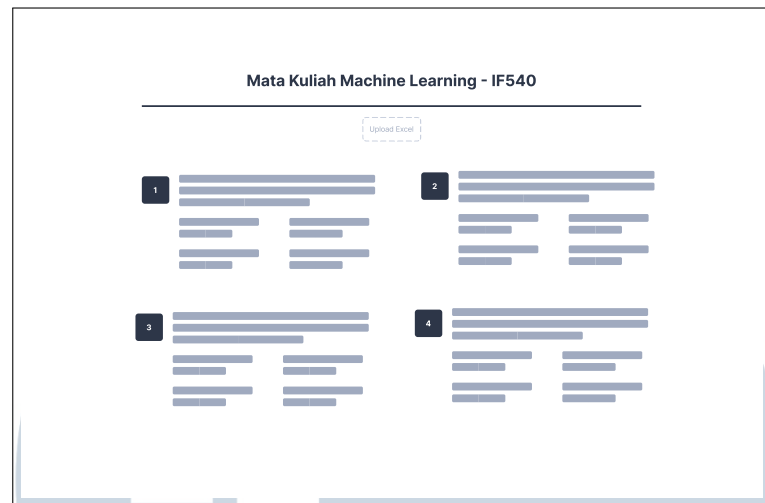
Selain itu, dosen juga dapat membuat atau melakukan *generate* Tugas 1 hingga Tugas 4 yang akan diberikan kepada mahasiswa. Sistem akan secara otomatis memilih soal secara acak dari *bank soal* berdasarkan tingkat kesulitan awal, yaitu 4 soal mudah (*easy*), 3 soal sedang (*medium*), dan 3 soal sulit (*hard*). Soal akan diacak berdasarkan tingkatan kesulitan tersebut.



Gambar 3.17. Mockup Course Lecturer

Pada Gambar 3.18 ditampilkan tampilan *bank soal* yang diakses oleh dosen. Pada halaman ini, dosen dapat mengunggah atau membuat soal secara manual maupun melalui berkas *Excel*. Setiap soal yang dimasukkan akan dilengkapi dengan kunci jawaban serta tingkat kesulitan (*difficulty*) yang ditetapkan secara manual oleh dosen.

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



Gambar 3.18. Mockup Bank Soal Lecturer

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA