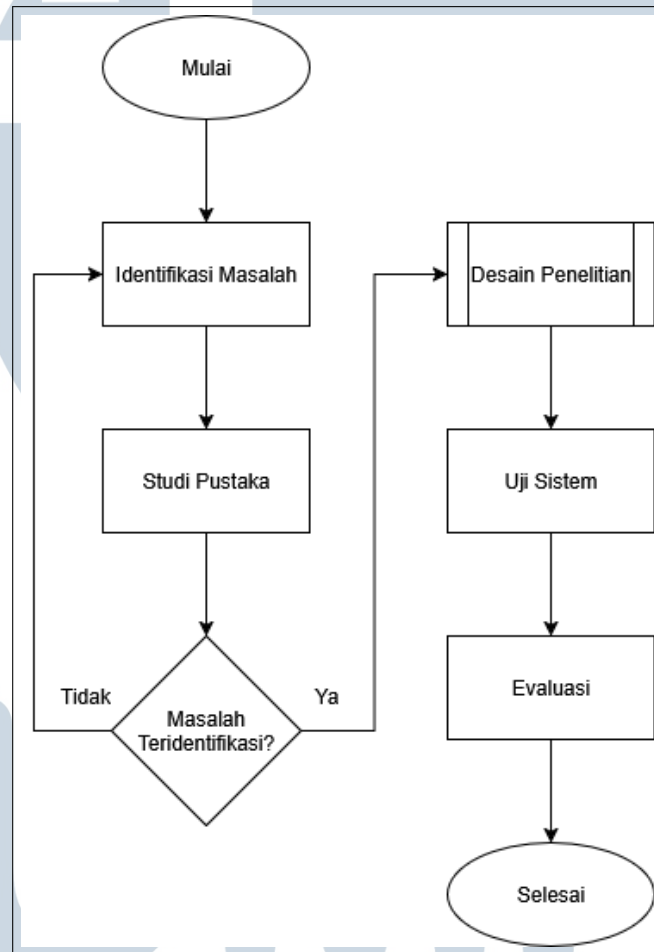


BAB 3

METODOLOGI PENELITIAN

Gambar 3.1 menunjukkan alur proses metodologi penelitian yang dilakukan dalam penerapan aplikasi ITS berbasis RL.



Gambar 3.1. Diagram alur proses metodologi penelitian

3.1 Identifikasi Masalah

Pada tahapan ini dilakukan identifikasi masalah berdasarkan berbagai sumber kredibel seperti artikel, jurnal ilmiah, dan data statistik internasional. Berdasarkan survei global yang dilakukan oleh *United Nations International Children's Emergency Fund* (UNICEF) pada tahun 2023 terhadap siswa usia 15 tahun di 15 negara, ditemukan bahwa 52% responden menganggap matematika sebagai mata pelajaran paling sulit dalam proses pembelajaran *hybrid* (gabungan

daring dan luring) [7]. Secara lebih spesifik, 63% di antaranya menyatakan kesulitan dalam memahami materi aljabar, yang merupakan dasar penting dalam kurikulum matematika menengah.

Selain itu, data dari *Programme for International Student Assessment* (PISA) 2022 yang dirilis oleh *Organisation for Economic Co-operation and Development* (OECD), menunjukkan bahwa hanya 31% siswa usia 15 tahun di negara-negara anggota OECD yang berhasil mencapai tingkat kemahiran dasar dalam matematika [8]. Sebaliknya, 42% siswa tidak mampu mencapai level minimum (Level 2) dalam penguasaan konsep matematika dasar seperti aljabar dan fungsi. Temuan ini menegaskan adanya kesenjangan pemahaman dan kemampuan siswa dalam pelajaran matematika yang sangat signifikan.

Berdasarkan informasi dari kedua sumber tersebut, dapat diidentifikasi bahwa tantangan utama dalam proses pembelajaran matematika adalah kurangnya sistem yang mampu menyesuaikan tingkat kesulitan materi dengan kemampuan siswa. Oleh karena itu, diperlukan pendekatan inovatif berupa penerapan algoritma *Reinforcement Learning* (RL) pada *Intelligent Tutoring System* (ITS). Sistem ini diharapkan mampu menyesuaikan konten soal berdasarkan performa siswa melalui proses interaksi berkelanjutan, sehingga menciptakan lingkungan belajar yang personal, dan mampu mengurangi kesenjangan dalam penguasaan materi matematika di kalangan pelajar.

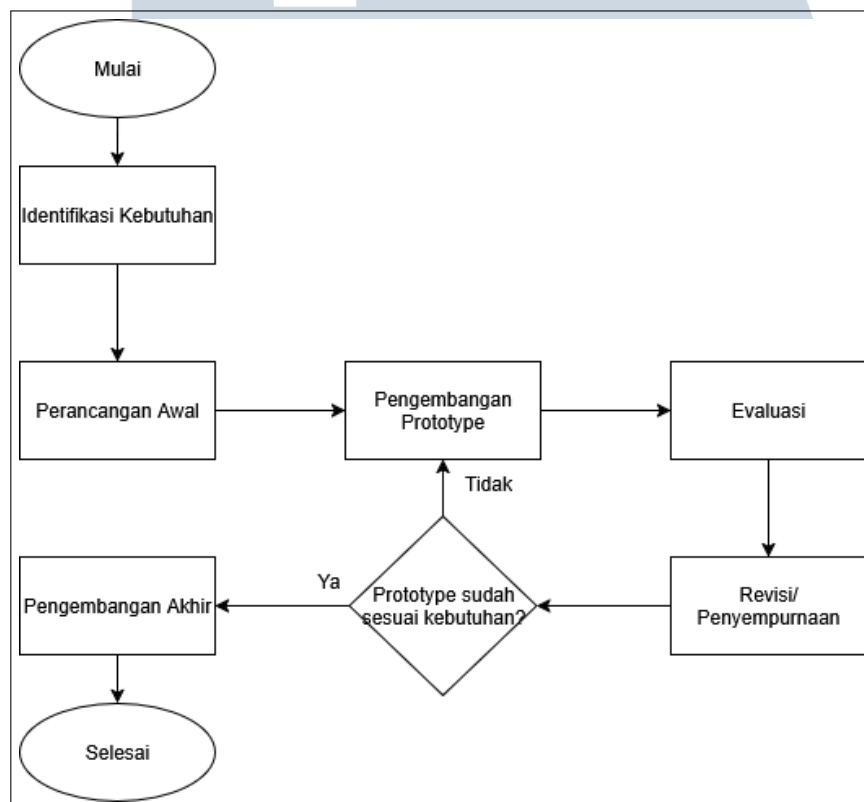
3.2 Studi Pustaka

Pada tahapan ini dilakukan studi pustaka yang bertujuan untuk mengumpulkan dan mempelajari berbagai literatur yang relevan sebagai landasan teoritis dalam penelitian. Literatur yang dikaji mencakup kajian mengenai *Intelligent Tutoring System* (ITS), konsep dan algoritma *Reinforcement Learning* (RL), serta metode adaptasi tingkat kesulitan soal berbasis pengalaman belajar siswa. Selain itu, turut dikaji pula literatur yang membahas penerapan algoritma *Reinforcement Learning* dalam sistem pembelajaran, khususnya model *Q-learning* dalam konteks pendidikan. Studi pustaka ini juga mencakup referensi pengembangan aplikasi berbasis web menggunakan HTML, CSS, JavaScript, Flask, dan Jupyter Notebook sebagai media implementasi sistem ITS, serta literatur terkait perancangan alur sistem secara terstruktur. Literatur-literatur tersebut diperoleh dari berbagai sumber terpercaya, seperti artikel ilmiah, jurnal, prosiding konferensi, dokumentasi teknis, dan hasil penelitian sebelumnya yang memiliki

keterkaitan erat dengan fokus penelitian ini.

3.3 Desain Penelitian

Penelitian pada Gambar 3.2 ini bertujuan untuk menerapkan sistem *Intelligent Tutoring System* (ITS) berbasis *Reinforcement Learning* (RL) guna mempersonalisasi tingkat kesulitan soal matematika.



Gambar 3.2. Diagram Metodologi Prototyping

Model pengembangan yang digunakan mengadaptasi model *Prototyping*, di mana sistem dikembangkan secara iteratif melalui pembuatan prototipe awal, pengujian, evaluasi, dan perbaikan berdasarkan umpan balik. Pemilihan pendekatan ini didasarkan pada kebutuhan untuk menghasilkan sistem ITS yang mampu beradaptasi terhadap performa masing-masing siswa. Dengan mengintegrasikan teknologi *Reinforcement Learning*, sistem dapat menentukan level soal (mudah, sedang, sulit) secara otomatis dan memberikan jalur pembelajaran yang bersifat individual.

Pendekatan prototyping memungkinkan keterlibatan langsung pengguna dalam proses pengembangan, sehingga sistem dapat disesuaikan secara dinamis

berdasarkan interaksi aktual. Hal ini penting untuk memastikan bahwa sistem ITS yang dibangun tidak hanya adaptif secara teknis, tetapi juga sesuai dengan kebutuhan nyata pengguna dalam konteks pembelajaran matematika.

3.4 Identifikasi Kebutuhan Sistem

Proses identifikasi kebutuhan sistem dilakukan untuk merumuskan permasalahan yang dihadapi oleh pelajar serta menentukan fitur dan fungsi utama yang dibutuhkan dalam pengembangan aplikasi *Intelligent Tutoring System* (ITS). Berdasarkan hasil studi literatur dan data global dari UNICEF (2023) dan OECD PISA (2022), ditemukan bahwa matematika merupakan mata pelajaran yang paling menantang dalam sistem pembelajaran daring maupun *hybrid*. Mayoritas pelajar usia 15 tahun mengalami kesulitan dalam memahami materi aljabar dasar dan menunjukkan performa yang rendah dalam asesmen literasi matematika. Permasalahan utama yang diidentifikasi adalah belum adanya sistem pembelajaran digital yang mampu menyesuaikan tingkat kesulitan soal berdasarkan performa belajar masing-masing siswa.

Berdasarkan identifikasi masalah tersebut, tujuan utama dari pengembangan sistem ini adalah menerapkan ITS berbasis algoritma *Reinforcement Learning* (RL), khususnya model *Q-Learning*, yang mampu mempersonalisasi tingkat kesulitan soal matematika. Penerapan aplikasi ini diharapkan mampu menyesuaikan penyajian soal berdasarkan interaksi pengguna, dengan fitur utama berupa pelacakan performa siswa, penyesuaian level soal (mudah, sedang, sulit), serta pemilihan soal berdasarkan *reward* dan *state* pembelajaran. Selain itu, sistem juga harus terintegrasi dengan antarmuka web berbasis HTML, CSS, JavaScript, dan Flask, serta mendukung eksekusi melalui Jupyter Notebook, guna menciptakan pengalaman belajar yang personal, efisien, dan berbasis data.

3.5 Perancangan Awal

Tahap kedua dalam proses penelitian ini adalah perancangan awal dari sistem aplikasi *Intelligent Tutoring System* (ITS) yang berfokus pada personalisasi tingkat kesulitan soal matematika berdasarkan performa siswa menggunakan pendekatan algoritma *Reinforcement Learning* (RL), khususnya model *Q-Learning*.

Perancangan sistem bertujuan untuk menggambarkan secara rinci struktur internal sistem, alur data, serta komponen-komponen utama yang terlibat sebelum

diimplementasikan dalam bentuk aplikasi nyata. Dalam merancang sistem ini, digunakan pendekatan *Structured Analysis and Design* (SA/SD) yang memisahkan aspek analisis fungsional sistem dari perancangan teknis. Pemisahan ini penting untuk memastikan bahwa setiap tahap pengembangan dilakukan secara bertahap, sistematis, dan terdokumentasi dengan baik.

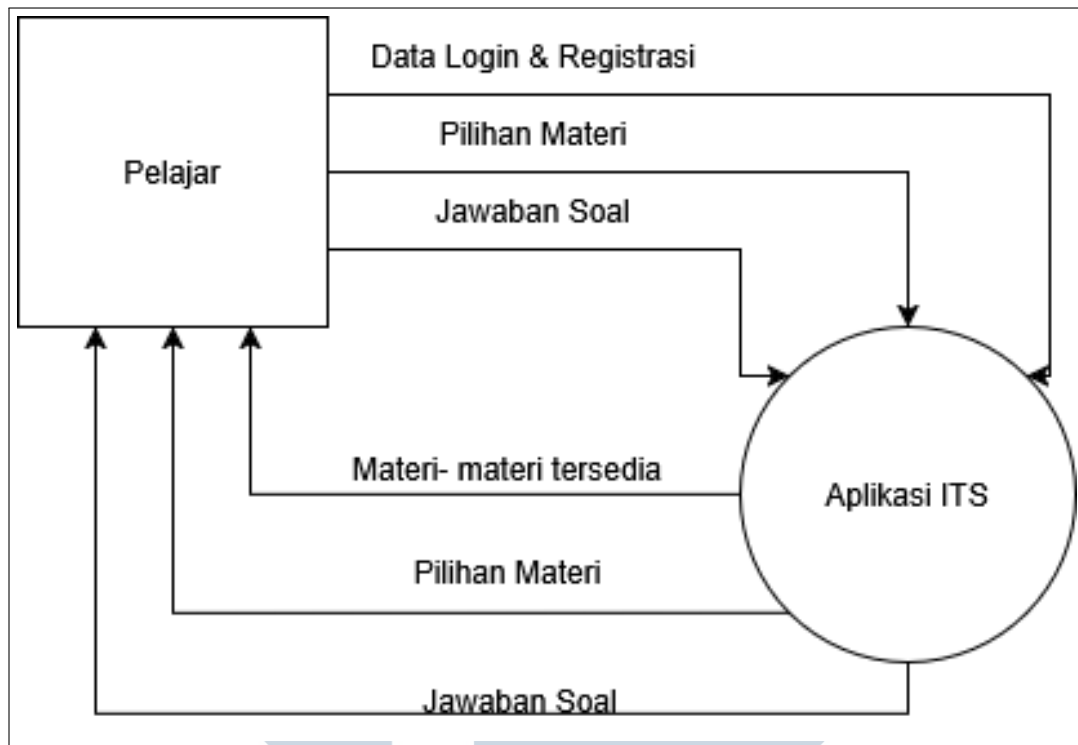
Pendekatan SA/SD dianggap paling sesuai untuk sistem yang bersifat prosedural dan terdiri dari rangkaian proses yang saling terkait, seperti proses login, registrasi, pengambilan soal, pengumpulan jawaban, dan pemberian rekomendasi tingkat kesulitan soal. Fokus utama sistem ini adalah pada alur data dan proses-proses pembelajaran adaptif berdasarkan hasil interaksi pengguna, sehingga pendekatan berbasis objek seperti *Object-Oriented Design* (OOD) kurang relevan untuk konteks pengembangan sistem ini.

Structured Analysis digunakan untuk memodelkan sistem dalam bentuk hirarki proses dan aliran data menggunakan *Data Flow Diagram* (DFD). Analisis ini menjelaskan bagaimana data bergerak melalui sistem, dimulai dari masukan pengguna hingga keluaran yang dihasilkan oleh sistem. Sementara itu, *Structured Design* diterapkan untuk menerjemahkan hasil analisis menjadi rancangan teknis sistem, melalui penyusunan struktur basis data dalam bentuk *Entity Relationship Diagram* (ERD), serta diagram *flowchart* yang menjelaskan logika sistem secara prosedural.

Perancangan sistem ini juga mencakup rancangan antarmuka pengguna yang dirancang dengan prinsip minimalis dan fungsional untuk mendukung pengalaman belajar yang personal, intuitif, dan terarah. Setiap halaman dalam aplikasi web dirancang untuk merepresentasikan proses pembelajaran berbasis sistem adaptif, mulai dari autentikasi pengguna, pengambilan soal, hingga visualisasi hasil dan rekomendasi lanjutan yang diberikan oleh algoritma RL.

3.5.1 Data Flow Diagram (DFD)

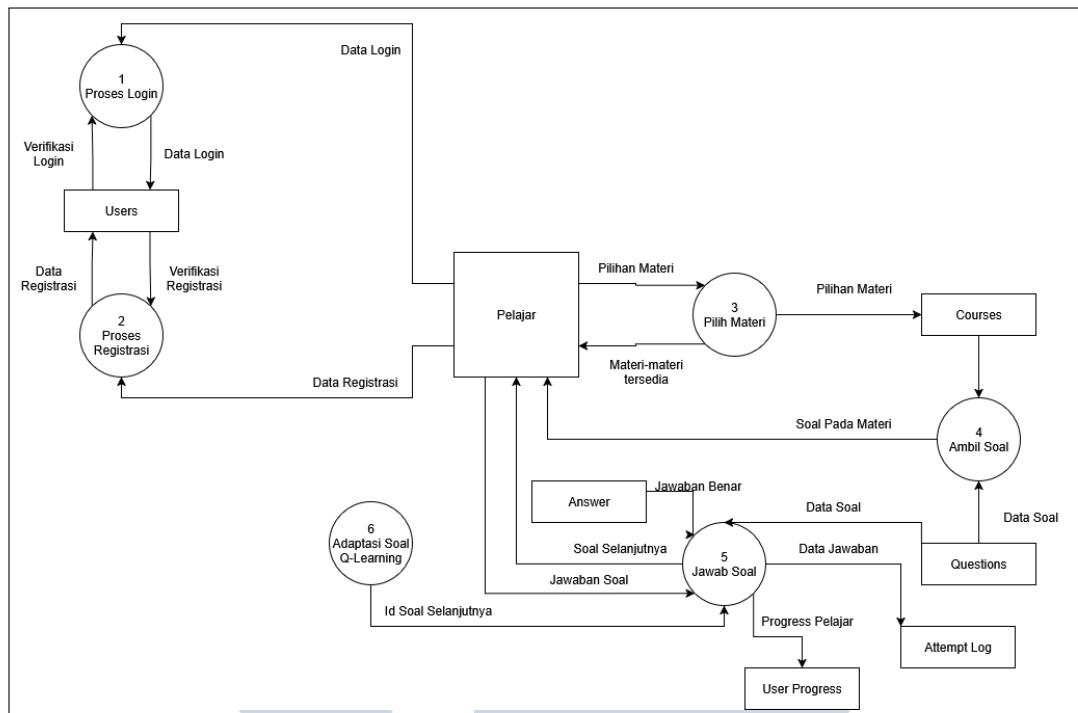
Proses *Structured Design* dalam pengembangan sistem dilakukan dengan merancang *Data Flow Diagram* (DFD). DFD digunakan untuk menggambarkan aliran data yang terjadi antara pengguna dan sistem, serta antar proses internal dalam sistem. Terdapat dua level pada diagram ini, yaitu DFD Level 0 dan DFD Level 1.



Gambar 3.3. Data Flow Diagram Level 0 ITS

Pada Gambar 3.3 *Data Flow Diagram Level 0* berfungsi sebagai representasi umum dari sistem secara keseluruhan, serta interaksinya dengan entitas eksternal. Dalam konteks penelitian ini, hanya terdapat satu entitas eksternal yang terlibat, yaitu pelajar. Gambar 3.3 menampilkan DFD Level 0 dari sistem *Intelligent Tutoring System* berbasis *Reinforcement Learning*. Pada diagram tersebut, digambarkan interaksi antara pelajar dan sistem melalui proses pengiriman dan penerimaan data. Pelajar dapat mengirimkan data login atau registrasi, memilih materi pembelajaran, serta mengerjakan soal yang disediakan. Sebagai tanggapan, sistem akan memberikan daftar materi yang tersedia, soal-soal sesuai materi, dan soal berikutnya yang telah dipersonalisasi berdasarkan kinerja pelajar melalui algoritma *Reinforcement Learning*.

Selain menggambarkan aliran data utama antara pelajar dan sistem, diagram ini juga menunjukkan proses inti yang berlangsung di dalam sistem ITS, seperti autentikasi pengguna, pengelolaan materi dan soal, serta mekanisme personalisasi soal berbasis performa. Data interaksi pelajar, seperti hasil pengerjaan soal dan tingkat akurasi, akan diproses oleh modul *Reinforcement Learning*, yang kemudian menentukan tingkat kesulitan soal selanjutnya.



Gambar 3.4. Data Flow Diagram Level 1 ITS

Pada Gambar 3.4 *Data Flow Diagram Level 1* menjabarkan sistem menjadi beberapa proses utama yang terhubung melalui aliran data. Gambar 3.4 menyajikan DFD Level 1 dari sistem ITS yang diimplementasikan. Diagram ini menunjukkan rincian proses-proses internal, seperti otentikasi pengguna, pemilihan materi, pengambilan soal, pemrosesan jawaban, hingga proses adaptasi tingkat kesulitan soal berdasarkan pendekatan *Q-Learning*. Setiap proses terhubung dengan komponen *data store* yang berfungsi menyimpan informasi penting dalam sistem.

Adapun proses-proses utama yang terdapat dalam sistem berdasarkan DFD Level 1 antara lain:

- Proses autentikasi pengguna, yang mencakup login dan registrasi akun.
- Pemilihan materi oleh pelajar.
- Pengambilan soal berdasarkan materi yang dipilih.
- Pemrosesan jawaban dan penilaian performa.
- Adaptasi tingkat kesulitan soal dengan menggunakan algoritma *Reinforcement Learning*.

Selain itu, terdapat beberapa *data store* utama yang digunakan dalam sistem, yaitu:

- Users, untuk menyimpan informasi akun pelajar.
- Courses, untuk menyimpan daftar materi yang tersedia.
- Questions, untuk menyimpan data soal-soal matematika.
- Answers, untuk menyimpan kunci jawaban soal.
- AttemptLog, untuk menyimpan riwayat pengerjaan soal.
- UserProgress, untuk menyimpan progres dan performa pelajar.

Terdapat empat jalur utama alur data yang tergambar pada DFD Level 1, yaitu:

1. Alur Login:

Pelajar mengirimkan data login berupa email dan kata sandi ke sistem. Proses login mencocokkan data tersebut dengan data pada *data store* Users. Hasil verifikasi dikirim kembali dan pelajar diarahkan ke dashboard.

2. Alur Registrasi:

Pelajar mengirimkan data registrasi berupa identitas dan kredensial akun. Sistem menyimpan data ini ke dalam *Users* dan mengirimkan hasil verifikasi ke pelajar.

3. Alur Pemilihan Materi:

Pelajar memilih materi yang tersedia. Sistem memproses pilihan tersebut dengan mengambil data dari *Courses*, kemudian mengambil soal-soal relevan dari *Questions*, dan menyajikannya kepada pelajar.

4. Alur Pengerjaan Soal dan Adaptasi:

Pelajar menjawab soal, Sistem memverifikasi jawaban dengan membandingkan data di *Answers*, lalu menyimpan hasilnya ke dalam *AttemptLog*. Sistem kemudian mengevaluasi progres pelajar dan mengirim data tersebut ke proses *Adaptasi Soal* yang mengimplementasikan algoritma Q-Learning. Berdasarkan evaluasi ini, sistem menentukan soal lanjutan yang sesuai dengan kemampuan pelajar, dan memperbarui *UserProgress*. Penerapan algoritma *Reinforcement Learning* pada sistem ini menjadi kunci

utama dalam memberikan pengalaman belajar yang dipersonalisasi. Melalui proses adaptasi otomatis, sistem ITS mampu menyesuaikan tingkat kesulitan soal dengan performa aktual pelajar, sehingga dapat meningkatkan efektivitas pembelajaran matematika secara signifikan.

3.5.2 Struktur Database

Database dirancang untuk menyimpan berbagai data penting yang dibutuhkan oleh sistem *Intelligent Tutoring System* (ITS) berbasis *Reinforcement Learning*, seperti data pelajar, data soal, data keterampilan (*skills*), jawaban siswa, serta log aktivitas interaksi siswa. Basis data ini juga mendukung penyimpanan hasil personalisasi tingkat kesulitan soal berdasarkan hasil prediksi model *Q-Learning*.

Pembangunan database dilakukan berdasarkan *Entity Relationship Diagram* (ERD) yang ditampilkan pada Gambar 3.5. Dalam implementasinya, sistem menggunakan *SQLite* sebagai sistem manajemen basis data, dan *SQLAlchemy* sebagai *Object-Relational Mapping* (ORM) yang menghubungkan struktur database dengan kode Python di dalam aplikasi *Flask*. *SQLAlchemy* memungkinkan pengembang untuk melakukan operasi CRUD (Create, Read, Update, Delete) tanpa menulis query SQL secara eksplisit, karena tabel-tabel dimodelkan dalam bentuk objek Python.

Tabel-tabel yang digunakan pada sistem ITS untuk mendukung algoritma *Reinforcement Learning* antara lain adalah **users**, **questions**, **answers**, **user_progress**, **attempt_log**, dan **skill_mastery**. Tabel-tabel tersebut saling berelasi dan digunakan dalam proses adaptasi soal sesuai performa masing-masing pelajar.

A Tabel users

Tabel 3.1 ini menyimpan informasi pelajar yang mendaftar ke dalam sistem, seperti nama, email, dan sandi akun. Selain itu, tabel 3.1 ini juga menyimpan status level terkini dan total skor pelajar yang terus diperbarui berdasarkan hasil pengerjaan soal.

Tabel 3.1. Desain tabel `users`

Kolom	Tipe Data	Keterangan
<code>id</code>	CHAR(36)	Primary Key (PK) . ID unik pengguna yang di-generate secara otomatis dan digunakan sebagai referensi (foreign key) di tabel lain seperti <code>quiz_session</code> , <code>user_progress</code> , dan <code>user_answered_question</code> .
<code>name</code>	NVARCHAR(100)	Nama lengkap pelajar pengguna sistem.
<code>email</code>	NVARCHAR(100)	Alamat email untuk login dan identifikasi pengguna. Unik untuk setiap pengguna.
<code>password</code>	NVARCHAR(50)	Kata sandi pengguna, disimpan dalam bentuk hash untuk keamanan.
<code>current_level</code>	VARCHAR(100)	Level soal terakhir yang sedang dikerjakan pelajar (easy/medium/hard).
<code>total_score</code>	INT	Akumulasi skor dari semua jawaban soal yang telah dikerjakan.
<code>q_learning_state</code>	TEXT	State pembelajaran dalam bentuk serialisasi, digunakan oleh algoritma Q-Learning untuk personalisasi.
<code>createdAt</code>	DATETIME(7)	Tanggal dan waktu akun dibuat.
<code>updatedAt</code>	DATETIME(7)	Tanggal dan waktu terakhir akun diperbarui.

B Tabel `courses`

Tabel 3.2 ini menyimpan informasi mengenai daftar materi atau topik yang tersedia dalam sistem ITS. Setiap entri merepresentasikan satu topik pembelajaran seperti Algebra, Geometry, atau Statistics, yang nantinya akan digunakan untuk mengelompokkan soal-soal sesuai kategori materi.

Tabel ini berfungsi sebagai referensi utama dalam pengelompokan soal berdasarkan bidang keilmuan matematika. Relasi antara tabel `courses` dengan tabel lainnya, seperti `questions` dan `quiz_sessions`, memungkinkan sistem untuk mengatur distribusi soal sesuai topik yang dipilih siswa di halaman dasbor. Setiap topik memiliki `id`, `name`, dan deskripsi yang bersifat unik, sehingga memudahkan proses integrasi data serta pengambilan keputusan oleh agen *Reinforcement Learning* dalam menyesuaikan materi pembelajaran berdasarkan performa siswa.

Tabel ini juga memungkinkan pengembangan topik baru secara modular di masa mendatang.

Tabel 3.2. Desain tabel `courses`

Kolom	Tipe Data	Keterangan
id	CHAR(36)	Primary Key (PK) . ID unik untuk setiap topik atau materi pembelajaran.
name	NVARCHAR(100)	Nama topik atau judul materi pembelajaran (misalnya: Algebra, Geometry, Statistics).
createdAt	DATETIME(7)	Tanggal dan waktu materi dibuat.
updatedAt	DATETIME(7)	Tanggal dan waktu materi diperbarui.

C Tabel questions

Tabel 3.7 ini digunakan untuk menyimpan seluruh bank soal yang tersedia dalam sistem. Setiap soal terhubung dengan topik tertentu dari tabel 3.7, memiliki tingkat kesulitan (easy, medium, hard), dan ditandai dengan keterampilan atau kompetensi matematika yang diuji. Tabel 3.7 ini juga menyimpan pilihan jawaban dan jawaban benar dari tiap soal.

Tabel 3.3. Desain tabel `questions`

Kolom	Tipe Data	Keterangan
id	CHAR(36)	Primary Key (PK) . ID unik untuk setiap soal.
course_id	CHAR(36)	Foreign Key (FK) yang mengacu pada <code>courses.id</code> , menandakan soal milik topik tertentu.
question_text	TEXT	Isi atau deskripsi dari soal matematika.
options	TEXT	Pilihan jawaban dalam format JSON atau teks serialisasi.
correct_answer	NVARCHAR(100)	Jawaban benar dari soal.
difficulty	VARCHAR(50)	Tingkat kesulitan soal (easy/medium/hard).
skill_builder	NVARCHAR(100)	Label keterampilan atau kompetensi matematika terkait (misalnya: operasi aljabar, probabilitas, dll).
createdAt	DATETIME(7)	Tanggal dibuatnya soal.
updatedAt	DATETIME(7)	Tanggal terakhir soal diperbarui.

D Tabel answers

Tabel 3.4 ini mencatat jawaban pelajar terhadap setiap soal yang telah dikerjakan. Data ini meliputi ID soal, jawaban yang dipilih, status benar atau salah, serta waktu yang dihabiskan untuk menjawab. Informasi ini sangat penting untuk mengevaluasi performa pelajar dan digunakan sebagai umpan balik ke dalam sistem pembelajaran adaptif.

Tabel 3.4. Desain tabel answers

Kolom	Tipe Data	Keterangan
id	CHAR(36)	Primary Key (PK) . ID unik untuk setiap data jawaban.
user_id	CHAR(36)	Foreign Key (FK) yang mengacu ke <code>users.id</code> , menunjukkan siapa yang menjawab soal.
question_id	CHAR(36)	Foreign Key (FK) ke <code>questions.id</code> . Menyatakan soal yang dijawab.
answer	NVARCHAR(100)	Jawaban yang dipilih oleh pengguna.
is_correct	BOOLEAN	Apakah jawaban yang diberikan benar atau tidak.
response_time	FLOAT	Waktu (dalam detik) yang dihabiskan untuk menjawab soal.
createdAt	DATETIME(7)	Waktu pengiriman jawaban.

E Tabel user_progress

Tabel 3.5 ini mencatat perkembangan belajar setiap pelajar dalam tiap topik. Informasi yang disimpan meliputi level terakhir yang dicapai, jumlah soal yang telah dikerjakan, serta akurasi rata-rata jawaban. Tabel 3.5 ini menjadi sumber data utama untuk memantau personalisasi proses belajar pelajar.

Tabel ini berperan penting dalam mendukung fungsi adaptif dari sistem ITS, terutama dalam penentuan tingkat kesulitan soal berikutnya yang akan diberikan oleh agen *Q-Learning*. Dengan menyimpan informasi seperti `current_level`, `accuracy`, dan `total_attempts`, sistem dapat melakukan evaluasi performa secara kuantitatif dan menentukan apakah seorang pelajar layak naik ke level soal yang lebih tinggi, tetap di level saat ini, atau perlu diturunkan. Data dalam

tabel ini diperbarui secara otomatis setelah pelajar menyelesaikan satu set soal, sehingga memungkinkan pembelajaran yang bersifat dinamis dan berkelanjutan sesuai dengan kemampuan masing-masing individu.

Tabel 3.5. Desain tabel `user_progress`

Kolom	Tipe Data	Keterangan
<code>id</code>	CHAR(36)	Primary Key (PK). ID unik untuk setiap catatan progress.
<code>user_id</code>	CHAR(36)	Foreign Key (FK) mengacu pada <code>users.id</code> .
<code>skill_id</code>	CHAR(36)	Foreign Key (FK) ke <code>courses.id</code> , menunjukkan progress per topik.
<code>current_level</code>	VARCHAR(50)	Level saat ini (easy/medium/hard) untuk topik tersebut.
<code>completed_questions</code>	INT	Jumlah soal yang sudah diselesaikan.
<code>accuracy</code>	FLOAT	Akurasi rata-rata jawaban.
<code>updatedAt</code>	DATETIME(7)	Tanggal terakhir progress diperbarui.

F Tabel `attempt_log`

Tabel 3.6 ini menyimpan riwayat setiap percobaan menjawab soal oleh pelajar. Setiap entri mencakup informasi pelajar, ID soal, hasil benar/salah, serta waktu menjawab. Log ini digunakan sebagai dasar data historis oleh algoritma Reinforcement Learning dalam menyesuaikan soal berikutnya.

Tabel 3.6. Desain tabel `attempt_log`

Kolom	Tipe Data	Keterangan
<code>id</code>	CHAR(36)	Primary Key (PK). ID unik untuk setiap catatan percobaan menjawab.
<code>user_id</code>	CHAR(36)	Foreign Key (FK) mengacu pada <code>users.id</code> .
<code>question_id</code>	CHAR(36)	Foreign Key (FK) ke <code>questions.id</code> .
<code>skill_id</code>	CHAR(36)	Foreign Key (FK) ke <code>courses.id</code> .
<code>is_correct</code>	BOOLEAN	Status benar atau salah.
<code>answered_at</code>	DATETIME(7)	Waktu saat menjawab.

G Tabel questions

Tabel 3.7 ini menyimpan nilai penguasaan keterampilan *question* tiap pelajar untuk setiap kompetensi matematika yang diuji. Nilai penguasaan ini dihitung berdasarkan riwayat menjawab soal dan digunakan untuk memberikan umpan balik personal dalam sistem, sekaligus menjadi metrik keberhasilan pembelajaran adaptif.

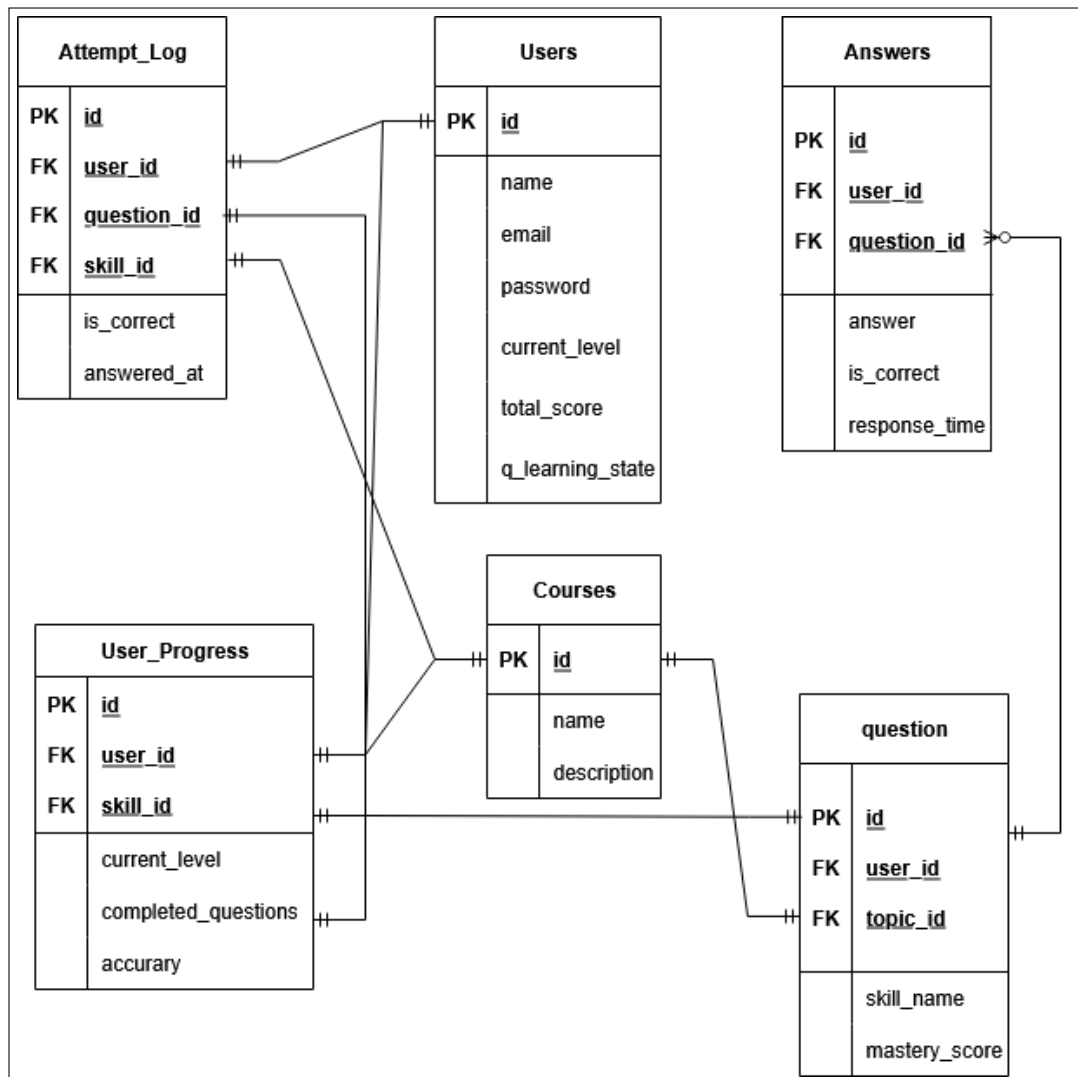
Tabel 3.7. Desain tabel questions

Kolom	Tipe Data	Keterangan
id	CHAR(36)	Primary Key (PK). ID unik untuk tiap penguasaan keterampilan.
user_id	CHAR(36)	Foreign Key (FK) ke <code>users.id</code> .
skill_name	VARCHAR(50)	Nama keterampilan matematika (misalnya: faktorisasi, probabilitas).
skill_id	CHAR(36)	Foreign Key (FK) ke <code>courses.id</code> .
mastery_score	FLOAT	Skor penguasaan yang dihitung dari performa siswa.
updatedAt	DATETIME(7)	Waktu terakhir skor diperbarui.

3.5.3 Entity Relationship Diagram (ERD)

Penyimpanan data yang ditunjukkan pada DFD Level 1 dimodelkan dalam sebuah *Entity Relationship Diagram* (ERD). ERD digunakan untuk memodelkan struktur basis data yang mendukung sistem *Intelligent Tutoring System* (ITS) berbasis *Reinforcement Learning*. Diagram ini menggambarkan entitas (tabel-tabel data) yang digunakan dalam sistem, atribut dari masing-masing entitas, serta relasi antar tabel-tabel tersebut. ERD berperan penting dalam perancangan basis data yang digunakan oleh sistem untuk menyimpan seluruh aktivitas, jawaban, dan perkembangan pelajar, serta mendukung personalisasi tingkat kesulitan soal berdasarkan pembelajaran mesin.

Melalui ERD ini, sistem dapat memastikan integritas dan konsistensi data antar proses yang berjalan, seperti pencatatan progres pelajar, manajemen soal berdasarkan topik dan tingkat kesulitan, serta penyimpanan hasil evaluasi dari algoritma *Reinforcement Learning*. Entitas-entitas seperti *Users*, *Courses*, *Questions*, dan *UserProgress* membentuk struktur relasional yang memungkinkan sistem melakukan pelacakan pembelajaran secara individual.



Gambar 3.5. Entity Relationship Diagram database ITS

Entity Relationship Diagram (ERD) dari sistem ini ditampilkan pada Gambar 3.5. Berdasarkan diagram tersebut, terdapat tujuh entitas utama yang dibangun pada sistem ITS, yaitu Users, Courses, Questions, Answers, Attempt_Log, User_Progress, dan Skill_Mastery. Tabel-tabel tersebut saling terhubung dan berkontribusi langsung dalam mekanisme adaptasi dan rekomendasi soal berdasarkan algoritma *Q-Learning*.

Relasi antar entitas dijelaskan sebagai berikut:

- *Users* → *Attempt_Log* (*one-to-many*): satu pelajar dapat memiliki banyak catatan riwayat menjawab soal. Hal ini memungkinkan sistem memonitor dan merekam semua interaksi pelajar dalam menjawab pertanyaan sebagai data historis bagi algoritma RL.

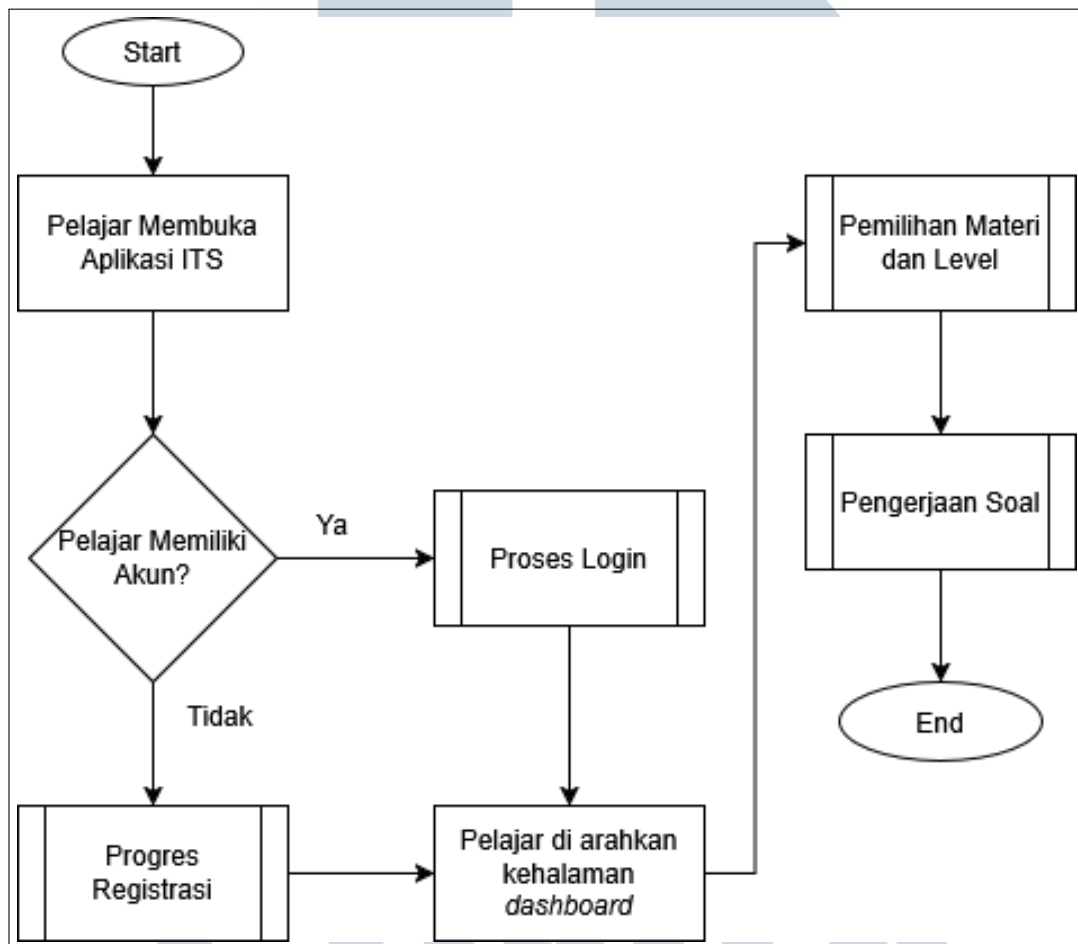
- *Users* → *User_Progress* (*one-to-many*): satu pelajar dapat memiliki banyak progres belajar pada berbagai topik yang tersedia. Informasi ini digunakan untuk menentukan status kemajuan dan pelacakan level kesulitan.
- *Users* → *Skill_Mastery* (*one-to-many*): satu pelajar memiliki banyak skor penguasaan keterampilan (*skill*) berdasarkan performa mereka di berbagai materi. Ini sangat penting dalam personalisasi jalur pembelajaran.
- *Courses* → *Questions* (*one-to-many*): setiap materi atau topik (*Courses*) memiliki sejumlah soal (*Questions*) yang sesuai dengan silabus atau keterampilan tertentu.
- *Courses* → *User_Progress* (*one-to-many*): satu topik dapat memiliki progres belajar dari banyak pelajar.
- *Courses* → *Attempt_Log* (*one-to-many*): relasi ini merepresentasikan pelajar yang menjawab soal dari topik tertentu dan terekam dalam log upaya menjawab.
- *Questions* → *Answers* (*one-to-many*): satu soal dapat memiliki lebih dari satu kemungkinan jawaban, mendukung fleksibilitas sistem soal pilihan ganda.
- *Questions* → *Attempt_Log* (*one-to-many*): satu soal dapat dijawab oleh banyak pelajar di waktu berbeda, dan tiap upaya dicatat dalam *Attempt_Log*.
- *Questions* → *Skill_Mastery* (*many-to-one*): tiap soal diasosiasikan dengan keterampilan spesifik (*skill_name*), yang digunakan untuk menghitung tingkat penguasaan pelajar terhadap keterampilan tersebut.

Dengan struktur ERD ini, sistem dapat secara efisien mengelola data pelajar dan menghasilkan adaptasi soal yang optimal berdasarkan umpan balik performa aktual. Pendekatan ini memastikan bahwa ITS dapat berfungsi sebagai sistem pembelajaran personal adaptif yang terintegrasi dengan mekanisme pembelajaran mesin.

3.5.4 Flowchart System

Flowchart sistem digunakan untuk menggambarkan alur proses utama yang terjadi di dalam aplikasi *Intelligent Tutoring System* (ITS), mulai dari autentikasi pelajar hingga pemrosesan jawaban dan adaptasi tingkat kesulitan soal berdasarkan

algoritma *Reinforcement Learning*. Flowchart ini memberikan gambaran visual terhadap logika sistem dan mendukung pemahaman dalam implementasi perangkat lunak. Perancangan flowchart disusun mengacu pada DFD Level 1 yang ditunjukkan pada Gambar 3.4.



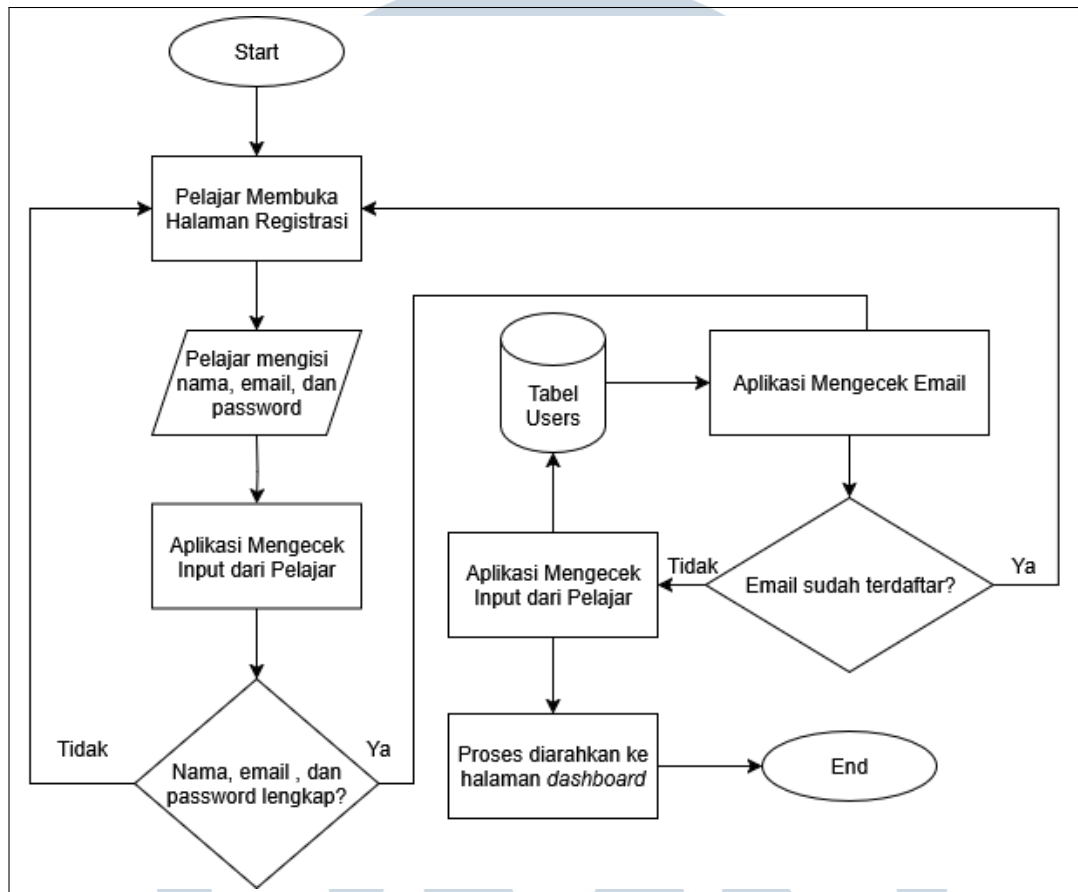
Gambar 3.6. Flowchart Sistem Aplikasi ITS

Gambar 3.6 menunjukkan flowchart yang dibuat untuk menggambarkan alur kerja sistem aplikasi ITS. Alur sistem dapat dibagi menjadi beberapa bagian utama, yaitu:

A Flowchart Autentikasi

Autentikasi adalah verifikasi identitas pengguna sebelum diberikan akses ke sistem, layanan, atau informasi tertentu. Pada penelitian ini, proses autentikasi dilakukan untuk memastikan bahwa hanya pelajar terdaftar yang dapat mengakses

aplikasi ITS. Proses autentikasi pada aplikasi ITS terbagi menjadi dua proses, yaitu proses login dan proses registrasi.

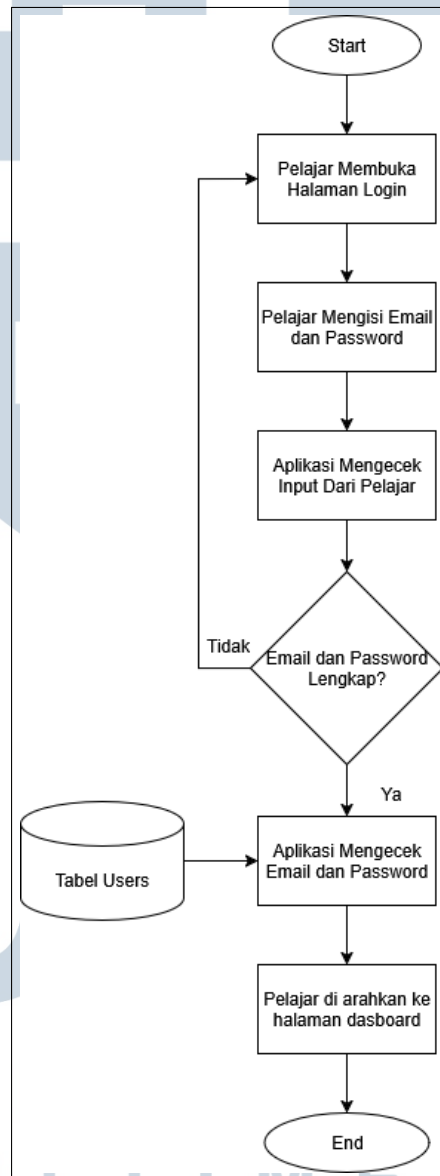


Gambar 3.7. Flowchart Proses Registrasi Aplikasi ITS

Flowchart dari proses registrasi dapat dilihat pada Gambar 3.7. Apabila pelajar belum memiliki akun pada aplikasi *Intelligent Tutoring System* (ITS), maka pelajar dapat membuka halaman registrasi untuk mendaftarkan akun baru. Pada halaman ini, pelajar diminta mengisi nama lengkap, alamat email, dan kata sandi pada kolom input yang disediakan. Setelah data diisi, aplikasi akan melakukan validasi kelengkapan data dan melakukan pengecekan ke tabel *Users* untuk memastikan bahwa email yang dimasukkan belum pernah digunakan untuk mendaftarkan akun sebelumnya.

Proses ini bertujuan untuk mencegah terjadinya duplikasi akun berdasarkan alamat email. Jika seluruh data valid dan email belum terdaftar, maka sistem akan menyimpan data akun baru ke dalam basis data dan pelajar secara otomatis akan diarahkan ke halaman *dashboard*. Selanjutnya, pelajar dapat mengakses

fitur pembelajaran dan mulai menjawab soal-soal adaptif yang disesuaikan dengan kemampuannya oleh algoritma *Reinforcement Learning*. Namun, apabila data yang dimasukkan tidak lengkap atau email telah digunakan, maka aplikasi akan menampilkan pesan kesalahan dan meminta pelajar untuk mengulangi proses registrasi.



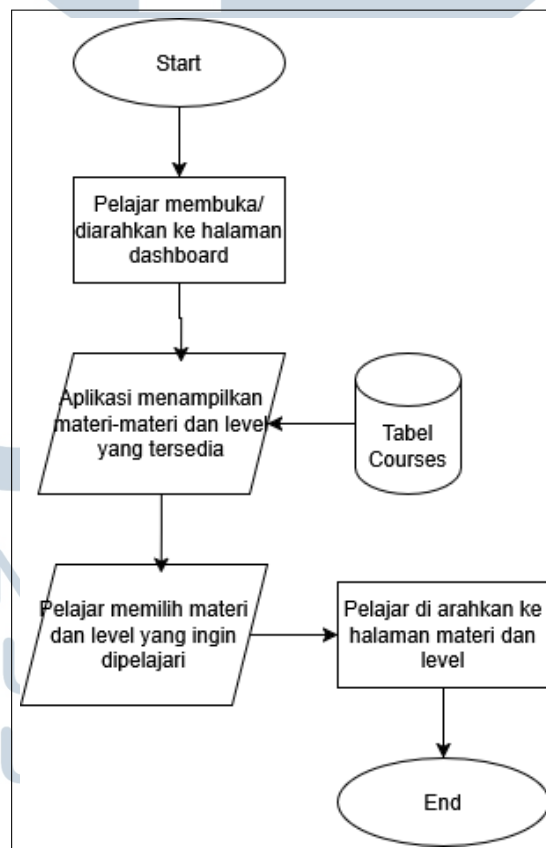
Gambar 3.8. Flowchart Proses Login Aplikasi ITS

Flowchart dari proses login ditampilkan pada Gambar 3.8. Proses ini terjadi ketika pelajar membuka halaman utama aplikasi ITS. Pada halaman login, pelajar dapat memasukkan kombinasi email dan password yang telah terdaftar pada sistem. Aplikasi akan memverifikasi input dengan mencocokkan data pada tabel Users.

Jika kombinasi email dan password sesuai, maka proses autentikasi berhasil dan pelajar diarahkan ke halaman *dashboard* untuk melanjutkan proses belajar. Pada halaman ini, pelajar dapat memilih materi, mengerjakan soal, dan melihat perkembangan hasil belajarnya yang dianalisis secara otomatis oleh algoritma *Q-Learning* sebagai bagian dari sistem *Reinforcement Learning* yang diterapkan. Jika terjadi kesalahan dalam memasukkan email atau password, maka pelajar akan diminta untuk memasukkan ulang data login yang benar.

B Flowchart Pemilihan Materi

Proses pemilihan materi dimulai setelah pelajar berhasil masuk ke dalam sistem melalui proses login atau registrasi akun baru. Flowchart dari proses pemilihan materi yang terdapat pada halaman *dashboard* dapat dilihat pada Gambar 3.9. Ketika halaman *dashboard* ditampilkan, aplikasi *Intelligent Tutoring System* (ITS) akan mengambil seluruh data materi dari tabel *Courses* yang tersedia dalam basis data.



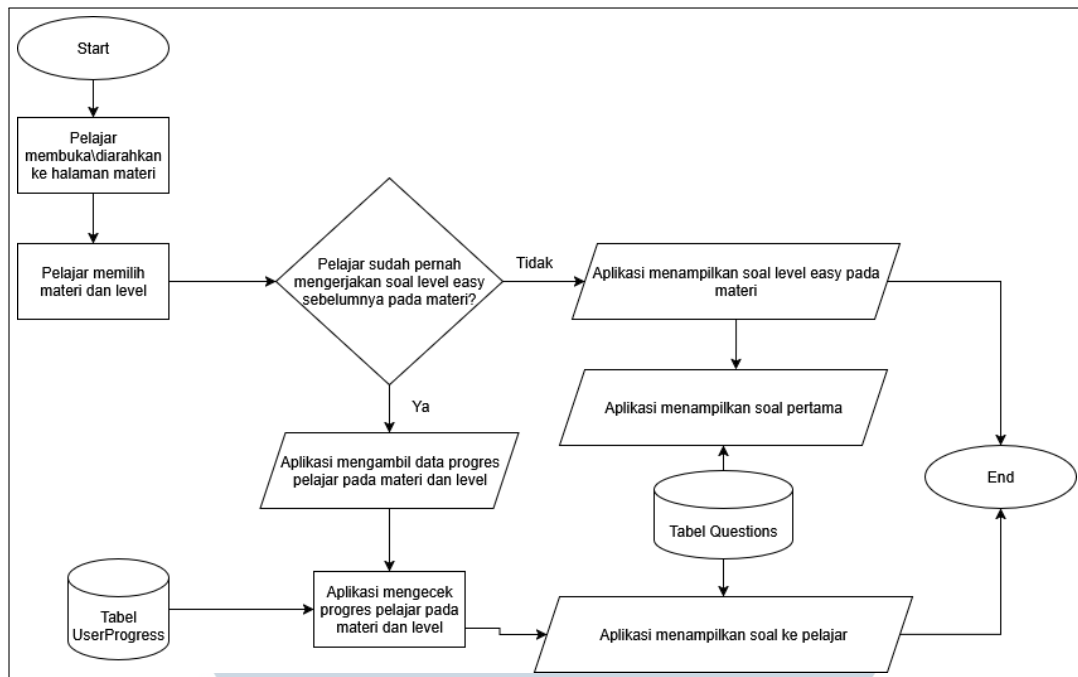
Gambar 3.9. Flowchart pemilihan materi

Data materi yang diambil kemudian ditampilkan dalam bentuk daftar topik pada halaman antarmuka pengguna. Setiap topik atau materi tersebut mencakup deskripsi singkat dan opsi untuk memulai pembelajaran. Pelajar dapat memilih salah satu dari daftar materi yang ditampilkan sesuai dengan minat atau kebutuhannya. Setelah pelajar memilih salah satu materi dari daftar, aplikasi ITS akan mencatat pilihan materi tersebut dan mengarahkan pelajar ke halaman pembelajaran topik yang bersangkutan. Di halaman ini, pelajar akan mulai menerima soal-soal yang dipersonalisasi tingkat kesulitannya berdasarkan model *Reinforcement Learning* (khususnya algoritma *Q-Learning*) yang telah terintegrasi dalam sistem. Proses pemilihan materi ini menjadi langkah awal yang penting dalam sistem adaptif karena akan menentukan konteks domain soal yang akan disesuaikan berdasarkan kemampuan individu pelajar.

C Pemilihan Soal dan Level

Setelah pelajar diarahkan ke halaman materi, aplikasi akan memproses pengambilan soal berdasarkan histori pembelajaran yang tersimpan pada sistem. Flowchart pada Gambar 3.10 menunjukkan proses pemilihan soal berdasarkan personalisasi tingkat kesulitan. Pada tahap awal, aplikasi akan memeriksa apakah pelajar telah memiliki riwayat pengerjaan soal pada materi yang dipilih melalui tabel *UserProgress*. Informasi ini digunakan untuk menyesuaikan soal yang akan ditampilkan, termasuk level soal yang terakhir dikerjakan dan akurasi jawaban sebelumnya.

Jika pelajar belum memiliki riwayat pengerjaan soal pada materi tersebut, maka sistem akan memberikan soal dengan tingkat kesulitan awal, yaitu *easy*. Namun jika riwayat sudah tersedia, maka sistem akan menggunakan data terakhir dari tabel *UserProgress* untuk menentukan tingkat kesulitan berikutnya. Logika penentuan level ini berdasarkan aturan adaptasi, seperti: akurasi $\geq 80\%$ untuk naik level, akurasi antara 51–79% untuk tetap di level yang sama, dan akurasi $< 50\%$ untuk menurunkan level ke bawah. Keputusan ini diambil oleh agen *Reinforcement Learning* melalui perhitungan *Q-value* yang dipengaruhi oleh nilai *reward* dari sesi sebelumnya. Proses ini terus berulang pada setiap soal yang dikerjakan sehingga jalur belajar setiap pelajar menjadi adaptif dan personal.

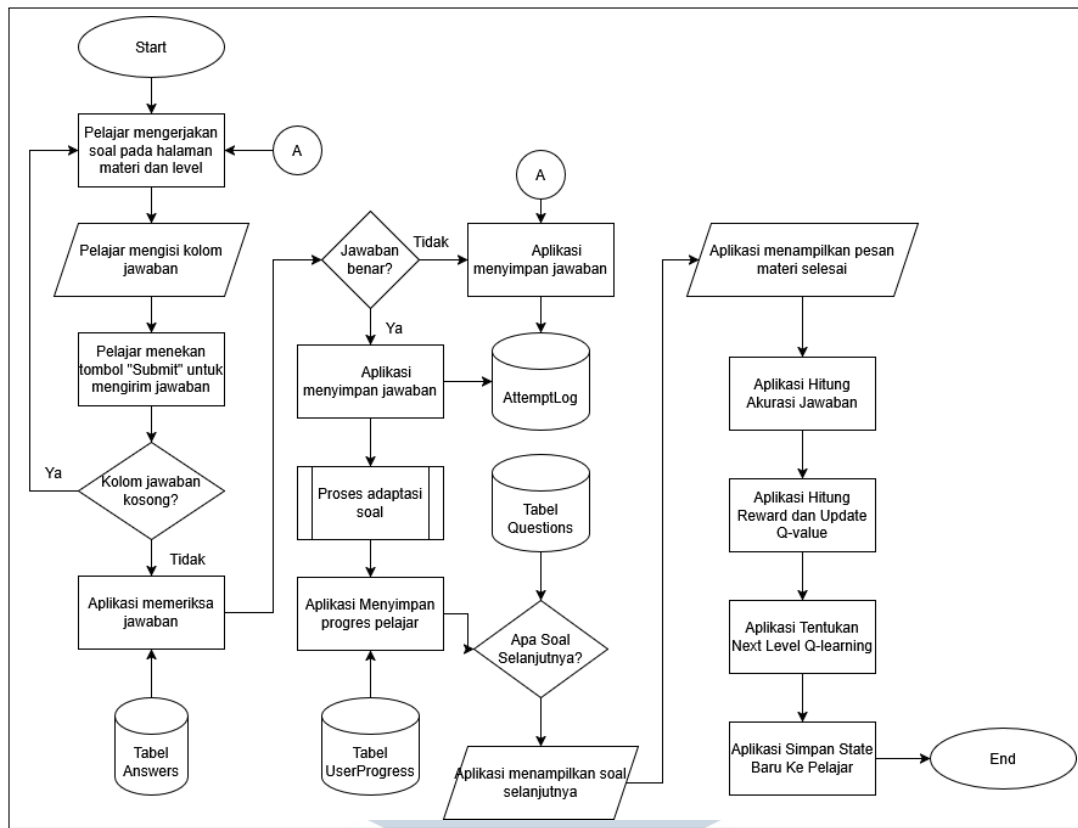


Gambar 3.10. Flowchart proses pemilihan soal pada aplikasi ITS

Jika data progress ditemukan, aplikasi menampilkan soal terakhir yang telah dijalani oleh pelajar dan melanjutkan ke soal berikutnya sesuai dengan rekomendasi sistem berbasis *Reinforcement Learning*. Jika pelajar belum pernah membuka materi tersebut sebelumnya, aplikasi akan menampilkan soal pertama pada level *easy*. Data soal diambil dari tabel *Questions* berdasarkan identifikasi topik yang dipilih pelajar.

D Menjawab Soal

Setelah soal ditampilkan, pelajar mengisi kolom jawaban pada halaman materi. Flowchart pada Gambar 3.11 menunjukkan proses pemrosesan jawaban oleh sistem. Ketika jawaban dikirim, sistem akan memverifikasi keakuratan jawaban tersebut dengan membandingkannya terhadap kunci jawaban yang terdapat pada tabel *Answers*. Setiap interaksi pelajar disimpan dalam tabel *AttemptLog* sebagai riwayat menjawab, termasuk informasi seperti pelajar, soal, topik, dan apakah jawaban benar atau salah.

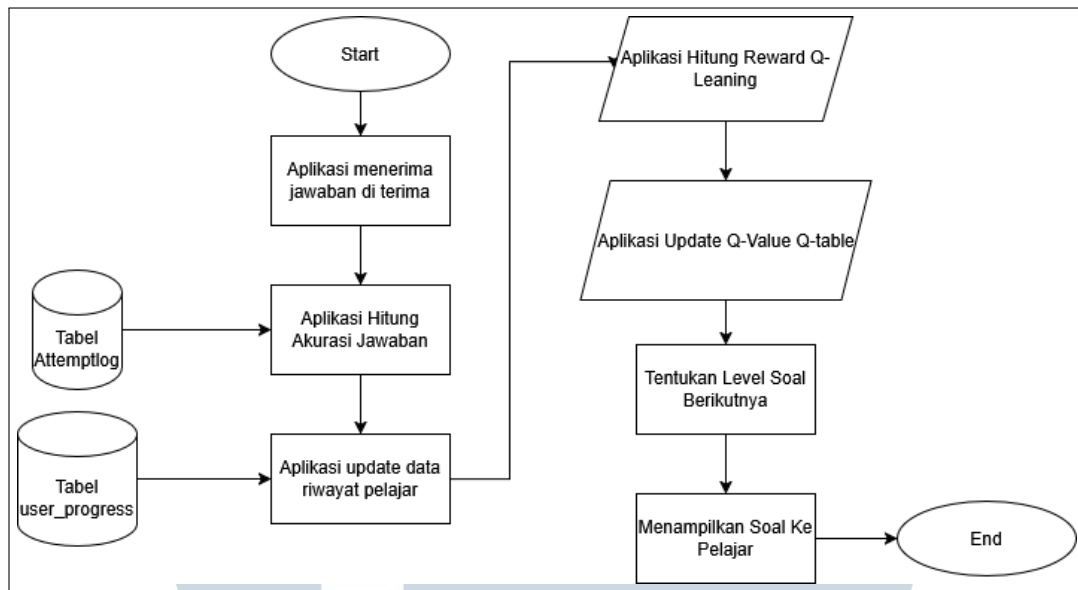


Gambar 3.11. Flowchart pemrosesan jawaban pada aplikasi ITS

Jika jawaban benar, sistem melanjutkan ke proses adaptasi tingkat kesulitan berdasarkan performa terkini pelajar. Proses ini bertujuan untuk menyesuaikan tantangan yang diberikan pada soal berikutnya. Sistem juga memperbarui kemajuan belajar pelajar ke tabel *UserProgress*. Apabila pelajar telah menyelesaikan semua soal pada materi tersebut, maka sistem akan menampilkan pesan penyelesaian kepada pelajar.

E Adaptasi Soal

Proses adaptasi tingkat kesulitan soal merupakan bagian inti dari penerapan algoritma *Reinforcement Learning* dalam sistem ini. Flowchart pada Gambar 3.12 menggambarkan proses adaptasi soal menggunakan pendekatan *Q-Learning*. Setelah jawaban diterima dan diverifikasi benar, sistem menghitung reward berdasarkan hasil jawaban pelajar dan memperbarui nilai *Q-Value* dalam *Q-Table*.



Gambar 3.12. Flowchart proses adaptasi soal pada aplikasi ITS

Sistem kemudian menentukan level soal selanjutnya (easy, medium, atau hard) berdasarkan nilai maksimum dari Q-Table untuk state pelajar saat ini. Level tersebut akan digunakan untuk menampilkan soal berikutnya dari tabel Questions, dan state terbaru pelajar juga akan diperbarui dalam sistem untuk digunakan pada sesi belajar berikutnya. Proses ini memungkinkan aplikasi ITS untuk terus beradaptasi dengan kemampuan pelajar dan memberikan jalur pembelajaran yang dipersonalisasi secara optimal.

3.5.5 Rancangan Tampilan Antarmuka

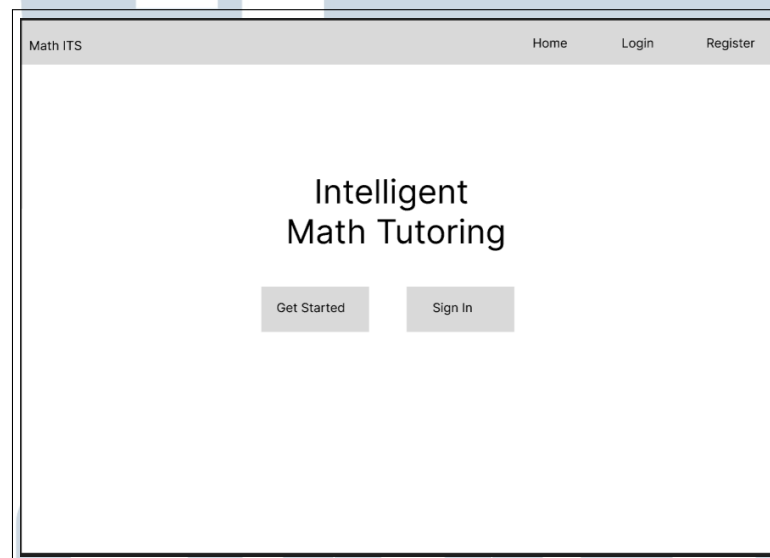
Perancangan antarmuka pengguna (*User Interface/UI*) bertujuan untuk menggambarkan bentuk akhir dari aplikasi web *Intelligent Tutoring System (ITS)* sebelum proses implementasi dilakukan. Tampilan yang dirancang meliputi beberapa halaman penting seperti halaman login, dasbor pelajar, tampilan soal, hingga halaman evaluasi hasil belajar. Seluruh desain antarmuka dikembangkan menggunakan platform desain antarmuka modern Figma.

Figma merupakan aplikasi berbasis web yang memungkinkan perancangan UI/UX secara kolaboratif dan interaktif. Dengan menggunakan Figma, rancangan tampilan dapat divisualisasikan secara realistis dan responsif, sehingga memberikan gambaran fungsional terhadap alur sistem dan pengalaman pengguna. Pendekatan ini bertujuan untuk memastikan bahwa sistem ITS tidak hanya berfungsi secara

teknis, tetapi juga memiliki desain antarmuka yang ramah pengguna dan mendukung pembelajaran adaptif berbasis algoritma *Reinforcement Learning*.

A Halaman Homepage

Halaman homepage merupakan tampilan awal yang muncul ketika pengguna pertama kali membuka aplikasi *Intelligent Tutoring System* (ITS). Wireframe dari halaman ini ditampilkan pada Gambar 3.13. Tujuan utama dari halaman ini adalah memberikan pengenalan singkat mengenai fungsi sistem ITS, serta memfasilitasi akses cepat ke proses registrasi dan login pengguna.



Gambar 3.13. Wireframe Homepage

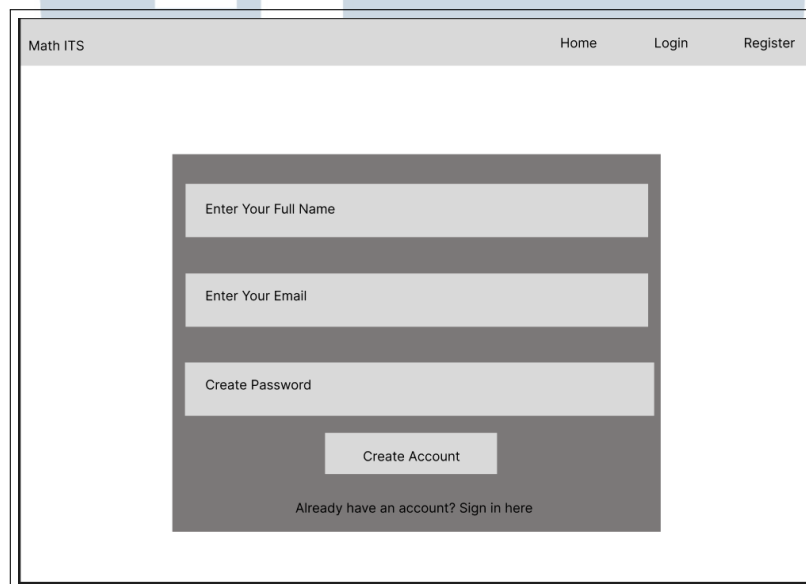
Desain halaman homepage dirancang dengan antarmuka yang minimalis dan fokus terhadap pengalaman pengguna (*user experience*). Pada bagian atas halaman, terdapat *navigation bar* dengan label “Math ITS” di sisi kiri dan menu navigasi “Home”, “Login”, serta “Register” di sisi kanan. Di tengah halaman, ditampilkan teks “Intelligent Math Tutoring” sebagai judul utama yang merepresentasikan fungsi sistem.

Di bawah judul, terdapat dua tombol utama: **Get Started** dan **Sign In**. Tombol “Get Started” mengarahkan pengguna ke proses registrasi jika belum memiliki akun, sedangkan tombol “Sign In” mengarahkan pengguna ke halaman login. Struktur sederhana dan langsung ini membantu pengguna dalam memahami langkah awal untuk menggunakan aplikasi ITS, yang nantinya akan

mengadaptasi tingkat kesulitan soal matematika berdasarkan performa pengguna melalui algoritma *Reinforcement Learning*.

B Halaman Register

Halaman register merupakan tampilan yang digunakan ketika pengguna ingin membuat akun baru untuk mengakses sistem *Intelligent Tutoring System* (ITS). Wireframe dari halaman ini ditampilkan pada Gambar 3.14. Tujuan utama dari halaman ini adalah memfasilitasi proses pendaftaran pengguna baru agar dapat masuk ke dalam sistem dan menggunakan fitur pembelajaran yang disediakan.



The wireframe shows a web page for 'Math ITS'. At the top, there is a navigation bar with the site name 'Math ITS' on the left and three links: 'Home', 'Login', and 'Register' on the right. The main body of the page features a central registration form. This form consists of three stacked text input fields with placeholder text: 'Enter Your Full Name', 'Enter Your Email', and 'Create Password'. Below the third field is a rectangular button labeled 'Create Account'. At the bottom of the form area, there is a text link that reads 'Already have an account? Sign in here'.

Gambar 3.14. Wireframe Register

Desain halaman register dirancang dengan antarmuka yang sederhana dan mudah dipahami, sehingga pengguna dapat melakukan proses registrasi dengan cepat tanpa kebingungan. Pada bagian atas halaman, terdapat *navigation bar* dengan label “Math ITS” di sisi kiri, serta menu navigasi “Home”, “Login”, dan “Register” di sisi kanan. Elemen ini memberikan akses cepat ke halaman penting lainnya.

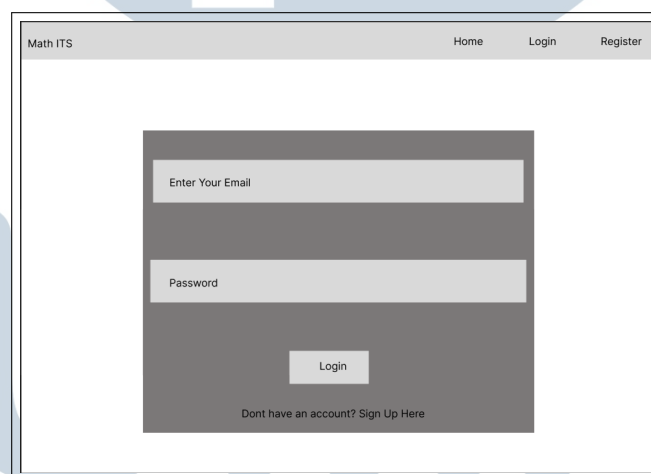
Di bagian tengah halaman, terdapat sebuah form pendaftaran dengan latar abu-abu gelap yang terdiri dari tiga kolom input, yaitu: *Enter Your Full Name*, *Enter Your Email*, dan *Create Password*. Ketiga kolom ini merupakan input wajib yang harus diisi oleh pengguna untuk membuat akun. Di bawah form, terdapat tombol utama yaitu **Create Account**, yang berfungsi untuk mengirimkan data registrasi ke

sistem. Selain itu, di bagian paling bawah, terdapat tautan teks bertuliskan "*Already have an account? Sign in here*" yang mengarahkan pengguna ke halaman login jika sudah memiliki akun sebelumnya.

Struktur halaman ini dirancang secara langsung dan efisien untuk memastikan pengguna dapat memahami alur registrasi dengan cepat. Hal ini penting dalam mendukung proses adaptif sistem ITS, yang nantinya akan mempersonalisasi tingkat kesulitan soal berdasarkan performa pengguna dengan menggunakan algoritma *Reinforcement Learning*.

C Halaman Login

Halaman login merupakan tampilan yang digunakan ketika pengguna ingin masuk ke dalam sistem *Intelligent Tutoring System* (ITS) menggunakan akun yang sudah terdaftar. Wireframe dari halaman ini ditampilkan pada Gambar 3.15. Tujuan utama dari halaman ini adalah memberikan akses autentikasi bagi pengguna agar dapat menggunakan seluruh fitur sistem ITS.



Gambar 3.15. Wireframe Login

Desain halaman login dirancang dengan tampilan yang sederhana dan fokus pada fungsionalitas utama, yaitu proses autentikasi pengguna. Pada bagian atas halaman terdapat *navigation bar* dengan label "Math ITS" di sisi kiri, serta menu navigasi "Home", "Login", dan "Register" di sisi kanan. Navigasi ini konsisten dengan halaman lain untuk memudahkan perpindahan antar halaman.

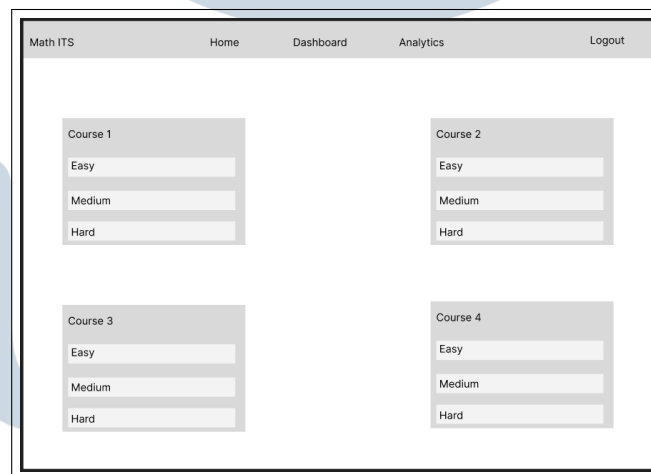
Bagian utama halaman menampilkan form *login* yang terdiri dari dua kolom input, yaitu *Enter Your Email* dan *Password*. Kolom pertama digunakan untuk

mengisi alamat email pengguna yang telah terdaftar, sedangkan kolom kedua digunakan untuk memasukkan password. Di bawah kolom input, terdapat tombol utama yaitu *Login* yang berfungsi untuk mengirimkan data autentikasi ke sistem.

Selain itu, di bagian bawah form terdapat teks tambahan yang bertuliskan "*Don't have an account? Sign Up Here*" sebagai tautan alternatif yang mengarahkan pengguna ke halaman registrasi apabila mereka belum memiliki akun. Struktur halaman ini dibuat agar pengguna dapat langsung memahami cara masuk ke dalam sistem dan melanjutkan proses pembelajaran yang dipersonalisasi berdasarkan performa mereka menggunakan algoritma *Reinforcement Learning*.

D Halaman Dashboard

Halaman dashboard merupakan tampilan utama yang muncul setelah pengguna berhasil login ke dalam sistem *Intelligent Tutoring System* (ITS). Wireframe dari halaman ini ditampilkan pada Gambar 3.16. Tujuan utama dari halaman dashboard adalah menyediakan akses langsung ke materi pembelajaran berdasarkan kategori mata pelajaran dan tingkat kesulitan.



Gambar 3.16. Wireframe Dashboard

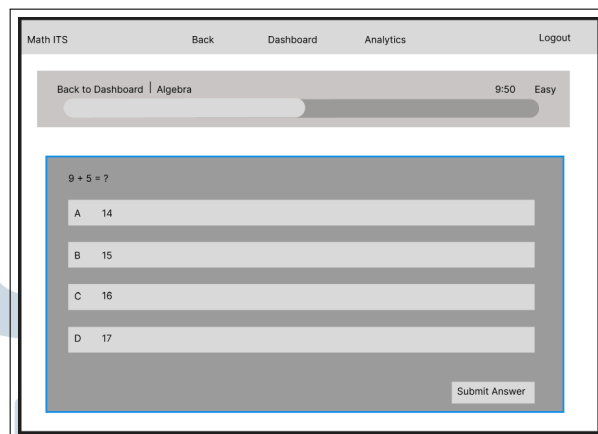
Tampilan halaman ini dirancang dengan antarmuka yang bersih dan terorganisir, agar pengguna dapat memilih materi dan tingkat kesulitan soal dengan mudah. Pada bagian atas halaman terdapat *navigation bar* yang terdiri dari label "Math ITS" di sisi kiri, dan menu navigasi "Home", "Dashboard", "Analytics", serta tombol "Logout" di sisi kanan. Navigasi ini membantu pengguna berpindah antar fitur utama sistem.

Bagian utama halaman menampilkan empat blok kursus, masing-masing diberi label *Course 1*, *Course 2*, *Course 3*, dan *Course 4*. Setiap course merepresentasikan topik atau materi pembelajaran matematika yang berbeda, seperti Algebra, Geometry, Calculus, dan Statistics. Di dalam setiap blok course, terdapat tiga tombol yang mewakili tiga tingkat kesulitan soal, yaitu Easy, Medium, dan Hard. Pengguna dapat memilih tingkat kesulitan sesuai dengan level yang telah dibuka berdasarkan performa mereka sebelumnya.

Desain halaman dashboard ini mendukung personalisasi pembelajaran berbasis Reinforcement Learning, di mana pengguna akan diarahkan secara adaptif ke tingkat kesulitan yang sesuai berdasarkan akurasi jawaban mereka di setiap level soal.

E Halaman Soal

Halaman soal merupakan tampilan yang digunakan oleh pengguna untuk menjawab pertanyaan-pertanyaan dalam sistem *Intelligent Tutoring System (ITS)*. Wireframe dari halaman ini ditampilkan pada Gambar 3.17. Tujuan utama dari halaman ini adalah untuk menampilkan soal matematika sesuai dengan mata pelajaran dan tingkat kesulitan yang dipilih oleh pengguna pada halaman dashboard.



Gambar 3.17. Wireframe Soal

Tampilan halaman ini dirancang agar fokus pada proses pengerjaan soal. Di bagian atas halaman terdapat *navigation bar* yang konsisten dengan halaman lainnya, yaitu label “Math ITS” di sisi kiri serta menu navigasi “Back”, “Dashboard”, “Analytics”, dan tombol “Logout” di sisi kanan. Menu ini memudahkan pengguna untuk kembali ke tampilan utama atau keluar dari sesi.

Di bawah navigasi, terdapat informasi breadcrumb yang menunjukkan posisi pengguna saat ini, misalnya “Back to Dashboard — Algebra”. Di sisi kanan bar tersebut ditampilkan informasi waktu pengerjaan soal (misalnya “9:50”) dan tingkat kesulitan soal yang sedang dikerjakan (misalnya “Easy”). Terdapat juga progress bar horizontal yang menunjukkan sejauh mana pengguna telah menyelesaikan soal pada level tersebut.

Bagian utama halaman menampilkan konten soal dalam kotak berwarna abu-abu. Soal ditampilkan dalam format pilihan ganda. Contoh soal yang terlihat adalah “ $9 + 5 = ?$ ”, dengan empat pilihan jawaban yang diberi label A, B, C, dan D, masing-masing dengan nilai seperti 14, 15, 16, dan 17. Di bagian bawah kotak soal, terdapat tombol Submit Answer yang digunakan untuk mengirimkan jawaban pengguna.

Tampilan halaman soal ini dirancang untuk memudahkan interaksi, fokus pada konten, serta mendukung sistem personalisasi berbasis *Reinforcement Learning* yang akan menyesuaikan tingkat kesulitan soal berikutnya berdasarkan performa pengguna.

3.6 Pengembangan *Prototype*

Tahap pengembangan *prototype* dilakukan untuk merealisasikan rancangan sistem menjadi sebuah aplikasi berbasis web yang dapat dijalankan secara fungsional. Sistem yang dikembangkan mengimplementasikan *Intelligent Tutoring System* (ITS) yang memanfaatkan algoritma *Reinforcement Learning* (RL), khususnya *Q-Learning*, untuk melakukan personalisasi tingkat kesulitan soal matematika berdasarkan performa pelajar secara dinamis.

Proses pengembangan dilakukan secara iteratif sesuai dengan pendekatan model *Prototyping*, dimulai dari perancangan awal, pembuatan fitur utama, hingga proses evaluasi dan perbaikan. *Prototype* ini terdiri dari tiga komponen utama, yaitu antarmuka pengguna (frontend), logika aplikasi dan integrasi model RL (backend), serta basis data (database) yang saling terhubung.

Pada bagian frontend, pengembangan dilakukan menggunakan HTML, CSS, dan JavaScript, dengan framework tambahan seperti React.js untuk antarmuka yang responsif dan dinamis. Sedangkan pada sisi backend, digunakan framework Flask (Python) untuk mengatur alur logika aplikasi serta berkomunikasi dengan model RL yang ditanamkan di dalam Jupyter Notebook. Model RL ini digunakan untuk menganalisis riwayat belajar pelajar dan menentukan level soal berikutnya

berdasarkan strategi *exploration-exploitation* dalam *Q-Learning*.

Database sistem dibangun menggunakan *Microsoft SQL Server*, yang diakses melalui ORM *Sequelize* untuk kemudahan dalam pengelolaan data. Struktur database dirancang agar mendukung penyimpanan informasi akun pelajar, data soal, hasil jawaban, riwayat percobaan menjawab, serta status progres pelajar. Dengan integrasi penuh antar ketiga komponen ini, *prototype* ITS yang dikembangkan mampu berfungsi sebagai platform pembelajaran adaptif yang cerdas dan responsif terhadap kebutuhan belajar masing-masing individu.

3.6.1 Penerapan Model *Reinforcement Learning* (RL)

Penerapan algoritma *Reinforcement Learning* (RL) dalam sistem *Intelligent Tutoring System* (ITS) bertujuan untuk mempersonalisasi tingkat kesulitan soal berdasarkan performa pelajar secara dinamis. Dalam penelitian ini, digunakan algoritma *Q-Learning*, yaitu salah satu metode RL yang mampu mempelajari kebijakan optimal melalui interaksi dengan lingkungan pembelajaran.

Algoritma *Q-Learning* dirancang untuk menentukan level soal selanjutnya berdasarkan state pelajar, yang terdiri dari progres belajar, tingkat akurasi, dan hasil jawaban sebelumnya. Setiap interaksi pelajar dengan soal akan membentuk *state-action-reward* triplet, yang kemudian digunakan untuk mengupdate nilai Q pada *Q-table*. *Q-table* merupakan representasi dari semua kemungkinan tindakan (aksi memilih level soal) yang dapat diambil dari suatu state tertentu.

Reward diberikan berdasarkan tingkat kebenaran jawaban pelajar. Jika pelajar menjawab dengan benar, maka algoritma memberikan reward positif dan cenderung menaikkan level soal. Sebaliknya, jika jawaban salah, reward akan negatif dan algoritma akan menyesuaikan dengan menurunkan atau mempertahankan tingkat kesulitan soal. Proses ini berulang secara iteratif sehingga sistem dapat belajar dari perilaku masing-masing pelajar.

Model *Q-Learning* diimplementasikan menggunakan Python dalam Jupyter Notebook, dan kemudian diintegrasikan ke dalam backend sistem berbasis Flask. Hasil prediksi dari model digunakan oleh aplikasi untuk menentukan level soal berikutnya secara otomatis, dengan mempertimbangkan keseimbangan antara eksplorasi (menjelajahi level baru) dan eksploitasi (memilih level yang sudah terbukti optimal). Parameter seperti *learning rate* dan *exploration rate* dikonfigurasi agar model dapat menyesuaikan dengan berbagai profil pelajar secara efektif.

Dengan penerapan model *Q-Learning* ini, sistem ITS mampu memberikan pengalaman belajar yang adaptif dan personal, meningkatkan motivasi pelajar, serta mengarahkan mereka secara bertahap ke tingkat penguasaan materi yang lebih tinggi sesuai dengan kemampuan masing-masing individu.

3.6.2 Implementasi *Frontend*

Implementasi *frontend* dilakukan untuk membangun antarmuka pengguna (User Interface/UI) dari aplikasi *Intelligent Tutoring System* (ITS) berbasis web. Komponen *frontend* bertugas untuk mengatur tampilan dan interaksi pengguna dengan sistem, mulai dari halaman login, registrasi, dasbor, hingga tampilan soal dan evaluasi hasil belajar. Pengembangan dilakukan menggunakan kombinasi teknologi HTML, CSS, dan JavaScript.

Struktur *frontend* dirancang berdasarkan prinsip kemudahan penggunaan (*usability*) dan pengalaman pengguna yang baik (*user experience/UX*). Navigasi antar halaman diatur agar intuitif, dengan tombol dan informasi yang mudah dipahami oleh pelajar. Elemen visual seperti progress bar, badge tingkat kesulitan (*easy, medium, hard*), dan waktu pengerjaan ditambahkan untuk mendukung keterlibatan pengguna selama proses belajar.

Halaman login dan registrasi dirancang agar sederhana dan cepat diakses. Setelah berhasil login, pengguna akan diarahkan ke halaman dasbor yang menampilkan daftar materi yang tersedia. Setiap materi dapat dipilih dan membawa pelajar ke halaman pengerjaan soal, di mana pelajar dapat menjawab pertanyaan dengan sistem yang menyesuaikan tingkat kesulitannya menggunakan algoritma *Reinforcement Learning*.

Antarmuka soal dikembangkan dengan desain responsif sehingga dapat diakses melalui berbagai perangkat. Setiap pertanyaan ditampilkan bersama opsi jawaban pilihan ganda, dan setelah pengguna menjawab, sistem akan menampilkan soal berikutnya sesuai tingkat kesulitan yang disesuaikan oleh model *Q-Learning*. Komunikasi antara *frontend* dan *backend* dilakukan melalui permintaan HTTP menggunakan *fetch API* atau *Axios*, agar data pelajar, jawaban, dan rekomendasi soal.

Secara keseluruhan, implementasi *frontend* tidak hanya berfungsi sebagai tampilan visual, tetapi juga sebagai penghubung penting antara pengguna dengan proses adaptasi pembelajaran yang dilakukan oleh sistem ITS.

3.6.3 Implementasi Backend

Implementasi *backend* dilakukan untuk menangani seluruh logika sistem, manajemen data, dan komunikasi antara pengguna dengan model *Reinforcement Learning* (RL). Komponen *backend* dibangun menggunakan framework Flask, yaitu framework berbasis Python yang ringan dan fleksibel untuk membangun aplikasi web.

Fungsi utama dari *backend* mencakup otentikasi pengguna, pengambilan soal, penyimpanan jawaban, pemrosesan hasil, serta integrasi dengan model *Q-Learning* untuk menentukan tingkat kesulitan soal berikutnya. Setiap permintaan dari *frontend* akan dikirimkan ke Flask API endpoint, diproses oleh sistem, dan kemudian dikembalikan sebagai respons dalam format JSON.

Struktur *backend* mencakup beberapa modul utama, yaitu:

- Auth Module: Menangani proses registrasi dan login, termasuk enkripsi password dan validasi token menggunakan Flask-JWT-Extended.
- Question Module: Menyediakan soal berdasarkan materi dan level yang sesuai, serta mencegah pengulangan soal.
- Quiz Module: Menerima jawaban pelajar, memvalidasi jawaban terhadap kunci jawaban di database, dan menyimpan hasil ke tabel `AttemptLog`.
- RL Module: Menghubungkan sistem dengan model *Q-Learning*, yang akan memproses state pelajar dan menentukan level soal selanjutnya.

Untuk pengelolaan basis data, SQLAlchemy digunakan sebagai Object-Relational Mapping (ORM), memudahkan pembuatan dan manipulasi data tanpa menulis query SQL secara langsung. Seluruh data seperti informasi pengguna, soal, jawaban, progres belajar, dan hasil evaluasi disimpan pada tabel-tabel sesuai dengan skema database yang telah dirancang.

Integrasi antara *backend* dan model RL dilakukan melalui pemanggilan fungsi dari modul `q_learning_model.py` dan `data_processor.py`. Model akan membaca state pelajar dari database, menghitung Q-value untuk setiap tindakan yang mungkin (misalnya melanjutkan ke soal level lebih sulit), dan mengembalikan level soal yang paling sesuai berdasarkan strategi *exploration-exploitation*.

Melalui implementasi ini, sistem ITS dapat menyesuaikan pengalaman belajar secara adaptif dan real-time, memastikan bahwa setiap pelajar mendapatkan soal dengan tingkat kesulitan yang optimal sesuai performa mereka.

3.6.4 Integrasi Model RL Ke Sistem

Integrasi model *Reinforcement Learning* (RL) ke dalam sistem *Intelligent Tutoring System* (ITS) bertujuan untuk menghubungkan komponen pembelajaran mesin dengan sistem backend agar proses adaptasi tingkat kesulitan soal dapat dilakukan secara otomatis dan personal. Model RL yang digunakan dalam sistem ini adalah algoritma *Q-Learning*, yang berfungsi sebagai agen pembelajaran yang menentukan aksi terbaik (tingkat soal selanjutnya) berdasarkan histori interaksi pelajar.

Proses integrasi dilakukan melalui modul `q_learning_model.py` dan `data_processor.py` pada bagian backend aplikasi. Ketika pelajar menyelesaikan suatu soal, data jawaban dan performa akan disimpan terlebih dahulu ke dalam tabel `AttemptLog`. Backend kemudian memanggil fungsi `determine_next_level()` dari agen *Q-Learning* dengan parameter berupa tingkat akurasi, level saat ini, dan state historis dari pelajar.

Model RL akan mengevaluasi nilai *Q* untuk setiap tindakan yang mungkin dilakukan oleh sistem, seperti menaikkan level, mempertahankan level, atau menurunkan level soal. Berdasarkan kombinasi nilai reward dan kebijakan eksplorasi-eksploitasi, model menentukan aksi dengan probabilitas tertinggi dan mengembalikannya ke sistem backend.

Hasil keputusan model akan digunakan untuk:

- Menentukan level soal selanjutnya yang ditampilkan kepada pelajar.
- Memperbarui state *Q* pada pelajar di database (dalam bentuk serialisasi JSON).
- Menyesuaikan alur materi berdasarkan progres dan kemampuan pelajar.

Integrasi ini memungkinkan proses pembelajaran menjadi adaptif dan personal, karena setiap pelajar mendapatkan pengalaman belajar yang unik sesuai dengan performanya. Selain itu, model *Q-Learning* juga dapat terus belajar dari data historis pelajar untuk meningkatkan akurasi prediksi dalam jangka panjang.

Secara teknis, proses integrasi dilakukan secara sinkron saat pelajar mengirimkan jawaban, sehingga keputusan adaptasi dapat diterapkan secara langsung pada soal selanjutnya. Dengan pendekatan ini, aplikasi ITS tidak hanya menyajikan konten secara statis, tetapi mampu menyesuaikan tingkat kesulitan soal berdasarkan pembelajaran yang dilakukan oleh sistem secara terus-menerus.