

## BAB 3

### METODOLOGI PENELITIAN

#### 3.1 Studi Literatur

Tahap studi literatur dilakukan dengan mencari dan mempelajari berbagai referensi serta teori yang digunakan sebagai landasan pendukung penelitian. Teori-teori ini bersumber dari referensi pustaka seperti jurnal, artikel ilmiah, dan buku yang berkaitan dengan Teori Gamifikasi (khususnya framework Octalysis), Intelligent Tutoring Systems (ITS) atau sistem pembelajaran cerdas, Teknologi Web Modern (termasuk React/Next.js dan Supabase), serta Evaluasi User Experience (khususnya model Hedonic-Motivation System Adoption Model atau HMSAM).

#### 3.2 Perancangan Website

Perancangan *website* "Starcoder" dilakukan melalui beberapa tahap, yaitu perancangan gamifikasi, perancangan *Intelligent Tutoring System* (ITS), perancangan arsitektur teknis, perancangan *flowchart* dan *sitemap*, perancangan antarmuka pengguna (UI), serta perancangan aset visual dan musik.

##### 3.2.1 Perancangan Gamifikasi

Perancangan gamifikasi dalam sistem Starcoder didasarkan pada framework Octalysis yang dikembangkan oleh Yu-kai Chou. Framework ini mengidentifikasi delapan *core drives* yang memotivasi perilaku manusia dalam konteks permainan. Implementasi framework Octalysis dalam sistem Starcoder mencakup beberapa elemen kunci yang dirancang untuk meningkatkan motivasi dan keterlibatan mahasiswa.

##### 1. *Epic Meaning & Calling*

*Epic Meaning & Calling* diimplementasikan melalui narasi sci-fi yang menempatkan mahasiswa sebagai "Starcoder", seorang programmer elit yang bertugas menyelamatkan galaksi melalui kemampuan coding. Setiap misi pembelajaran dibingkai sebagai tantangan yang memiliki dampak signifikan terhadap storyline utama, memberikan sense of purpose yang kuat kepada mahasiswa.

## 2. *Development & Accomplishment*

*Development & Accomplishment* diwujudkan melalui sistem XP (Experience Points) yang komprehensif, tingkatan (levels), dan pencapaian (achievements). Mahasiswa memperoleh XP dari berbagai aktivitas: menyelesaikan materi pembelajaran, menjawab kuis dengan benar, memecahkan masalah coding, dan berpartisipasi dalam daily challenges. Sistem level dirancang dengan kurva progression yang seimbang untuk mempertahankan sense of achievement tanpa membuat mahasiswa merasa overwhelmed.

## 3. *Empowerment of Creativity & Feedback*

*Empowerment of Creativity & Feedback* diintegrasikan melalui AI companion bernama M.E.C.H.A. yang memberikan hint adaptif dan feedback real-time. AI companion menggunakan natural language processing untuk memberikan respons yang personal dan contextual, membantu mahasiswa ketika menghadapi kesulitan tanpa memberikan jawaban langsung.

## 4. *Ownership & Possession*

*Ownership & Possession* diwujudkan melalui sistem kustomisasi *Profile Picture* (PFP) yang dapat di-unlock berdasarkan level, serta "User Dossier" pribadi. Elemen ini dirancang untuk memberikan rasa kepemilikan dan identitas unik bagi setiap pengguna, yang membangun keterikatan personal dengan akun mereka.

## 5. *Social Influence & Relatedness*

*Social Influence & Relatedness* diimplementasikan melalui sistem leaderboard yang menampilkan ranking berdasarkan XP dan achievements. Fitur "Hall of Legends" memungkinkan mahasiswa untuk melihat prestasi teman-temannya dan termotivasi untuk mencapai posisi yang lebih tinggi.

## 6. *Scarcity & Impatience*

*Scarcity & Impatience* diwujudkan melalui sistem "Daily Challenge" yang hanya tersedia satu kali per hari. Kelangkaan ini menciptakan rasa urgensi dan nilai eksklusif pada kesempatan tersebut, mendorong pengguna untuk kembali setiap hari agar tidak ketinggalan.

## 7. *Unpredictability & Curiosity*

*Unpredictability & Curiosity* diwujudkan melalui mekanisme "Gacha Lootbox" yang memberikan *random rewards* (hadiah acak) setelah menyelesaikan *daily challenge*. Elemen kejutan dan antisipasi ini dirancang untuk mempertahankan engagement jangka panjang.

#### 8. *Avoidance & Loss*

*Avoidance & Loss* diimplementasikan melalui sistem streak yang mendorong mahasiswa untuk login dan berpartisipasi secara konsisten. Kehilangan streak mengakibatkan hilangnya bonus, menciptakan motivasi ekstrinsik yang sehat untuk mempertahankan progres.

### 3.2.2 Formulasi Matematis Sistem

Sistem menerapkan beberapa perhitungan matematis untuk memastikan konsistensi mekanisme permainan. Berikut adalah rincian formulasi yang digunakan.

Rumus 3.1 menunjukkan cara perhitungan *Level* pengguna berdasarkan total XP.

$$L = \max \left( 1, \left\lfloor \frac{XP_{total}}{200} \right\rfloor + 1 \right) \quad (3.1)$$

Dimana  $L$  adalah level saat ini dan fungsi  $\lfloor x \rfloor$  membulatkan nilai ke bawah.

Rumus 3.2 menunjukkan distribusi probabilitas hadiah *random* pada *Daily Challenge* berdasarkan bilangan acak  $r$  ( $0 \leq r < 1$ ).

$$R(r) = \begin{cases} \text{Legendary,} & 0 \leq r < 0.05 \\ \text{Epic,} & 0.05 \leq r < 0.15 \\ \text{Rare,} & 0.15 \leq r < 0.40 \\ \text{Common,} & 0.40 \leq r < 1.0 \end{cases} \quad (3.2)$$

Rumus 3.3 menunjukkan logika perhitungan *Streak* harian berdasarkan selisih hari ( $\Delta d$ ) login.

$$S_{new} = \begin{cases} S_{old}, & \text{jika } \Delta d = 0 \text{ (Hari sama)} \\ S_{old} + 1, & \text{jika } \Delta d = 1 \text{ (Berturut-turut)} \\ 1, & \text{jika } \Delta d \geq 2 \text{ (Terputus)} \end{cases} \quad (3.3)$$

Rumus 3.4 menunjukkan cara perhitungan skor penilaian untuk kuis dan coding.

$$Score = \left( \frac{C}{N} \right) \times 100 \quad (3.4)$$

Dimana  $C$  adalah jumlah jawaban benar dan  $N$  adalah total pertanyaan.

### 3.2.3 Perancangan Intelligent Tutoring System

Sistem pembelajaran cerdas didasarkan pada arsitektur *Intelligent Tutoring System* (ITS) dalam platform Starcoder yang dirancang mengikuti model klasik ITS. Model ini terdiri dari empat komponen utama yang diintegrasikan dengan elemen gamifikasi: *Domain Model*, *Student Model*, *Pedagogical Model*, dan *Interface Model*.

*Domain Model* merepresentasikan struktur pengetahuan yang diajarkan, termasuk hierarki konsep, dependensi prasyarat, dan kesalahan umum (*common misconceptions*). *Student Model* melacak data kinerja dan perilaku mahasiswa (seperti waktu per konsep, dan interaksi AI) menggunakan pendekatan *overlay model* yang diperkaya data gamifikasi.

*Pedagogical Model* berfungsi sebagai otak adaptif yang menerapkan strategi *adaptive sequencing*. Ketika mahasiswa terdeteksi mengalami kesulitan (melalui *failed attempts* atau *extended time on task*), model ini secara otomatis menyediakan misi remedial (*foundational missions*), mengaktifkan *AI companion* untuk bantuan intensif, atau menawarkan bantuan lainnya.

Terakhir, *Interface Model* adalah antarmuka *game-like* di mana *AI companion* (M.E.C.H.A.) berperan sebagai agen cerdas yang memberikan petunjuk kontekstual, menjelaskan *error*, dan memberikan dorongan personal. Keseluruhan algoritma adaptasi ini menggunakan pendekatan berbasis aturan (*rule-based*) untuk menganalisis kinerja secara *real-time* dan membuat keputusan pedagogis.



### 3.2.4 Perancangan Arsitektur Teknis

Arsitektur teknis sistem Starcoder dirancang menggunakan teknologi web modern dengan prinsip skalabilitas (*scalability*), pemeliharaan (*maintainability*), dan kinerja (*performance*). Arsitektur ini mengikuti pola *microservices* yang memungkinkan setiap komponen untuk dikembangkan dan di-*deploy* secara independen.

*Arsitektur Frontend* menggunakan Next.js 15 dengan React Server Components untuk kinerja optimal. Aplikasi diorganisasikan menggunakan App Router yang memungkinkan *file-based routing* dan *nested layouts*. Manajemen status (*state management*) menggunakan kombinasi Zustand untuk status global dan React hooks untuk status komponen lokal. Penataan gaya (*styling*) menggunakan Tailwind CSS dengan *custom design system* yang konsisten dengan tema sci-fi.

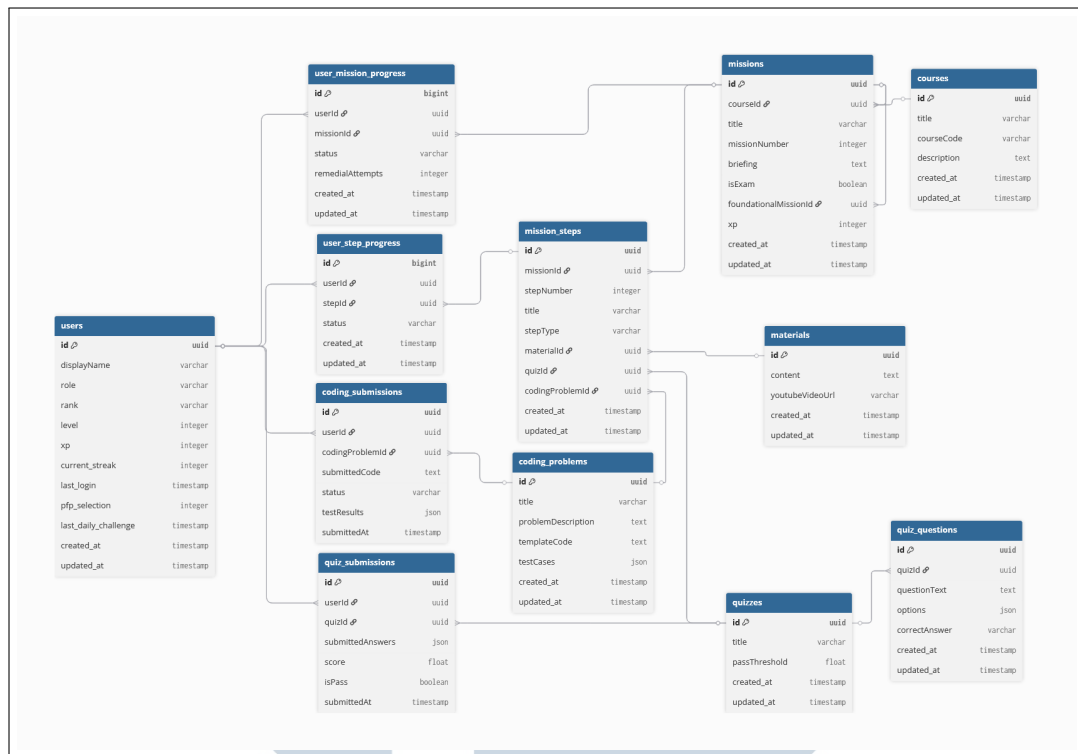
*Arsitektur Backend* menggunakan Supabase sebagai *Backend-as-a-Service* (BaaS) yang menyediakan database PostgreSQL, *real-time subscriptions*, otentikasi, dan *edge functions*. Skema database dirancang dengan normalisasi yang optimal untuk mendukung *query* kompleks yang diperlukan untuk fungsionalitas ITS.

*Integrasi AI* menggunakan Google Gemini API untuk *natural language processing* (NLP) pada AI companion. Integrasi dilakukan melalui *custom API endpoints* yang menangani konteks percakapan, analisis sentimen, dan pembuatan respons adaptif. Sistem juga menggunakan Judge0 API untuk eksekusi kode (*secure code execution*) dan pengujian otomatis.

*Arsitektur Keamanan* mengimplementasikan *Row Level Security* (RLS) pada level database untuk memastikan isolasi data antar pengguna. Otentikasi menggunakan token JWT dengan mekanisme *refresh*. Semua komunikasi API menggunakan HTTPS dengan konfigurasi CORS yang tepat.

### 3.2.5 Perancangan Database

Perancangan database sistem Starcoder menggunakan PostgreSQL sebagai *database management system* melalui platform Supabase. Struktur database dirancang dengan pendekatan normalisasi untuk mendukung fungsionalitas *Intelligent Tutoring System* (ITS), gamifikasi, dan *real-time progress tracking*. *Entity Relationship Diagram* (ERD) dari sistem Starcoder dapat dilihat pada Gambar 3.1.



Gambar 3.1. Entity Relationship Diagram Sistem Starcoder

Desain database Starcoder pada Gambar 3.1 terbagi menjadi tiga kategori utama: *Core Tables* untuk entitas utama, *Progress Tracking* untuk pelacakan kemajuan pengguna, dan *Submissions* untuk menyimpan hasil pengerjaan. Pemisahan ini bertujuan untuk mengoptimalkan *query performance* dan memudahkan skalabilitas sistem.

Tabel User dirancang sebagai entitas sentral yang menyimpan informasi dasar pengguna termasuk elemen gamifikasi seperti level, xp, rank, dan current\_streak. Field pfp\_selection mendukung sistem kustomisasi *Profile Picture*, sementara last\_daily\_challenge mengontrol mekanisme *scarcity* pada *daily challenge*. Penggunaan UUID sebagai *primary key* memastikan kompatibilitas dengan sistem autentikasi Supabase dan mencegah *enumeration attacks*.

Hierarki konten pembelajaran direpresentasikan melalui relasi Course → Mission → MissionStep. Tabel Mission memiliki field foundationalMissionId yang memungkinkan implementasi sistem ITS dengan menghubungkan misi utama ke misi remedial. Field isExam memungkinkan diferensiasi antara misi pembelajaran reguler dan ujian akhir. Setiap MissionStep dapat bertipe *material*, *quiz*, atau *live\_coding* yang dinyatakan dalam enum StepType, dengan foreign key yang menunjuk ke tabel konten spesifik (Material,

Quiz, atau CodingProblem).

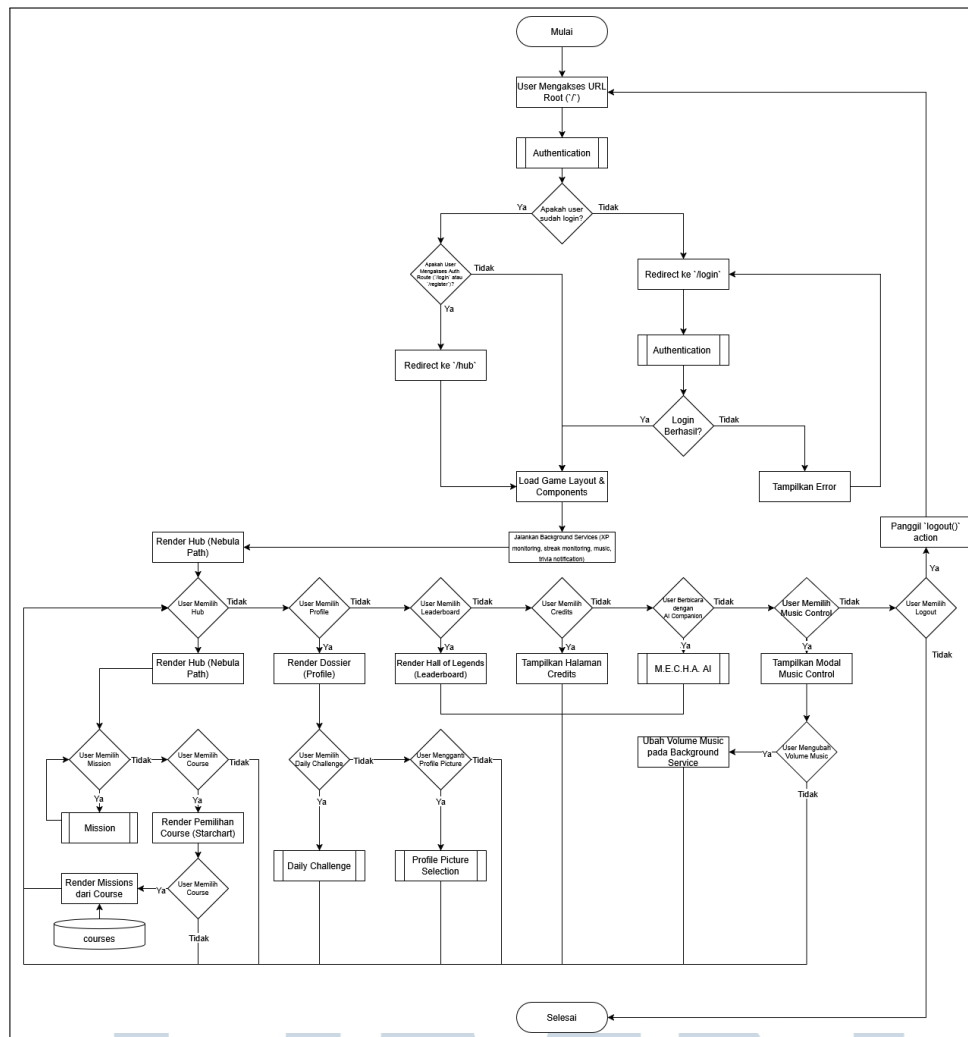
Sistem pelacakan kemajuan menggunakan enum `ProgressStatus` dengan nilai *locked*, *unlocked*, dan *complete* untuk mengontrol *adaptive sequencing*. Tabel `UserMissionProgress` mencatat status misi pengguna dan field `remedialAttempts` untuk melacak jumlah percobaan remedial. Tabel `UserStepProgress` memberikan granularitas pelacakan pada level step individual, memungkinkan sistem untuk mengetahui secara tepat di mana pengguna mengalami kesulitan.

Penyimpanan hasil pengerjaan dipisahkan berdasarkan tipe konten. Tabel `QuizSubmission` menyimpan jawaban dalam format JSON untuk fleksibilitas struktur pertanyaan, sementara `CodingSubmission` menggunakan enum `SubmissionStatus` untuk melacak status eksekusi kode melalui *Judge0 API*. Field `testResults` dalam format JSON memungkinkan penyimpanan hasil *test case* yang kompleks dan detail *error messages*.

### 3.2.6 Perancangan Flowchart

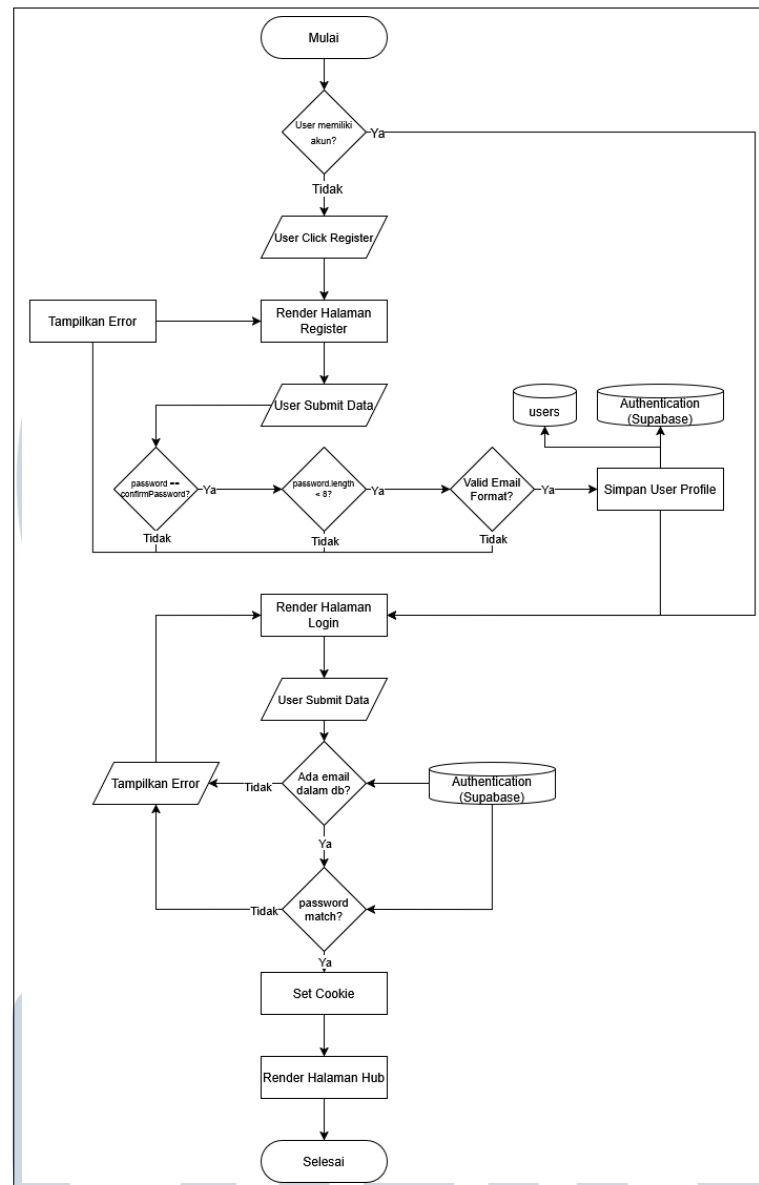
Bagian ini menjelaskan tujuh *flowchart* utama yang menggambarkan alur proses dalam sistem Starcoder. Setiap *flowchart* dirancang untuk mempresentasikan alur kerja yang spesifik dalam aplikasi. Berikut flowchart-flowchartnya:





Gambar 3.2. Flowchart Utama Sistem (Master Flow)

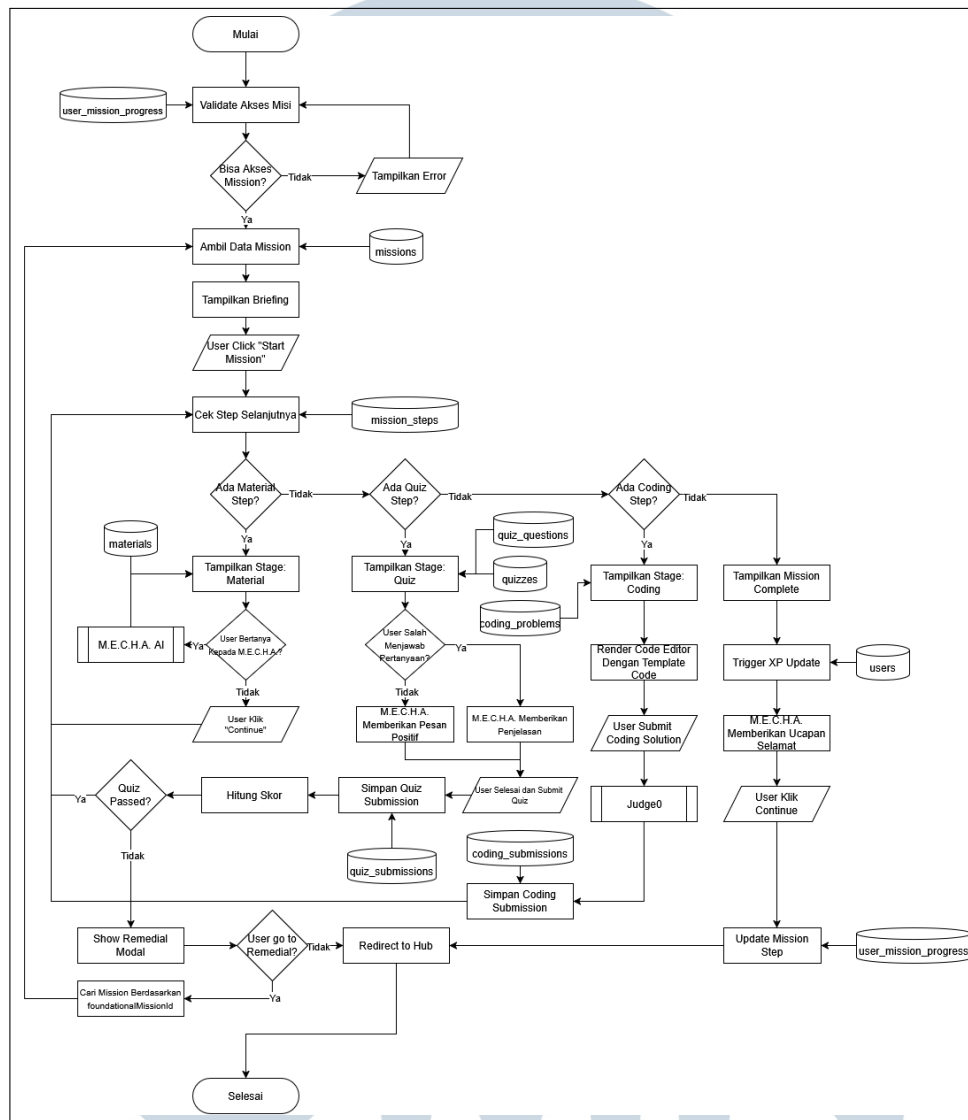
Gambar 3.2 memberikan gambaran menyeluruh tentang arsitektur alur sistem Starcoder. Proses dimulai ketika pengguna mengakses URL root aplikasi. Sistem secara otomatis melakukan pengecekan status autentikasi; jika pengguna belum login, mereka diarahkan ke halaman login. Setelah berhasil masuk, pengguna diarahkan ke Hub Utama (*Nebula Path*) yang berfungsi sebagai pusat navigasi. Dari sini, pengguna dapat mengakses berbagai modul fitur utama seperti Misi Pembelajaran, *Daily Challenge*, *Leaderboard*, Profil Pengguna, dan interaksi dengan Agen AI (M.E.C.H.A.). Dalam aplikasi juga ada *background services* yang berjalan secara paralel untuk memantau perolehan XP, status *streak*, dan trivia dari M.E.C.H.A. AI.



Gambar 3.3. Flowchart Proses Autentikasi Pengguna

Proses autentikasi yang ditampilkan pada Gambar 3.3 memisahkan alur registrasi dan login secara eksplisit. Jika pengguna memilih *register*, sistem meminta input data registrasi dan melakukan validasi form. Apabila validasi berhasil, sistem memanggil *Server Action* `register()` untuk memproses registrasi ke database. Sistem kemudian memeriksa apakah akun tersedia melalui validasi database. Jika tersedia, sistem membuat user baru dan menampilkan pesan sukses, namun jika tidak tersedia sistem menampilkan error dan kembali ke halaman input. Untuk alur *login*, sistem meminta input email dan password, melakukan validasi form, kemudian memanggil *Server Action* `login()`. Setelah validasi database, jika

kredensial benar sistem membuat sesi dan mengarahkan pengguna ke /hub, namun jika salah sistem menampilkan error dan kembali ke halaman input.

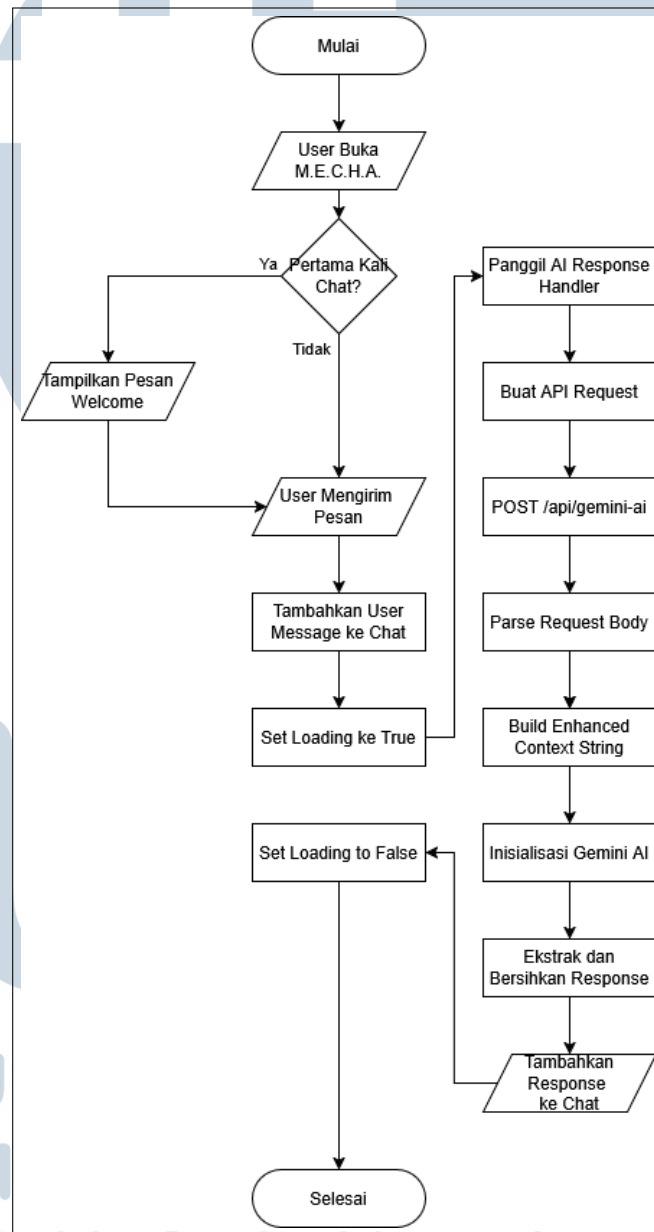


Gambar 3.4. Flowchart Alur Misi Pembelajaran Terintegrasi

Gambar 3.4 mengilustrasikan alur inti dari pengalaman pembelajaran. Sistem melakukan *load* data misi dan memeriksa apakah misi ditemukan - jika tidak ditemukan di-*redirect* ke /hub. Setelah misi ditemukan, sistem menampilkan *Mission Briefing* dan memulai *mission steps*. Sistem mengidentifikasi tipe *step* yang dijalankan. Untuk *step* bertipe "Material", sistem menampilkan konten materi dan menunggu pengguna mengklik tombol "Continue". Untuk *step* bertipe "Quiz", sistem menampilkan kuis dan menunggu *submit* dari pengguna, kemudian mengevaluasi apakah skor lebih dari *passThreshold* - jika tidak lulus menampilkan



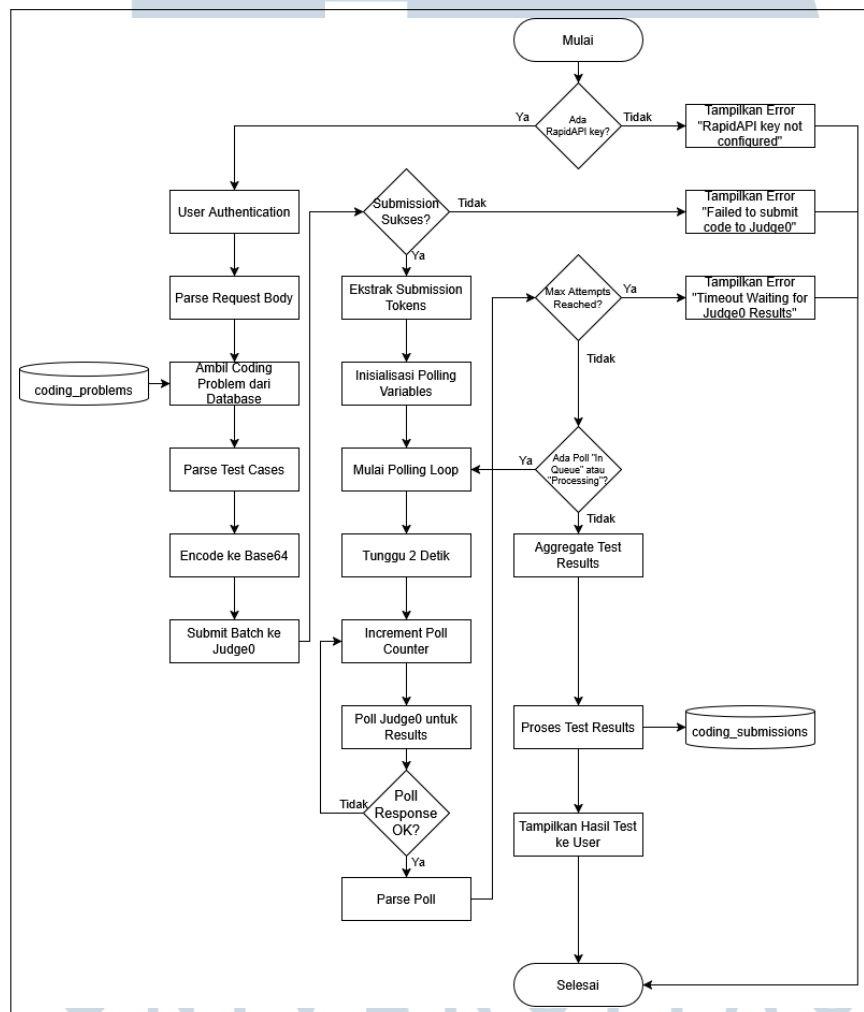
”Quiz Failed” dan kembali ke tampilan kuis. Untuk *step* bertipe ”Coding”, sistem menampilkan *code editor* dan menunggu *submit* kode dari pengguna, kemudian memproses melalui *Judge0* (dijelaskan pada *flowchart* terpisah). Jika semua *step* berhasil diselesaikan, sistem memberikan *reward* XP, menandai misi selesai di database, menampilkan *Mission Complete*, dan mengarahkan kembali ke /hub.



Gambar 3.5. Flowchart Logika Agen AI (M.E.C.H.A.)

Mekanisme kerja Agen AI digambarkan secara rinci pada Gambar 3.5. Berbeda dengan *chatbot* sederhana, sistem ini membangun konteks yang diperkaya

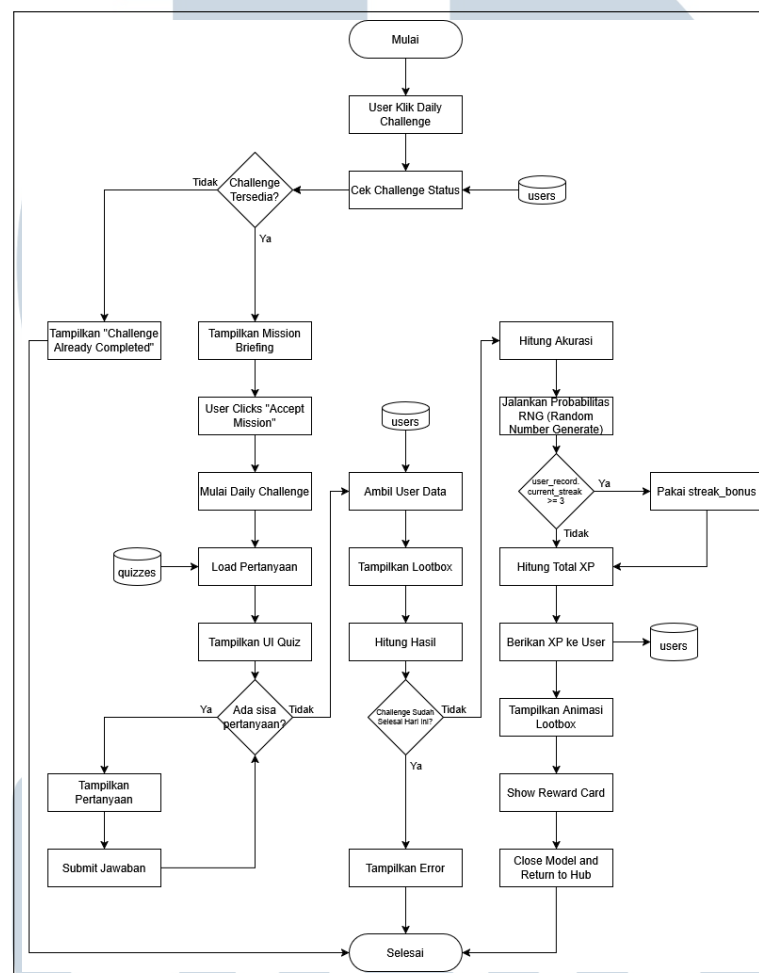
(*Enhanced Context String*) sebelum mengirim permintaan ke API Gemini. Konteks ini mencakup data misi saat ini, kode yang sedang ditulis pengguna, dan riwayat interaksi sebelumnya. Proses dimulai ketika pengguna mengirim pesan, yang kemudian diproses oleh *AI Response Handler*. Handler ini menyusun *payload* API, mengirimkannya ke *endpoint* /api/gemini-ai, dan memproses respons balasan untuk dibersihkan dari format yang tidak perlu sebelum ditampilkan kembali ke antarmuka obrolan pengguna.



Gambar 3.6. Flowchart Eksekusi Kode Asinkron dengan Judge0

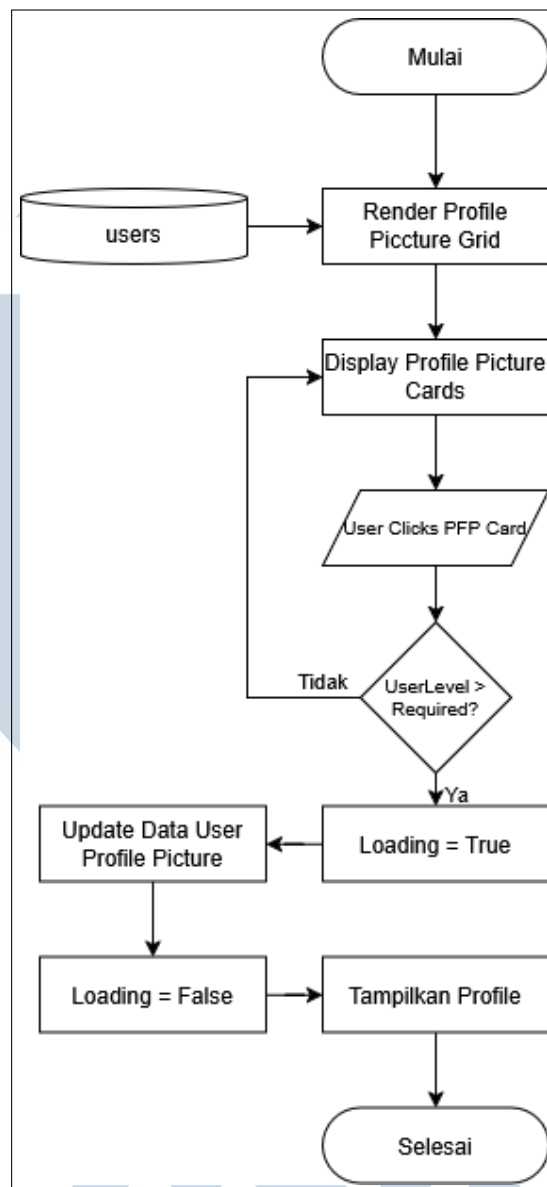
Gambar 3.6 menjelaskan proses teknis eksekusi kode yang aman. Alur ini menggunakan mekanisme *polling* untuk menangani sifat asinkron dari kompilasi kode. Setelah pengguna mengirimkan solusi, sistem mengambil *test cases* dari tabel `coding_problems`, melakukan penyandian Base64, dan mengirimkan *batch submission* ke Judge0 API. Sistem kemudian memasuki siklus *looping* (menunggu

2 detik setiap iterasi) untuk memeriksa status eksekusi. Proses ini berlanjut hingga Judge0 mengembalikan status selesai atau hingga batas waktu tercapai. Hasil akhirnya kemudian diproses untuk menentukan kelulusan berdasarkan perbandingan output kode dengan output yang diharapkan.



Gambar 3.7. Flowchart Logika Gamifikasi Daily Challenge

Logika gamifikasi untuk tantangan harian divisualisasikan pada Gambar 3.7. Sistem menerapkan pembatasan akses berdasarkan status *challenge* harian pengguna. Jika tersedia, pengguna diberikan kuis acak. Poin penting dalam alur ini adalah implementasi algoritma RNG (*Random Number Generation*) untuk penentuan hadiah setelah penyelesaian sukses. Sistem juga secara eksplisit memeriksa status *streak* pengguna ( $\text{current\_streak} \geq 3$ ); jika kondisi terpenuhi, bonus tambahan (*streak\_bonus*) ditambahkan ke total XP yang diperoleh, memberikan insentif untuk menjawab semua pertanyaan kuis dengan benar.

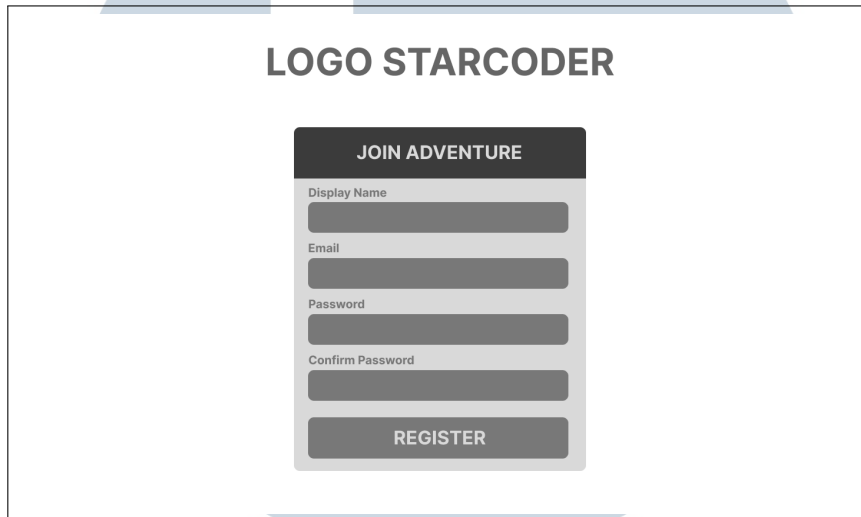


Gambar 3.8. Flowchart Sistem Kustomisasi Profil

Gambar 3.8 menggambarkan logika validasi untuk fitur personalisasi. Alur ini menekankan pada mekanisme *gating* berbasis level (*UserLevel > Required?*). Sistem tidak hanya menampilkan daftar gambar profil, tetapi juga memvalidasi kelayakan pengguna untuk menggunakan gambar tertentu berdasarkan pencapaian level mereka. Proses pembaruan data hanya dieksekusi jika validasi level berhasil, memastikan bahwa item kosmetik dalam aplikasi berfungsi sebagai simbol status pencapaian yang valid.

### 3.2.7 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna sistem Starcoder dibuat dalam bentuk *low-fidelity mockups* yang berfokus pada tata letak dan fungsionalitas setiap elemen *user interface*. Antarmuka dirancang untuk mendukung dua jenis pengguna utama: pengguna reguler (kadet) dan administrator sistem.

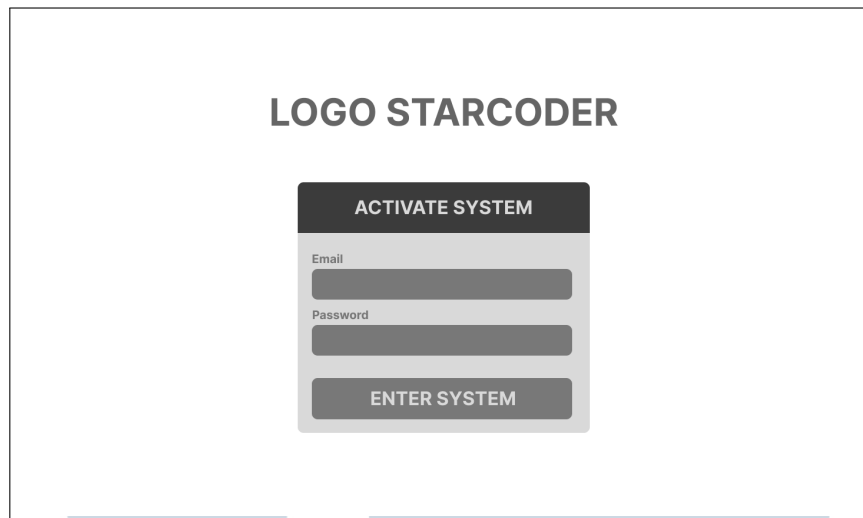


The mockup shows a registration form titled "JOIN ADVENTURE" within a larger container labeled "LOGO STARCODER". The form includes four input fields: "Display Name", "Email", "Password", and "Confirm Password". Below these fields is a "REGISTER" button.

Gambar 3.9. Mockup Halaman Registrasi

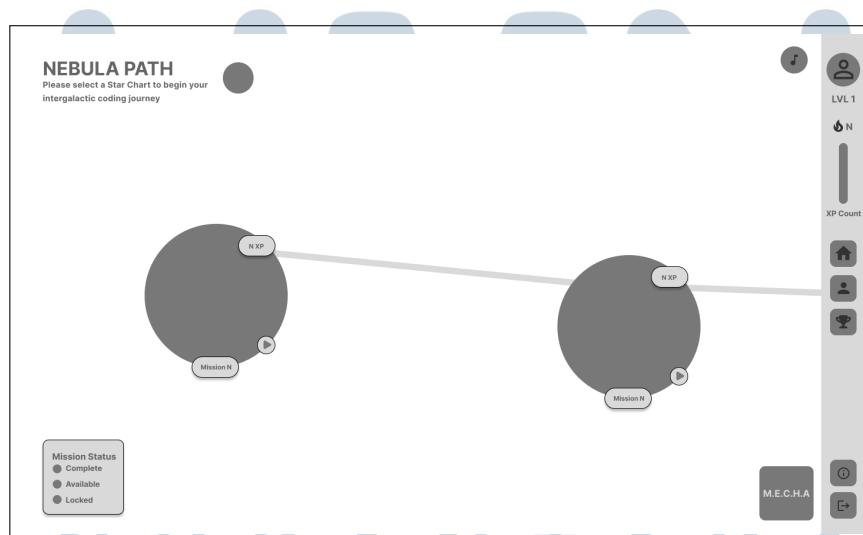
Halaman registrasi pada Gambar 3.9 menampilkan *form* "Join Adventure" yang berfungsi sebagai gerbang masuk pengguna baru ke sistem. *Form* ini terdiri dari tiga *input field* utama: nama lengkap pengguna untuk identifikasi dalam sistem, alamat email sebagai *username* dan sarana komunikasi, serta *password* untuk keamanan akun. Bagian bawah *form* dilengkapi dengan tombol "Register" yang mengeksekusi proses pembuatan akun dan tombol navigasi untuk beralih ke halaman *login* bagi pengguna yang sudah memiliki akun.

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 3.10. Mockup Halaman Login

Halaman *login* pada Gambar 3.10 menyediakan *form* "Activate System" yang memfasilitasi proses autentikasi pengguna terdaftar. *Interface* ini menampilkan dua *input field* utama: alamat email sebagai identifikator pengguna dan *password* untuk verifikasi identitas. Area bawah *form* memuat tombol "Enter System" yang memproses kredensial masuk dan mengarahkan pengguna ke *dashboard* utama. Terdapat pula tautan navigasi yang memungkinkan pengguna beralih ke halaman registrasi jika belum memiliki akun.

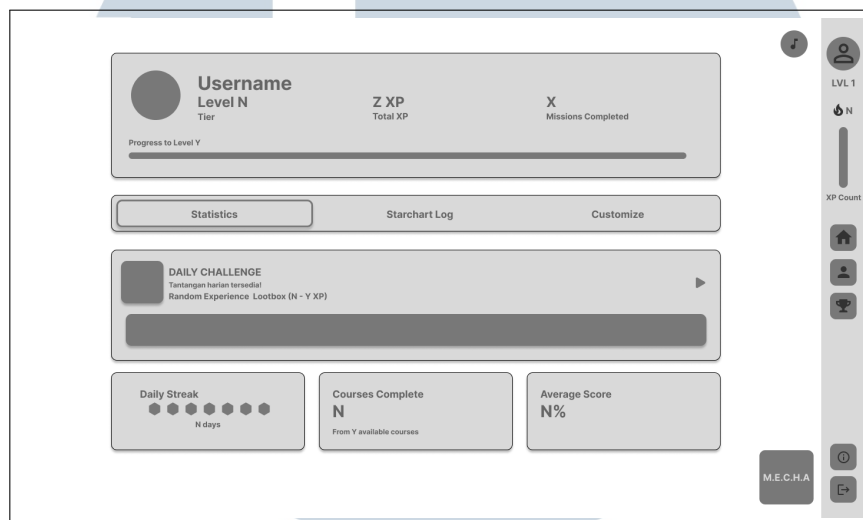


Gambar 3.11. Mockup Halaman Hub Utama (Nebula Path)

Halaman *hub* utama pada Gambar 3.11 berfungsi sebagai pusat navigasi yang menampilkan peta misi dalam bentuk jalur planet-planet. Area tengah memuat



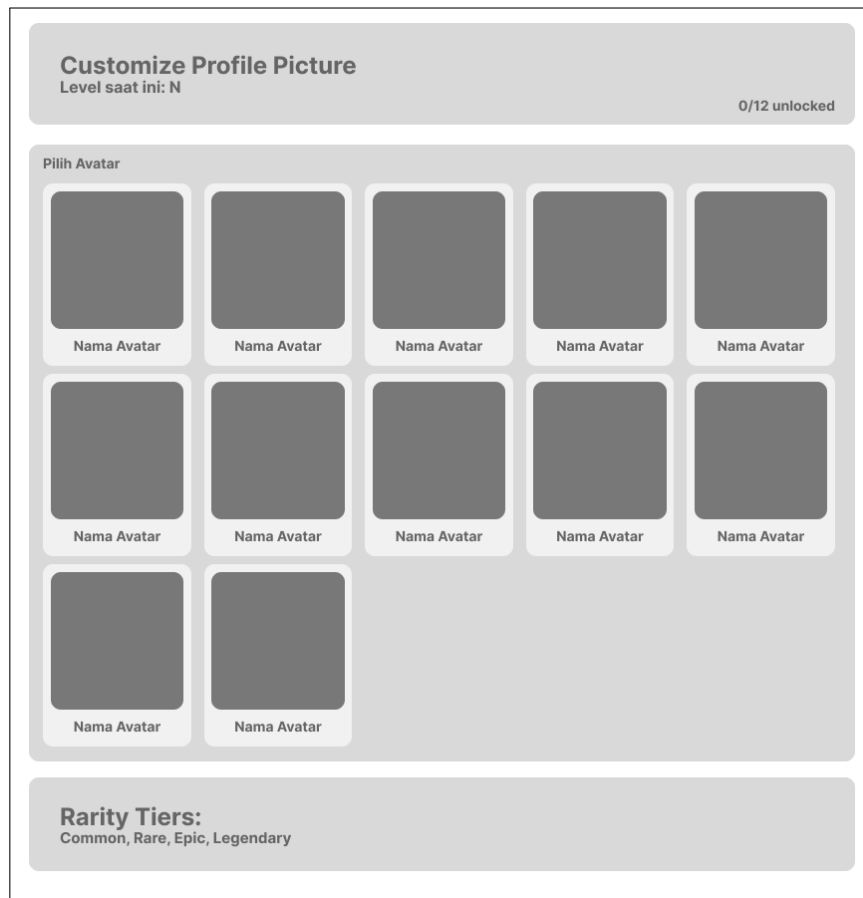
*mission map* yang memvisualisasikan progres pembelajaran pengguna melalui representasi planet yang dapat diklik untuk mengakses misi tertentu. *Sidebar* kanan menampilkan informasi status pengguna meliputi nama, *level* saat ini, jumlah XP, dan *progress bar* menuju *level* berikutnya. Panel kiri menyediakan *Mission Status Legend* yang mengkategorikan misi berdasarkan status: terkunci, tersedia, sedang dikerjakan, atau telah diselesaikan.



Gambar 3.12. Mockup Halaman Profil Pengguna (User Dossier)

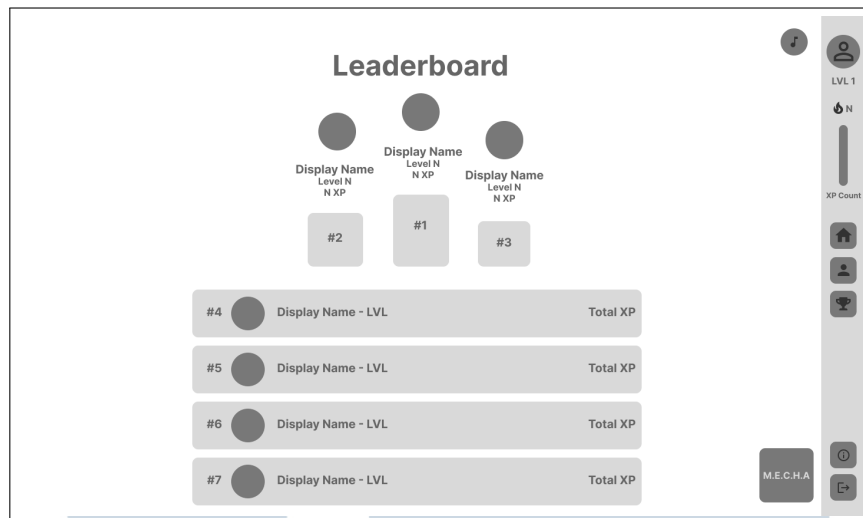
Halaman profil pengguna pada Gambar 3.12 menampilkan informasi komprehensif tentang status dan pencapaian kadet dalam sistem. *Header* bagian atas memuat informasi identitas pengguna termasuk *username*, *level* saat ini, total XP yang diperoleh, dan *progress bar* yang menunjukkan kemajuan menuju *level* berikutnya. Area utama dilengkapi dengan sistem tab navigasi yang terdiri dari tiga bagian: "Statistics" untuk menampilkan statistik pembelajaran, "Starchart Log" untuk melacak riwayat aktivitas pengguna, dan "Customize" untuk mengakses fitur kustomisasi *profile picture* dan preferensi lainnya.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



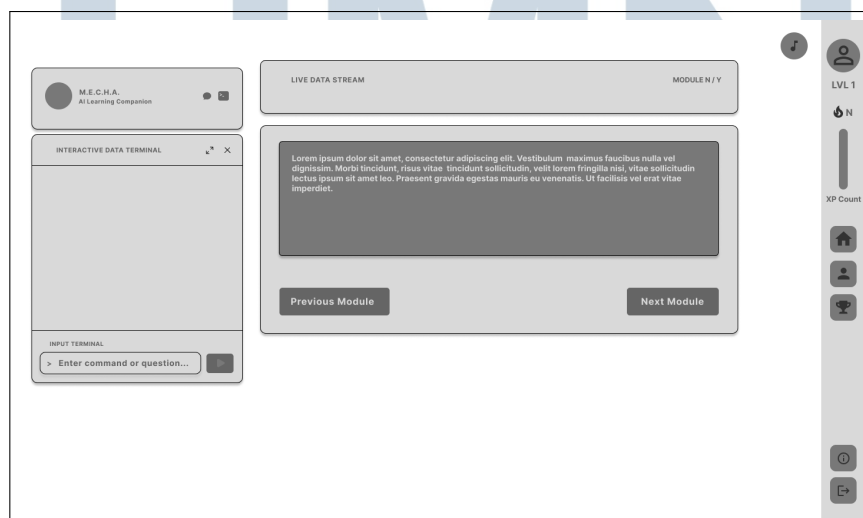
Gambar 3.13. Mockup Halaman Kustomisasi Profile Picture

Halaman kustomisasi *Profile Picture* pada Gambar 3.13 menyediakan antarmuka untuk personalisasi avatar pengguna berdasarkan pencapaian *level*. Area utama menampilkan *grid layout* yang memuat koleksi *profile picture* yang tersedia dengan indikator visual untuk membedakan avatar yang sudah ter-*unlock* dan yang masih terkunci. Panel informasi menunjukkan status *level* pengguna saat ini dan jumlah total avatar yang telah berhasil di-*unlock*. Setiap avatar dilengkapi dengan informasi persyaratan *level* yang diperlukan untuk membuka kunci dan tombol seleksi untuk mengaktifkan avatar yang dipilih.



Gambar 3.14. Mockup Halaman Leaderboard

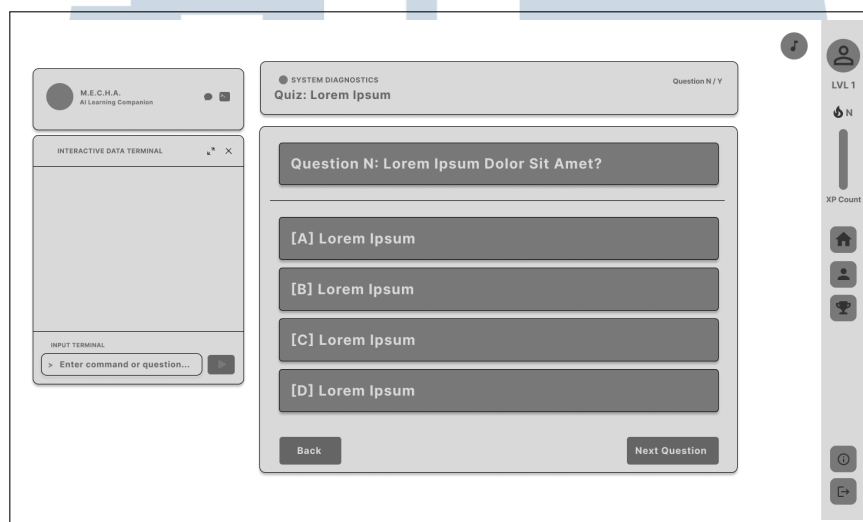
Halaman *leaderboard* pada Gambar 3.14 berfungsi sebagai sistem peringkat yang menampilkan pencapaian pengguna secara kompetitif. Area atas menampilkan podium Top 3 dengan desain khusus yang menonjolkan tiga pengguna dengan XP tertinggi beserta informasi *level* dan total XP mereka. Bagian bawah memuat daftar peringkat lengkap dalam format tabel yang mencantumkan posisi, nama pengguna, *level*, dan total XP secara berurutan. *Interface* ini memungkinkan pengguna untuk melihat posisi mereka relatif terhadap pengguna lain dan memotivasi kompetisi sehat dalam proses pembelajaran.



Gambar 3.15. Mockup Halaman Mission Material

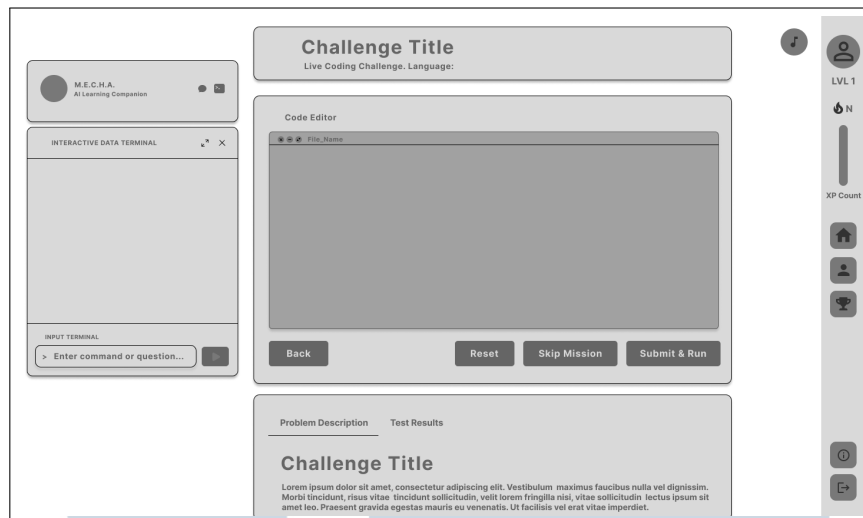
Halaman misi material pada Gambar 3.15 menggunakan *layout* dua kolom

dengan komponen M.E.C.H.A. (*AI companion*) di sebelah kiri dan area konten pembelajaran di sebelah kanan. Panel kiri menampilkan antarmuka M.E.C.H.A. yang menyediakan panduan kontekstual dan *feedback* selama proses pembelajaran. Area kanan memuat *display* materi pembelajaran dalam format teks, gambar, atau multimedia yang disesuaikan dengan topik misi. Bagian bawah dilengkapi dengan tombol navigasi "Previous Module" dan "Next Module" yang memungkinkan pengguna berpindah antar bagian materi secara sekuensial dengan kontrol progres yang jelas.



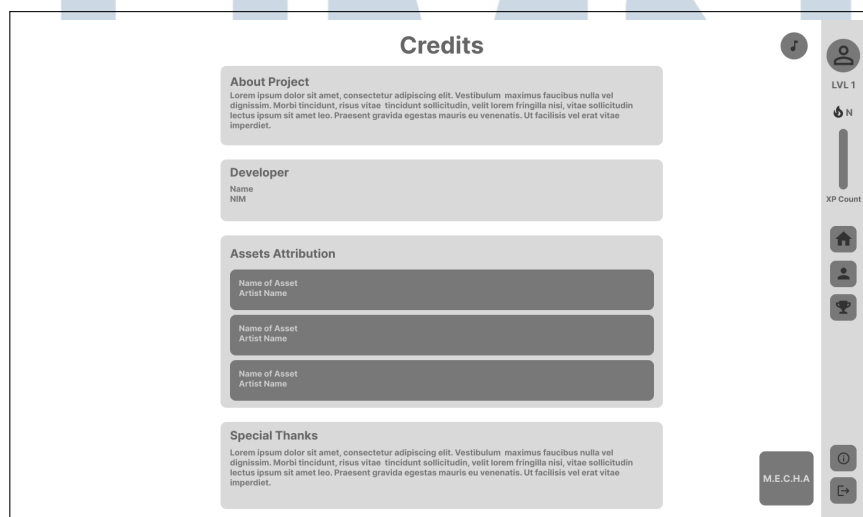
Gambar 3.16. Mockup Halaman Mission Quiz

Halaman misi kuis pada Gambar 3.16 menggunakan *layout* yang konsisten dengan komponen M.E.C.H.A. di panel kiri untuk memberikan bimbingan dan motivasi selama pengerjaan soal. Area kanan terbagi menjadi dua bagian utama: blok pertanyaan di atas yang menampilkan soal dengan format yang jelas dan mudah dibaca, serta blok empat opsi jawaban [A, B, C, D] di bawahnya yang disusun dalam format *multiple choice* standar. Panel navigasi di bagian bawah menyediakan tombol "Back Question" dan "Next Question" yang memungkinkan pengguna berpindah antar soal dengan fleksibilitas untuk meninjau kembali jawaban sebelumnya.



Gambar 3.17. Mockup Halaman Mission Coding

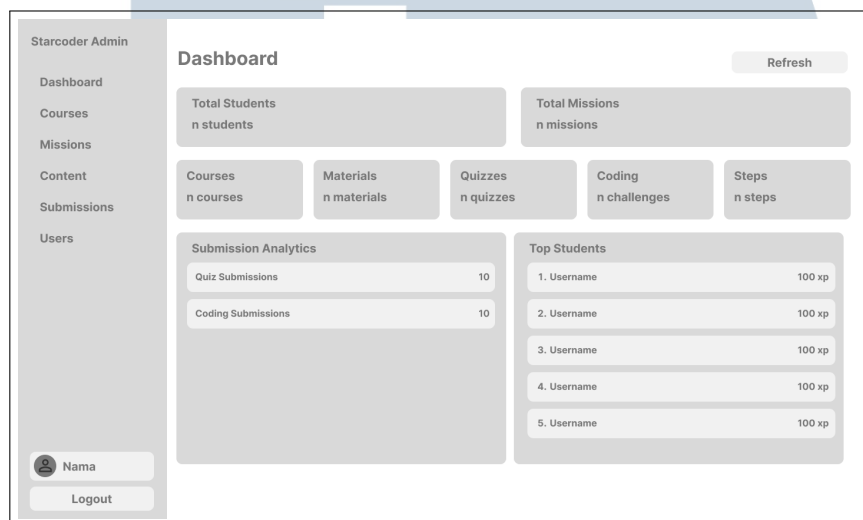
Halaman misi *coding* pada Gambar 3.17 menampilkan *layout* tiga panel dengan M.E.C.H.A. di panel kiri sebagai asisten pemrograman yang memberikan *hints* dan *debugging assistance*. Panel kanan atas memuat editor kode dengan fitur *syntax highlighting*, *line numbering*, dan *auto-completion* untuk mendukung proses penulisan kode. Panel kanan bawah terdiri dari sistem tab yang menampilkan "Problem Description" berisi deskripsi soal dan spesifikasi *input/output*, serta tab "Test Results" yang memuat hasil eksekusi kode termasuk status *pass/fail* untuk setiap *test case* dan pesan error jika diperlukan.



Gambar 3.18. Mockup Halaman Credits

Halaman *credits* pada Gambar 3.18 berfungsi sebagai halaman statis yang memberikan pengakuan dan informasi terkait pengembangan sistem. *Layout* terdiri

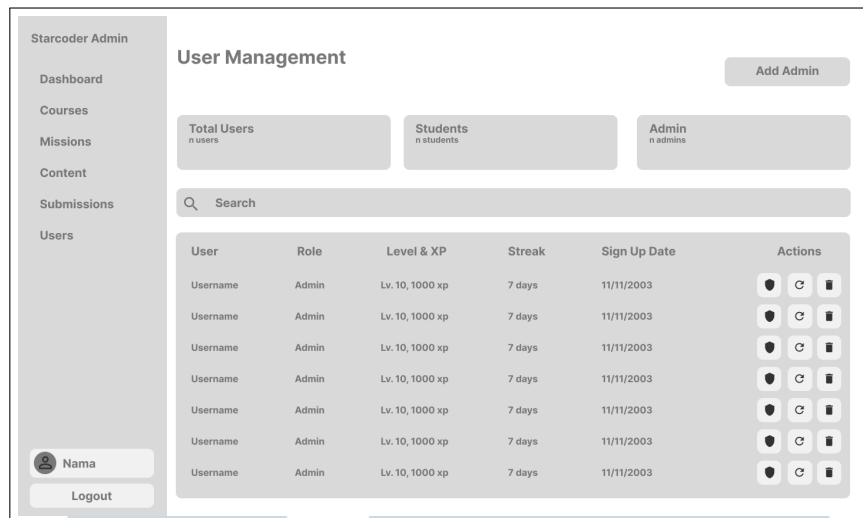
dari empat blok informasi utama yang disusun secara vertikal: "About Project" yang menjelaskan tujuan dan latar belakang pengembangan sistem Starcoder, "Developer" yang mencantumkan informasi pengembang dan kontributor, "Assets" yang memberikan kredit kepada sumber *asset* visual dan audio yang digunakan, serta "Special Thanks" yang mengakui pihak-pihak yang memberikan dukungan selama proses pengembangan. Setiap blok dirancang dengan hierarki visual yang jelas untuk memudahkan pembacaan informasi.



Gambar 3.19. Mockup Dashboard Administrator

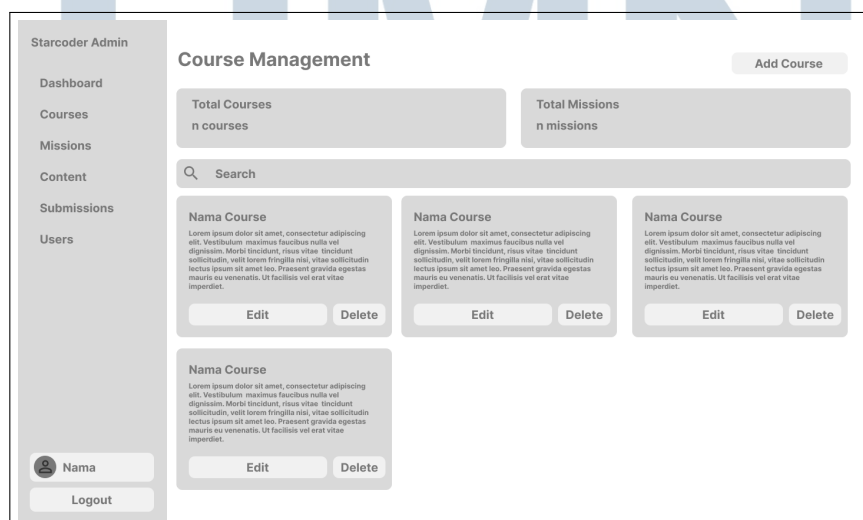
*Dashboard* administrator pada Gambar 3.19 menyediakan antarmuka pengelolaan sistem dengan *sidebar* navigasi di sebelah kiri yang memuat menu akses cepat ke berbagai modul administrasi. Area utama menampilkan empat *card* statistik kunci: "Total Students" yang menunjukkan jumlah pengguna terdaftar, "Total Missions" yang mencantumkan jumlah misi tersedia dalam sistem, "Submission Analytics" yang menyajikan metrik pengerjaan soal, dan "Top Students" yang menampilkan daftar pengguna dengan performa terbaik. Setiap *card* dilengkapi dengan visualisasi data dan indikator tren untuk membantu administrator memahami kondisi sistem secara cepat.





Gambar 3.20. Mockup Halaman User Management

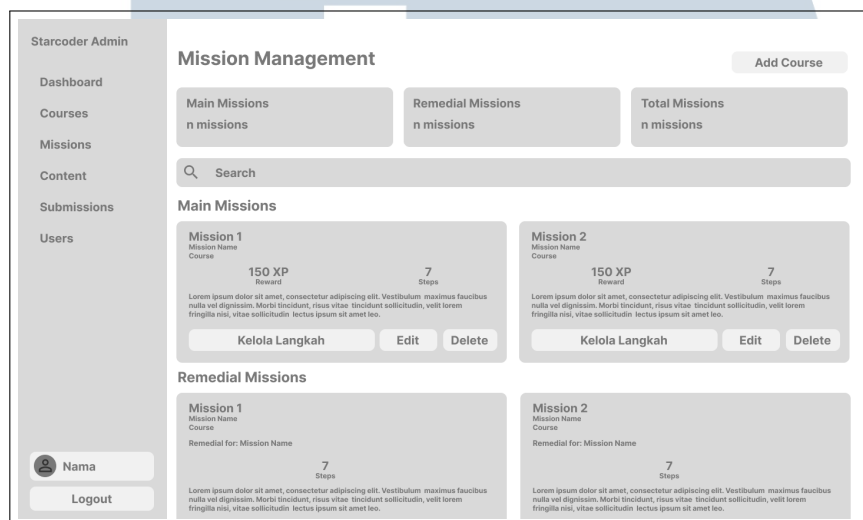
Halaman *User Management* pada Gambar 3.20 menyediakan antarmuka komprehensif untuk pengelolaan pengguna sistem. Area atas menampilkan *card* statistik yang merangkum informasi demografi pengguna dan metrik aktivitas. *Search bar* memungkinkan administrator melakukan pencarian pengguna berdasarkan nama atau email dengan fungsi *real-time filtering*. Tabel utama menampilkan daftar pengguna dengan kolom: "User" yang mencantumkan nama dan email, "Role" yang menunjukkan status pengguna (student/admin), "Level & XP" yang menampilkan progress pembelajaran, dan "Actions" yang menyediakan tombol untuk operasi edit, hapus, atau *suspend* akun pengguna.



Gambar 3.21. Mockup Halaman Course Management

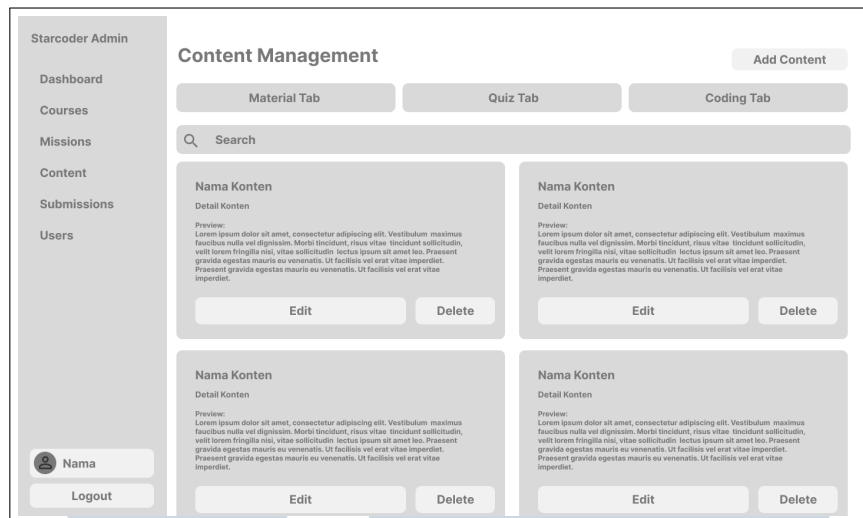
Halaman *Course Management* pada Gambar 3.21 dirancang untuk

pengelolaan kurikulum pembelajaran dengan *search bar* di bagian atas yang memfasilitasi pencarian *course* berdasarkan nama atau kategori. Area utama menampilkan daftar *course* dalam format *card layout* yang memuat informasi setiap *course* meliputi nama, deskripsi singkat, jumlah misi yang terkandung, dan status aktif/nonaktif. Setiap *card course* dilengkapi dengan tombol "Edit" untuk memodifikasi konten dan pengaturan *course*, serta tombol "Delete" untuk menghapus *course* dari sistem dengan konfirmasi keamanan.



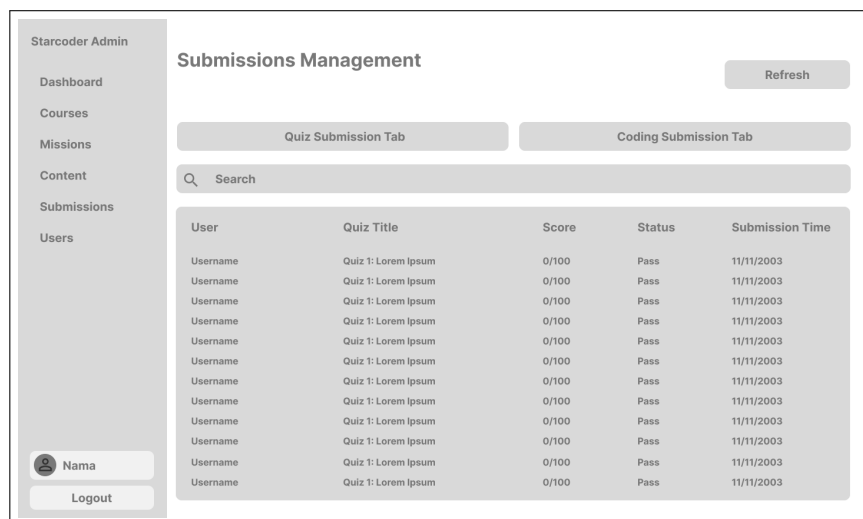
Gambar 3.22. Mockup Halaman Mission Management

Halaman *Mission Management* pada Gambar 3.22 menyediakan antarmuka pengelolaan misi dengan *card* statistik di bagian atas yang menampilkan jumlah total misi, misi aktif, dan tingkat penyelesaian rata-rata. *Search bar* memungkinkan administrator mencari misi berdasarkan judul atau kategori dengan fungsi *auto-suggestion*. Area konten terbagi menjadi dua kategori utama: "*Main Missions*" yang menampilkan daftar misi pembelajaran utama, dan "*Remedial Missions*" yang memuat misi tambahan untuk sistem *Intelligent Tutoring*. Setiap misi ditampilkan dalam format *card* dengan informasi tingkat kesulitan, durasi estimasi, dan tombol aksi untuk pengelolaan konten misi.



Gambar 3.23. Mockup Halaman Content Management

Halaman *Content Management* pada Gambar 3.23 menggunakan sistem tab untuk mengorganisir berbagai jenis konten pembelajaran. Tiga tab utama terdiri dari "Material" untuk pengelolaan konten teks dan multimedia, "Quiz" untuk manajemen soal dan jawaban, serta "Coding" untuk pengelolaan soal pemrograman dan *test cases*.



Gambar 3.24. Mockup Halaman Submissions Management

Halaman *Submissions Management* pada Gambar 3.24 menyediakan antarmuka monitoring dan evaluasi hasil pengerjaan pengguna melalui sistem tab yang terdiri dari "Quiz" untuk melihat hasil pengerjaan kuis dan "Coding" untuk memantau *submission* kode program. Tabel utama menampilkan daftar

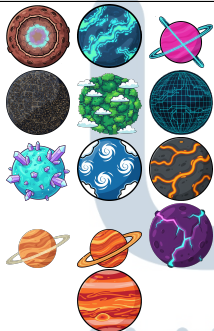
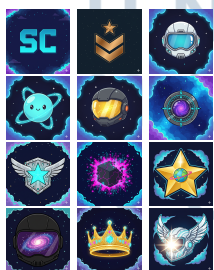
kiriman dengan kolom: "User" yang mencantumkan identitas pengguna, "Title" yang menunjukkan judul misi atau soal, "Score" yang menampilkan nilai yang diperoleh dengan indikator visual *pass/fail*, dan "Status" yang menunjukkan kondisi *submission* (pending, evaluated, atau flagged). *Interface* ini memungkinkan administrator melakukan review manual, memberikan *feedback*, atau menandai *submission* yang memerlukan perhatian khusus.

### 3.2.8 Perancangan Aset Visual dan Musik


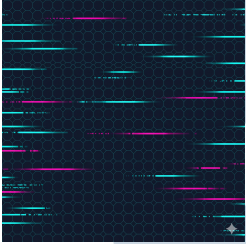


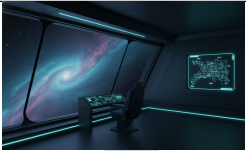

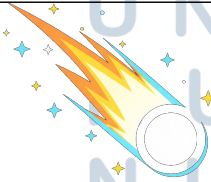
Perancangan aset visual dalam platform Starcoder dirancang untuk mendukung tema fiksi ilmiah dan mengimplementasikan prinsip desain "Clean Cockpit" yang memastikan antarmuka yang tidak berlebihan namun tetap imersif. Setiap elemen visual dipilih dan dirancang secara strategis untuk mendukung pengalaman pembelajaran yang kohesif dan menarik, sambil mempertahankan fokus pada konten edukatif yang esensial.

Aset-aset visual utama yang dirancang untuk platform Starcoder disajikan pada Tabel 3.1.




Tabel 3.1. Aset Visual Platform Starcoder

| Gambar  | Keterangan  | Sumber Gambar                     |
|---|---|-----------------------------------|
|  | Gambar planet yang di tunjukkan di dalam Nebula Path untuk memberikan rasa imersif dalam game                                 | AI Generated (Gemini Nano Banana) |
|  | Contoh PFP yang dapat di-unlock, mengimplementasikan Drive 4 (Kepemilikan) untuk memberikan rasa personalisasi dan pencapaian | AI Generated (Gemini Nano Banana) |
| Berlanjut ke halaman berikutnya   |   |                                   |

Tabel 3.1 – Aset Visual Platform Starcoder (lanjutan)

| Gambar  | Keterangan   | Sumber Gambar                     |
|---|--|-----------------------------------|
|    | Latar belakang tematik untuk Hall of Legends (Leaderboard)                 | AI Generated (Gemini Nano Banana) |
|    | Latar belakang untuk Daily Challenge                                       | AI Generated (Gemini Nano Banana) |
|   | Avatar untuk AI Companion M.E.C.H.A.                                       | AI Generated (Gemini Nano Banana) |
|  | Latar belakang tematik untuk Mission                                       | AI Generated (Gemini Nano Banana) |
|  | Latar belakang untuk User Dossier (Profile)                                | AI Generated (Gemini Nano Banana) |
|  | Overlay antarmuka 'Clean Cockpit', digunakan pada halaman autentikasi      | AI Generated (Gemini Nano Banana) |
|  | Aset animasi bintang jatuh, digunakan untuk efek visual di latar belakang. | AI Generated (Gemini Nano Banana) |
| Berlanjut ke halaman berikutnya   |  |                                   |

Tabel 3.1 – Aset Visual Platform Starcoder (lanjutan)

| Gambar   | Keterangan   | Sumber Gambar                     |
|--|--|-----------------------------------|
|   | Kursor kustom tematik Starcoder (versi besar)                                    | AI Generated (Gemini Nano Banana) |
|   | Icon untuk browser tab, mewakili logo aplikasi.                                  | AI Generated (Gemini Nano Banana) |
|  | Logo utama Starcoder, digunakan di halaman login, register dan sebagai branding. | AI Generated (Gemini Nano Banana) |

Selain aset visual, platform Starcoder menggunakan aset audio tematik untuk meningkatkan imersi. Atribusi untuk setiap trek musik latar yang digunakan disajikan pada Tabel 3.2.

Tabel 3.2. Atribusi Aset Musik

| File Aset                           | Konteks Penggunaan              | Atribusi (Sumber / Artis)                         |
|-------------------------------------|---------------------------------|---|
| gateway-cinematic-intro-theme.mp3   | Halaman Login & Registrasi      | "Phantom From Space" oleh Kevin MacLeod (YouTube) |
| nebula-path-ambient-exploration.mp3 | Halaman Utama / Nebula Path     | "Adrift" oleh Hayden Folker (YouTube)             |
| mission-focus-learning-calm.mp3     | Antarmuka Misi (Belajar/Coding) | "Space Aquarium" oleh CozyEra (YouTube)           |
| terminal-dossier-neutral-tech.mp3   | Menu Dossier / Profil Pengguna  | "Giving Way" oleh AERØHEAD (YouTube)              |
| Berlanjut ke halaman berikutnya     |                                 |   |



**Tabel 3.2 – Atribusi Aset Musik (lanjutan)**

| <b>File Aset</b>                       | <b>Konteks Penggunaan</b>         | <b>Atribusi (Sumber / Artis)</b>              |
|--|-----------------------------------|---|
| hall-of-legends-triumphant-ambient.mp3 | Papan Peringkat / Hall of Legends | "World of Magic" oleh Scott Buckley (YouTube) |

### **3.3 Pengembangan dan Implementasi**

Implementasi sistem Starcoder melibatkan beberapa penggunaan perangkat keras dan perangkat lunak spesifik. Berikut rincian spesifikasi perangkat yang digunakan dalam proses pengembangan:

#### **1. Perangkat Keras**

- (a) *Sistem Operasi: Windows 10*
- (b) *Prosesor: AMD Ryzen 5 5600U with Radeon Graphics, 2301 Mhz, 6 Core(s), 12 Logical Processor(s)*
- (c) *Memori: 16 GB DDR4 RAM*
- (d) *SSD: INTEL SSDPEKNW512G8H*
- (e) *Graphics Card: AMD Radeon(TM) Graphics*

#### **2. Perangkat Lunak**

- (a) *IDE: Visual Studio Code dengan extensions untuk React, TypeScript, dan Prisma*
- (b) *Bahasa Pemrograman: Typescript*
- (c) *Desain UI/UX: Figma untuk design UI dan mockup*
- (d) *Frontend Framework: Next.js 15 (React 19)*
- (e) *Styling: Tailwind CSS*
- (f) *Animasi: Framer Motion*
- (g) *Database: Supabase*
- (h) *ORM: Prisma*
- (i) *API Eksternal: Judge0 API untuk code execution, Google Gemini API untuk AI companion*

- (j) *Version Control System: Git dan GitHub*
- (k) *Package Manager: NPM (Node Package Manager)*
- (l) *Deployment: Vercel*

### 3.4 Uji Coba dan Evaluasi

Uji coba dilakukan kepada mahasiswa sebagai target utama dari sistem pembelajaran yang dirancang. Kriteria responden untuk penelitian ini adalah mahasiswa (dari universitas manapun) yang sedang mengambil atau telah mengambil mata kuliah Pemrograman Dasar, atau mata kuliah universitas yang setara. Evaluasi dilakukan menggunakan metode Hedonic-Motivation System Adoption Model (HMSAM), di mana setiap pertanyaan dinilai menggunakan skala Likert 5 poin. Pertanyaan-pertanyaan dari metode HMSAM telah disesuaikan agar relevan dengan konteks penggunaan platform "Starcoder".

Rincian pertanyaan yang diajukan kepada responden dapat dilihat pada Tabel 3.3. Tanda \* pada item menandakan skala dengan nilai terbalik (*reverse scale*).



Tabel 3.3. Kuesioner Evaluasi HMSAM untuk Sistem Starcoder

| Construct                       | Item Pertanyaan   |
|---------------------------------|---|
| Perceived Ease of Use (PEOU)    | <p><b>PEOU1</b> Berinteraksi dengan Starcoder terasa jelas dan mudah dipahami.</p> <p><b>PEOU2</b> Saya tidak perlu berusaha keras secara mental untuk menggunakan Starcoder.</p> <p><b>PEOU3</b> Saya merasa Starcoder berjalan lancar tanpa error atau masalah teknis.</p> <p><b>PEOU4</b> Saya merasa Starcoder responsif terhadap perintah saya.</p> <p><b>PEOU5</b> Sangat mudah bagi saya untuk mempelajari cara mengoperasikan Starcoder.</p> <p><b>PEOU6</b> Sangat mudah untuk menyelesaikan tugas-tugas saya menggunakan Starcoder.</p> <p><b>PEOU7</b> Saya dapat dengan cepat menjadi mahir dalam menggunakan Starcoder.</p> <p><b>PEOU8</b> Secara keseluruhan, saya merasa Starcoder mudah digunakan.</p> |
| Perceived Usefulness (PU)       | <p><b>PU1</b> Starcoder meningkatkan efektivitas belajar pemrograman saya.</p> <p><b>PU2</b> Starcoder membantu saya memanfaatkan waktu belajar saya dengan lebih produktif.</p> <p><b>PU3</b> Starcoder adalah alternatif yang bermanfaat dari metode belajar tradisional yang membosankan.</p> <p><b>PU4</b> Starcoder membantu saya menjernihkan pemahaman saya tentang konsep algoritma.</p> <p><b>PU5</b> Starcoder sangat berguna untuk meningkatkan motivasi belajar saya.</p>   |
| Berlanjut ke halaman berikutnya |   |

Tabel 3.3 Kuesioner Evaluasi HMSAM untuk Sistem Starcoder (lanjutan)

| Construct                       | Item Pertanyaan   |
|---------------------------------|---|
| Curiosity (CUR)                 | <p><b>CUR1</b> Menggunakan Starcoder membangkitkan rasa penasaran saya (untuk belajar lebih).</p> <p><b>CUR2</b> Starcoder membuat saya penasaran tentang apa yang akan terjadi selanjutnya.</p> <p><b>CUR3</b> Starcoder merangsang imajinasi saya.</p>  |
| Joy (JOY)                       | <p><b>JOY1</b> Saya merasa proses belajar di Starcoder itu menyenangkan.</p> <p><b>JOY2</b> Saya mendapatkan kesenangan saat menggunakan Starcoder.</p> <p><b>JOY3*</b> (R) Saya merasa Starcoder itu membosankan.</p> <p><b>JOY4*</b> (R) Saya merasa terganggu (frustrasi) saat menggunakan Starcoder.</p> <p><b>JOY5</b> Pengalaman menggunakan Starcoder secara keseluruhan dapat dinikmati.</p> <p><b>JOY6*</b> (R) Saya merasa tidak puas dengan pengalaman belajar di Starcoder.</p> |
| Berlanjut ke halaman berikutnya |   |

Tabel 3.3 Kuesioner Evaluasi HMSAM untuk Sistem Starcoder (lanjutan)

| Construct                         | Item Pertanyaan   |
|-----------------------------------|---|
| Control (CTL)                     | <p><b>CTL1</b> Saya merasa memiliki banyak kendali atas pengalaman saya di Starcoder.</p> <p><b>CTL2</b> Saya memiliki kebebasan untuk memilih apa yang ingin saya lihat atau lakukan di Starcoder.</p> <p><b>CTL3*</b> (R) Saya merasa hanya memiliki sedikit kendali atas apa yang bisa saya lakukan di Starcoder.</p> <p><b>CTL4</b> Saya merasa memegang kendali penuh atas alur pembelajaran saya sendiri.</p> <p><b>CTL5*</b> (R) Saya merasa tidak memiliki kendali atas bagaimana saya berinteraksi dengan Starcoder.</p> <p><b>CTL6</b> Starcoder memungkinkan saya untuk mengontrol interaksi saya di dalam platform.</p> |
| Behavioral Intention to Use (BIU) | <p><b>BIU1</b> Saya berencana untuk terus menggunakan Starcoder di masa depan.</p> <p><b>BIU2</b> Saya berniat untuk sering menggunakan Starcoder setelah ini.</p> <p><b>BIU3</b> Saya memprediksi bahwa saya akan terus menggunakan Starcoder.</p>   |
| Focused Immersion (FI)            | <p><b>FI1</b> Saya dapat mengabaikan gangguan di sekitar saya saat menggunakan Starcoder.</p> <p><b>FI2</b> Saya merasa sangat terserap (larut) saat menggunakan Starcoder.</p> <p><b>FI3</b> Saya merasa tenggelam dalam pengalaman menggunakan Starcoder.</p> <p><b>FI4*</b> (R) Pikiran saya mudah teralihkan ke hal lain saat saya menggunakan Starcoder.</p> <p><b>FI5</b> Fokus perhatian saya tetap terjaga saat menggunakan Starcoder.</p>  |

### 3.5 Penulisan Laporan

Penyusunan laporan penelitian bertujuan untuk mendokumentasikan keseluruhan proses pada setiap tahapan penelitian yang telah dilakukan. Laporan berisikan hasil analisis, landasan teori yang digunakan, pembahasan hasil penelitian dari rumusan masalah, serta kesimpulan dan saran.

