

Rancang Bangun LMS Mindlab Berbasis Web dengan Online Compiler dan Profile Showcase

Rino Raihan Gumilang¹, Rizky Basatha²

^{1,2} Pendidikan Teknologi Informasi, Universitas Negeri Surabaya
Indonesia

¹rinoraihan.22025@mhs.unesa.ac.id

²rizkybasatha@unesa.ac.id

Abstrak—Penelitian ini bertujuan untuk merancang dan membangun sebuah Learning Management System (LMS) "Mindlab" berbasis website yang dilengkapi dengan fitur Compiler Online dan Profile Showcase untuk mendukung praktik Pemrograman Terstruktur. Terdapat kebutuhan akan platform sistem informasi terpusat yang mampu menyediakan lingkungan eksekusi kode pemrograman secara real-time langsung di dalam peramban, serta fitur dokumentasi portofolio digital yang saling terintegrasi dalam satu sistem, guna mengatasi keterbatasan spesifikasi perangkat keras dan fragmentasi platform. Sistem dibangun menggunakan arsitektur full-stack dengan framework Next.js, basis data Supabase (PostgreSQL), serta integrasi layanan pihak ketiga berupa Piston API dan Wandbox sebagai mesin kompilator kode online. Metode pengembangan sistem menggunakan pendekatan Research and Development (R&D) melalui tahapan siklus perancangan ADDIE (Analyze, Design, Development, Implementation, dan Evaluation). Pengujian fungsionalitas sistem dievaluasi menggunakan metode Black Box Testing yang menghasilkan tingkat validitas sebesar 99,07% dari 216 skenario uji, membuktikan bahwa fungsionalitas sistem berjalan dengan sangat baik dan tanpa kecacatan (bug) mayor. Selain itu, evaluasi non-fungsional dari sisi antarmuka dan pengalaman pengguna diukur menggunakan User Experience Questionnaire (UEQ). Hasil pengukuran UEQ menunjukkan bahwa lima dari enam skala berada pada posisi Above Average terhadap tolok ukur (benchmark) global produk digital. Kesimpulannya, rancang bangun sistem LMS Mindlab terbukti andal secara teknis, fungsional, dan mampu memberikan kualitas pengalaman pengguna yang sangat baik sebagai platform eksekusi dan manajemen pemrograman terstruktur.

Kata Kunci— *Learning Management System, Mindlab, Compiler Online, Next.js, Black Box Testing, User Experience Questionnaire, Full-stack Development.*

I. PENDAHULUAN

Di era transformasi digital, perkembangan teknologi informasi menuntut adanya platform yang fleksibel dan terstruktur untuk menunjang berbagai kebutuhan operasional dan peningkatan kompetensi teknis. Dalam bidang pemrograman terstruktur, proses penulisan, pengujian, dan kompilasi kode sumber seperti bahasa Python secara tradisional membutuhkan instalasi Integrated Development

Environment (IDE) pada perangkat lokal. Ketergantungan pada perangkat lunak eksternal ini sering kali memunculkan kendala aksesibilitas, terutama bagi pengguna akhir (end-user) yang hanya memiliki perangkat dengan spesifikasi perangkat keras yang terbatas [1]. Selain itu, pengelolaan dokumentasi proyek serta manajemen aktivitas pemrograman umumnya masih terfragmentasi di berbagai platform terpisah, yang mengakibatkan tidak efisiennya proses peninjauan dan validasi kode sumber.

Untuk mengatasi permasalahan spesifikasi perangkat keras dan fragmentasi platform tersebut, diperlukan sebuah arsitektur sistem informasi terpadu dalam bentuk Learning Management System (LMS) berbasis web yang dirancang secara spesifik untuk memudahkan alur pemrograman [2], [3]. Penelitian ini mengusulkan rancang bangun sistem LMS bernama "Mindlab", yang berfungsi sebagai ruang digital interaktif untuk melakukan eksekusi kode dan penyimpanan portofolio. Kebaruan utama pada platform ini adalah integrasi fitur Compiler Online yang memungkinkan pengguna untuk menulis, menguji, dan mengeksekusi kode program secara real-time langsung melalui peramban (browser), tanpa memerlukan instalasi aplikasi tambahan apa pun di sisi client.

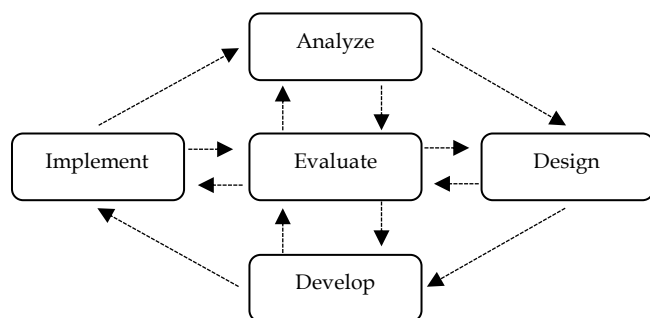
Secara arsitektural, rancang bangun LMS Mindlab dikembangkan menggunakan metode full-stack berbasis framework Next.js dan bahasa pemrograman JavaScript. Pengelolaan basis data relasional, mekanisme autentikasi terpusat, dan integrasi real-time diimplementasikan menggunakan layanan Backend-as-a-Service (BaaS) dari Supabase yang berbasis PostgreSQL. Untuk mengakomodasi eksekusi kode pemrograman di sisi server, platform ini dihubungkan melalui Application Programming Interface (API) dengan layanan Wandbox dan Piston API, yang bertindak sebagai mesin komputasi (code execution engine) untuk memproses output maupun mendeteksi log error (seperti syntax/runtime error).

Sebagai pelengkap fungsi manajemen proyek, sistem ini juga menyediakan modul Profile Showcase yang secara dinamis mengubah rekam jejak penyelesaian proyek komputasi menjadi portofolio digital yang terverifikasi. Seluruh siklus rekayasa perangkat lunak dalam penelitian ini dioperasikan menggunakan metode Research and Development (R&D) [4], [5] yang mengacu pada kerangka kerja ADDIE (Analyze, Design, Development, Implementation, Evaluation) [6]. Pendekatan sistematis ini digunakan guna menjamin kelayakan arsitektur sistem, memvalidasi fungsionalitas melalui pengujian Black Box

Testing, serta mengoptimalkan tingkat usability antarmuka pengguna.

II. METODE PENELITIAN

Metodologi penelitian yang digunakan pada rancang bangun ini adalah Research and Development (R&D) [5] dengan mengadopsi pendekatan siklus pengembangan ADDIE (Analyze, Design, Development, Implementation, dan Evaluation) [7]. Model ini digunakan karena kerangka kerjanya yang sangat terstruktur dan adaptif untuk mengembangkan arsitektur perangkat lunak serta aplikasi digital berbasis web.



Gbr. 1 Model Pengembangan ADDIE sebagai Siklus Pengembangan Perangkat Lunak (Sumber:)

A. Tahap Analisis (Analyze)

Tahap analisis bertujuan untuk mendefinisikan secara akurat spesifikasi teknis dan mengidentifikasi kebutuhan sistem informasi pengguna. Analisis ini memetakan 17 fungsi esensial pada perangkat lunak, yang secara arsitektural mencakup manajemen sistem, lingkungan Compiler Online, publikasi Profile Showcase, dan modul Forum Grup Diskusi terpusat. Dari sisi infrastruktur, analisis non-fungsional menetapkan parameter ketat yang wajib dipenuhi oleh sistem, yakni jaminan kinerja aplikasi (performance), kompatibilitas antar-peramban (cross-browser), responsibilitas antarmuka (usability), skalabilitas basis data (maintainability), dan keamanan server [8].

B. Tahap Perancangan (Design)

Kebutuhan fungsional yang telah diidentifikasi kemudian diterjemahkan ke dalam cetak biru (blueprint) dan rancangan teknis yang komprehensif. Pemodelan arsitektural sistem divisualisasikan dengan standar Unified Modeling Language (UML) melalui kerangka Use Case Diagram, Activity Diagram, dan Sequence Diagram. Pemodelan struktur penyimpanan dan pengorganisasian data relasional disusun melalui Class Diagram. Desain antarmuka pengguna direpresentasikan dalam bentuk purwarupa (wireframe) interaktif menggunakan perangkat lunak Figma [10]. Pada tahap ini juga disusun arsitektur komunikasi data asinkronus berformat JSON, yang dirancang khusus agar perintah eksekusi kode real-time berjalan terisolasi tanpa membebani performa antarmuka utama pengguna.

C. Tahap Pengembangan (Development)

Tahap ini merupakan fase realisasi dari desain arsitektur menuju penulisan kode sumber (coding) yang utuh. Rancang bangun ini direalisasikan dengan pendekatan full-stack menggunakan framework Next.js yang dikembangkan berbasis bahasa pemrograman JavaScript. Pengelolaan basis data relasional serta lapisan keamanan autentikasi diimplementasikan melalui arsitektur Backend-as-a-Service (BaaS) menggunakan layanan Supabase (PostgreSQL). Agar komputasi pemrograman jarak jauh dapat diproses secara online, sistem diintegrasikan dengan layanan pihak ketiga melalui konfigurasi Application Programming Interface (API) menuju endpoint Piston API dan Wandbox, yang beroperasi sebagai mesin komputasi (code execution engine) untuk menangani deteksi galat dan jalannya struktur kode program. Sistem yang sudah fungsional kemudian melalui proses deployment ke platform peladen awan (cloud server) Vercel.

D. Tahap Implementasi (Implementation)

Perangkat lunak yang telah berstatus Beta dan beroperasi penuh secara cloud kemudian diuji coba stabilitasnya pada lingkungan komputasi operasional yang sesungguhnya. Tahap ini sangat krusial untuk memastikan seluruh fungsionalitas sistem, khususnya kelancaran layanan kompilator kode secara real-time dan sinkronisasi data yang cepat ke basis data, dapat menahan beban keterlambatan jaringan (latency) ketika diakses serentak oleh banyak klien (client-side).

E. Tahap Evaluasi (Evaluation)

Stabilitas rancang bangun dievaluasi kelayakannya menggunakan kuesioner dan prosedur pengujian kualitas perangkat lunak secara kuantitatif. Pengujian tersebut terbagi ke dalam dua parameter utama:

1. Pengujian Fungsionalitas: Integritas sistem dan keberadaan cacat kode (bugs) dievaluasi melalui metode Black Box Testing. Pengujian ini dilakukan secara menyeluruh dengan menguji 216 skenario dari proses validasi input/output.
2. Pengujian Non-Fungsionalitas: Pengalaman pengguna (User Experience) dan efektivitas interaksi antarmuka diukur menggunakan kuesioner terstandarisasi berskala internasional, yakni User Experience Questionnaire (UEQ) [11], [12]. Kuesioner ini memetakan evaluasi produk digital melalui enam skala matriks sentral: Attractiveness (Daya Tarik), Perspicuity (Kejelasan), Efficiency (Efisiensi), Dependability (Ketepatan), Stimulation (Stimulasi), dan Novelty (Kebaruan). Data hasil kuesioner ditransformasikan melalui Data Analysis Tool untuk membandingkan matriks kualitas sistem (benchmark) yang dikembangkan terhadap parameter global dari pangkalan data produk digital lainnya.

III. HASIL DAN PEMBAHASAN

Hasil pengembangan Learning Management System (LMS) "Mindlab" berbasis website berfitur Online Compiler dan Profile Showcase ini mengacu pada lima tahapan model ADDIE, dengan penjabaran hasil sebagai berikut.

A. Tahap Analisis(Analyze)

Hasil dari tahap analisis adalah kebutuhan sistem yang dirancang untuk mengatasi permasalahan fragmentasi platform pemrograman, keterbatasan spesifikasi perangkat keras (device) pada sisi client, dan tidak terpusatnya rekam jejak portofolio digital pengguna. Kebutuhan ini dibagi menjadi kebutuhan fungsional dan non-fungsional. Adapun analisis kebutuhan pada Learning Management System yang dikembangkan, sebagai berikut:

1) *Identifikasi Pengguna*: terdapat tiga peran (*role*) pengguna pada sistem, yaitu (a) Administrator, sebagai entitas pengelola sistem utama yang bertugas mengelola basis data akun, memberikan otorisasi akses (*role-based access control*), serta mengonfigurasi *environment* utama pada platform; (b) Peninjau (*Reviewer/Manajer Proyek*), sebagai pengelola repositori studi kasus yang memiliki akses untuk mengunggah spesifikasi proyek atau logika algoritma, serta memberikan validasi dan umpan balik (*feedback*) terhadap keluaran kode program dan (c) Pengguna Akhir (*End-User/Programmer*), sebagai pengguna utama yang memiliki akses tak terbatas pada *workspace* untuk menulis dan mengeksekusi kode melalui *Online Compiler*, mempublikasikan hasil komputasi, hingga mengelola portofolio digital secara terpusat pada halaman *Profile Showcase*.

2) Kebutuhan Fungsional: berdasarkan hasil wawancara dan studi literatur, maka diperoleh hasil analisis kebutuhan fungsional pada Tabel 1 berikut.

Tabel 1 Kebutuhan Fungsional

Kebutuhan Fungsional		
Fungsi	Deskripsi	Aktor
Login dan logout	Pengguna dapat masuk dan keluar dari sistem menggunakan akun terdaftar	Admin, guru, siswa
Dashboard	Melihat ringkasan progres belajar, tugas, materi dan grafik nilai	Admin, guru, siswa
Kelola pengguna	Mengelola data pengguna (tambah, edit, hapus data guru dan siswa).	Admin
Kelola Kursus	Membuat, menyunting, dan mengatur struktur kurikulum serta informasi detail kelas yang akan diajarkan.	Guru
Akses Kursus	Menjelajahi daftar kelas yang tersedia, melihat tujuan pembelajaran, serta masuk ke dalam ruang kelas digital.	Guru, Siswa
Kelola materi	Mengunggah, mengedit, dan menghapus materi pelajaran (modul, video, presentasi) agar terpusat.	Guru

Kebutuhan Fungsional		
Fungsi	Deskripsi	Aktor
Akses materi	Melihat dan mengunduh materi pelajaran yang telah diunggah oleh guru.	Siswa, guru
Kelola Pendaftaran Kelas	Mengatur data siswa yang berhak mengikuti kursus tertentu (<i>enrollment</i>) dan memvalidasi akses peserta didik ke dalam kelas.	Guru
Kelola PBL	Membuat narasi skenario masalah, menentukan parameter masalah, dan mengunggah panduan penyelidikan (<i>Problem Sheet</i>).	Guru
Akses PBL	Mengakses detail skenario masalah, melakukan penyelidikan, dan mengunggah solusi atau hasil analisis masalah.	Siswa, guru
Profile showcase	Mengelola hasil akhir proyek yang telah diverifikasi, mengedit deskripsi atau refleksi hasil portofolio digital.	Siswa
Compiler online	Menulis, mengompilasi, dan menjalankan kode program (Python).	Guru, siswa
Kelola asesmen	Membuat asesmen/penilaian untuk mengevaluasi pengetahuan siswa.	Guru
Akses asesmen	Mengerjakan asesmen yang telah diberikan oleh guru	Siswa, guru
Penilaian	Memberikan nilai pada proyek PjBL.	Guru
Forum Grup Diskusi	Berinteraksi melalui pesan teks untuk mendiskusikan masalah, berbagi ide, dan berkolaborasi dalam mencari solusi.	Guru, siswa
Rekap nilai	Melihat dan mengelola rekapitulasi nilai siswa.	Guru, siswa

3) Kebutuhan Non-Fungsional: analisis kebutuhan non-fungsional dilakukan untuk menetapkan batasan dan standar kualitas sistem agar dapat berjalan secara optimal, dengan merujuk pada standar kualitas perangkat lunak dalam studi literatur terdahulu yang menekankan aspek kinerja dan

kemudahan akses [9]. Hasil analisis kebutuhan non-fungsional ditunjukkan pada Tabel 2 berikut.

Tabel 2 Kebutuhan Non-Fungsional

Kebutuhan Non Fungsional	
Aspek	Deskripsi
Kinerja (performance)	LMS harus mampu menampilkan halaman dan memproses perintah pengguna dengan waktu kurang dari dua menit agar tidak mengganggu proses pembelajaran.
Kompatibilitas (Compatibility)	Sistem dapat diakses dari browser/peramban seperti (google chrome, mozilla firefox, safari, microsoft edge)
Ketergunaan (Usability)	Antarmuka LMS harus mudah dipahami dan digunakan oleh administrator, guru, maupun peserta didik, dan akun hanya dapat dibuat/didaftarkan oleh admin untuk mencegah pendaftaran akun palsu.
Keamanan (Security)	Sistem melindungi data nilai siswa dan mekanisme otentikasi login. Membatasi jenis file yang dapat diunggah, yaitu pdf dan tautan.
Pemeliharaan (Maintainability)	Kode program harus disusun secara terorganisir dengan baik agar dapat dengan mudah diperbaiki dan dikembangkan di masa mendatang.

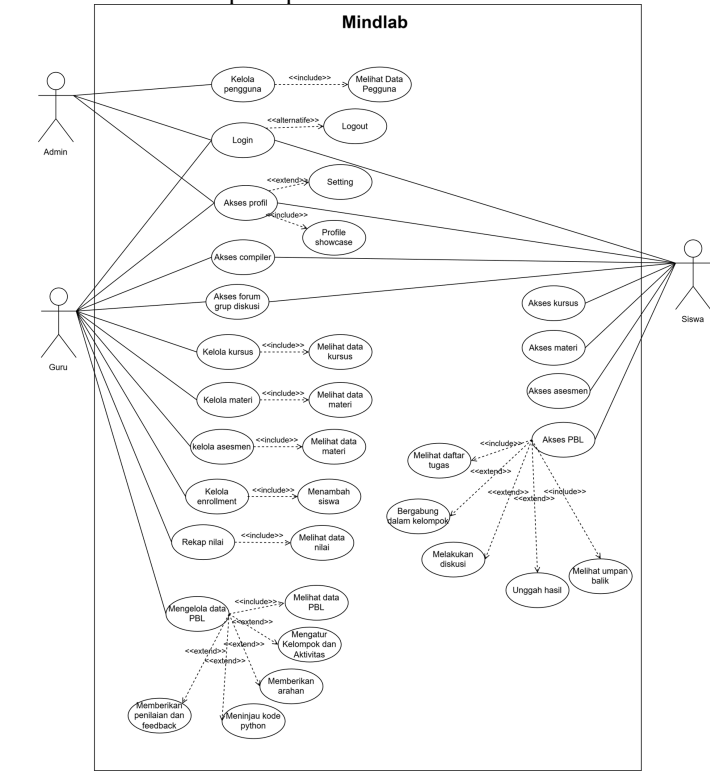
B. Tahap Perancangan (Design)

Berdasarkan hasil analisis kebutuhan, dirancang arsitektur sistem dan alur interaksi pengguna. Arsitektur aplikasi didesain dengan memisahkan antarmuka pengguna (*client*), logika peladen (*server*), dan layanan pihak ketiga, yaitu Supabase untuk manajemen basis data terpusat serta Piston API dan Wandbox untuk mesin kompilasi kode eksternal.

1) Arsitektur Sistem: arsitektur LMS Mindlab yang dikembangkan terdiri atas tiga lapisan utama. Lapisan client berupa browser yang menjalankan frontend Next.js (berbasis React.js dan Tailwind CSS). Antarmuka pada lapisan ini dirancang secara modern dengan mengimplementasikan estetika glassmorphism serta transisi elemen yang halus menggunakan pustaka Framer Motion, guna mengoptimalkan pengalaman interaksi pengguna. Lapisan client juga mengintegrasikan Monaco Code Editor untuk memfasilitasi penulisan kode sintaksis secara langsung, serta merender antarmuka Profile Showcase. Lapisan server berupa Next.js API Routes yang di-hosting pada layanan komputasi awan Vercel. Lapisan ini menangani seluruh logika bisnis full-stack, yang mencakup manajemen repositori studi kasus, validasi parameter keluaran (output), penyimpanan rekam jejak

penyelesaian tugas, hingga sinkronisasi muatan data (payload) menuju Online Compiler. Lapisan layanan pihak ketiga terdiri atas Supabase sebagai Backend-as-a-Service (BaaS) yang menyediakan basis data PostgreSQL dan sistem autentikasi, serta peladen Piston API yang beroperasi sebagai sandbox terisolasi untuk mengeksekusi kode pemrograman.

2) Use Case Diagram: Gambar 3 menggambarkan interaksi antara pengguna (Administrator, Peninjau, Pengguna Akhir) dengan sistem [8]. Diagram ini berfungsi mengidentifikasi fitur-fitur yang dapat diakses oleh setiap peran pengguna dan menjadi acuan dalam pengembangan modul-modul sistem pada penelitian ini.



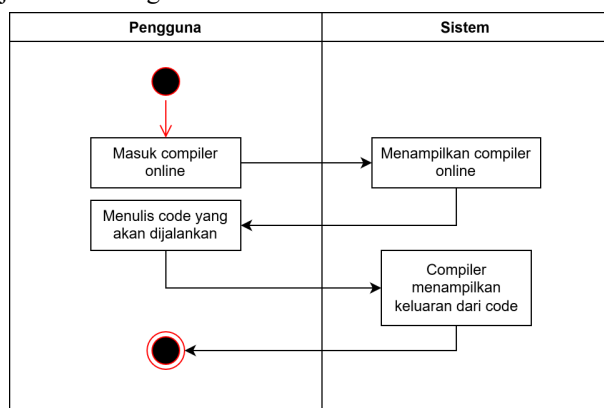
Gbr. 3 Use Case Diagram

Tabel 3 Spesifikasi Use Case

No	Use Case	Aktor	Kondisi Awal	Kondisi Akhir	Success Scenario	Alternative Scenario
1	Login	Admin, Guru, Siswa	Pengguna belum masuk ke LMS	Pengguna berhasil masuk ke LMS dan diarahkan ke dashboard	1) Pengguna masuk ke halaman login 2) Pengguna memasukkan username dan password yang diberikan Admin 3) Sistem melakukan validasi 4) Jika valid, pengguna diarahkan ke dashboard	Jika validasi gagal (username/password salah), sistem menampilkan pesan error.
2	Profile	Siswa	Belum	Profile	1) Siswa	Jika belum ada

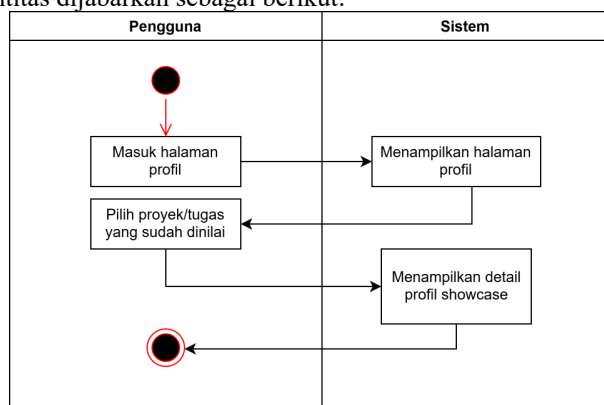
Integrated Development Environment (IDE) Visual Studio Code. Arsitektur sistem dibangun secara full-stack menggunakan framework Next.js. Untuk menjembatani komunikasi data antara logika peladen dan basis data relational PostgreSQL, sistem mengimplementasikan Prisma sebagai Object-Relational Mapping (ORM). Lapisan keamanan (Row Level Security), manajemen penyimpanan (storage), dan autentikasi dikelola melalui layanan Backend-as-a-Service (BaaS) Supabase.

Implementasi Antarmuka Pengguna: Tata letak antarmuka utama dirancang menggunakan pola spasial yang efisien, terbagi menjadi panel navigasi asinkron (sidebar) di sisi kiri dan area workspace dinamis di sisi kanan. Menu navigasi dirender secara kondisional mengikuti otorisasi peran pengguna. Hasil implementasi antarmuka untuk ketiga entitas dijabarkan sebagai berikut:



Gbr. 5 Activity Diagram Python Online Compiler

Gbr 6 Menunjukkan alur kerja dari fitur unggulan yaitu profile showcase, modul ini merupakan inovasi sistem untuk mengatasi fragmentasi dokumentasi hasil rekayasa perangkat lunak pengguna. Profile Showcase bertindak sebagai portofolio digital yang secara dinamis mengekstrak dan mempublikasikan rekam jejak penyelesaian studi kasus yang telah tervalidasi. Hasil implementasi antarmuka untuk ketiga entitas dijabarkan sebagai berikut:



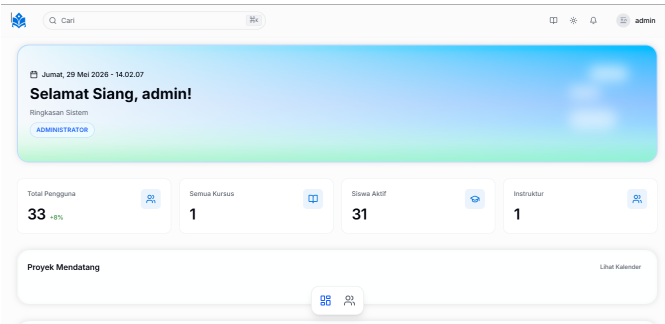
Gbr. 6 Activity Diagram Profile Showcase

1) Antarmuka Administrator: Gambar 7 menunjukkan implementasi antarmuka Administrator yang berfokus pada fungsi kendali pusat dan integritas keamanan platform. Pada

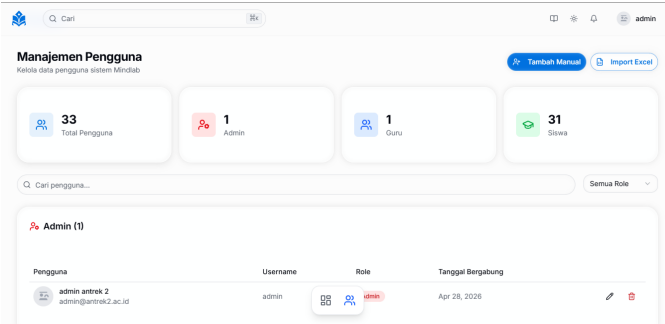
halaman Dashboard (Gambar 7a), sistem menampilkan statistik performa infrastruktur yang meliputi total workspace aktif, jumlah pengguna terdaftar, serta rasio pengguna aktif. Sementara itu, halaman Kelola User (Gambar 7b) memberikan wewenang eksklusif kepada Administrator untuk memanipulasi data akun pengguna melalui eksekusi kendali penuh pada operasi penambahan entitas, pembaruan (edit) data, serta penghapusan kredensial yang tidak aktif guna menjamin isolasi dan keamanan data.

2) Antarmuka Guru: Gambar 8 menunjukkan implementasi antarmuka untuk Guru yang bertindak sebagai Peninjau (Reviewer). Halaman Dashboard (Gambar 8a) memuat agregasi data metrik yang memvisualisasikan ringkasan progres komputasi pengguna, daftar tugas, dan grafik penyelesaian workspace. Untuk manajemen repositori, halaman Kelola Kursus (Gambar 8b) memfasilitasi Peninjau dalam mengonfigurasi struktur lingkungan proyek, merancang spesifikasi masalah teknis (bug/issue), serta menyeleksi otorisasi pengguna, sedangkan halaman Kelola Materi (Gambar 8c) digunakan untuk menyusun referensi teknis berformat dokumen atau tautan sebagai landasan algoritma. Selain itu, Peninjau memiliki akses ke Compiler Online (Gambar 8d) sebagai sandbox terpadu untuk menguji logika Python secara mandiri, serta antarmuka Forum Grup Diskusi (Gambar 8e) untuk melakukan monitoring asinkronus, memberikan scaffolding (bantuan logika), dan mengirimkan log umpan balik (feedback).

3) Antarmuka Siswa: Gambar 9 menunjukkan implementasi antarmuka untuk Siswa yang bertindak sebagai Pengguna Akhir (Programmer/End-User). Halaman Dashboard (Gambar 9a) merender visualisasi tenggat waktu pengiriman komit dan grafik progres kompetensi komputasi, yang didukung oleh halaman Akses Kursus (Gambar 9b) sebagai antarmuka interaktif untuk membaca spesifikasi teknis masalah dan mengunggah source code solusi, serta halaman Akses Materi (Gambar 9c) untuk mengunduh berkas teknis dari peladen. Untuk eksekusi dan penyelesaian masalah, halaman Compiler Online (Gambar 9d) menyediakan sandbox mandiri guna melakukan debugging Python secara real-time langsung di dalam peramban, sementara Forum Grup Diskusi (Gambar 9e) digunakan untuk kolaborasi pembagian peran dan resolusi galat dengan tim. Keseluruhan alur ini bermuara pada halaman Profile Showcase (Gambar 9f), yang bertindak sebagai ruang publikasi otomatis untuk mengekstraksi penyelesaian kode (resolved issue) menjadi portofolio digital yang dapat diverifikasi secara publik.

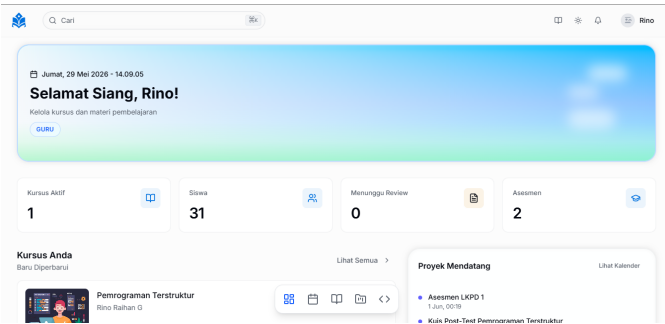


(a)

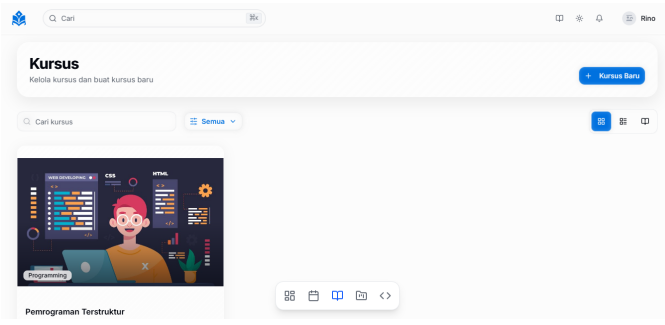


(b)

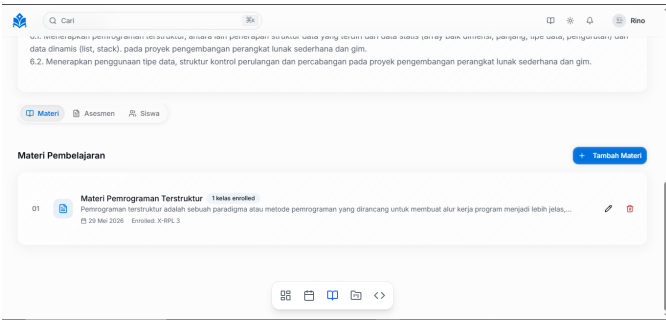
Gbr. 7 Implementasi Antarmuka Administrator: (a) Dashboard, (b) Kelola User



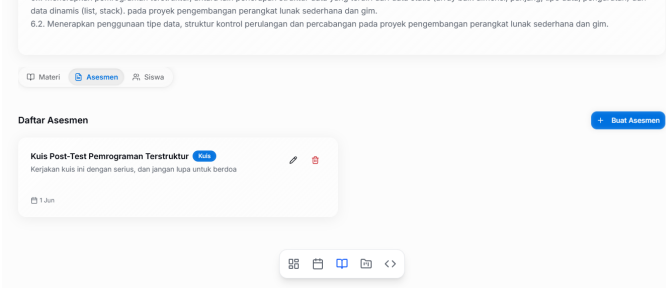
(a)



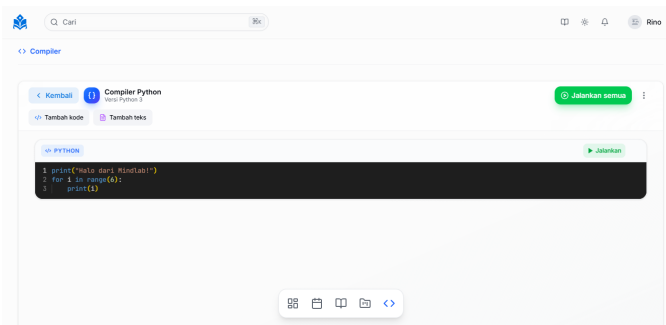
(b)



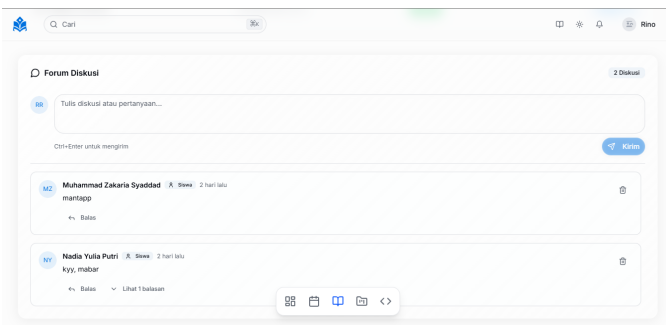
(b)



(c)

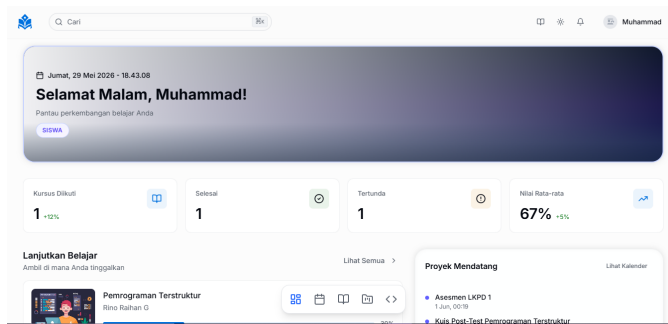


(d)

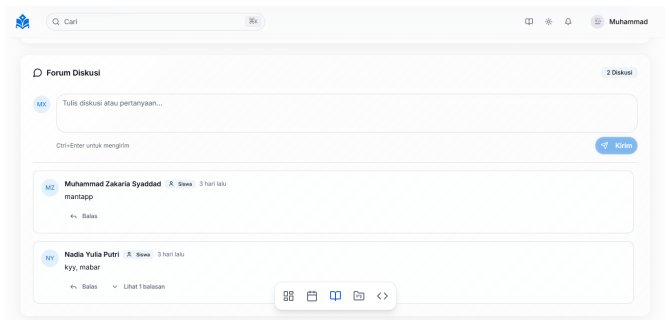


(e)

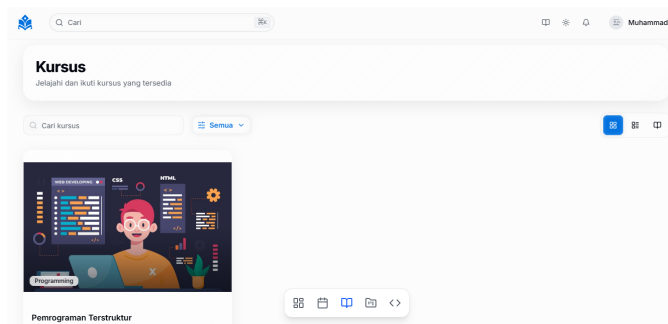
Gbr.8 Implementasi Antarmuka Guru: (a) Dashboard, (b) Kelola Kursus, (c) Kelola Materi, (d) Compiler Online, (e) Forum Grup Diskusi



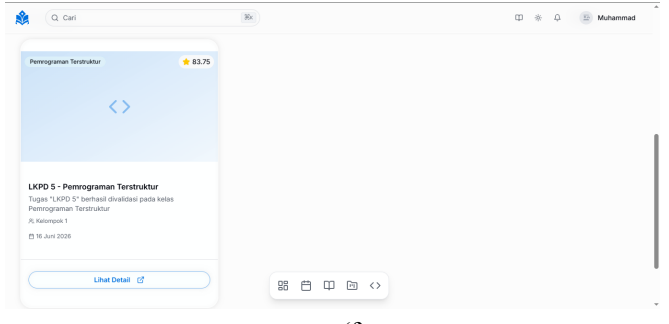
(a)



(e)

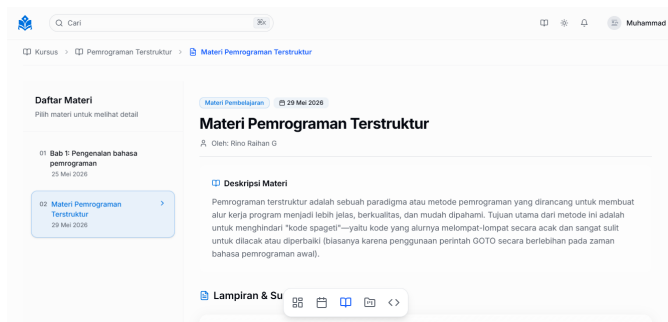


(b)

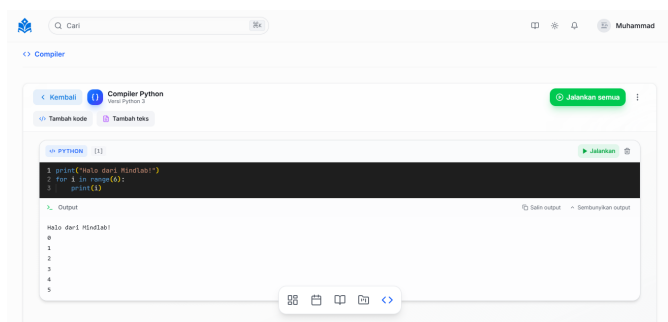


(f)

Gbr. 9 Implementasi Antarmuka Siswa: (a) Dashboard, (b) Akses Kursus, (c) Akses Materi, (d) Compiler Online, (e) Forum Grup Diskusi, (f) Profile Showcase



(c)



(d)

4) **Fitur Unggulan: Compiler Online:** Fitur ini dirancang agar pengguna akhir dapat menulis, menguji, dan mengeksekusi kode Pemrograman Terstruktur (Python) secara langsung melalui sistem di dalam peramban (browser), sebagaimana ditunjukkan pada Gambar 10. Pengguna tidak perlu melakukan konfigurasi instalasi Integrated Development Environment (IDE) tambahan pada perangkat lokal, seperti PyCharm ataupun Visual Studio Code. Antarmuka penulisan kode diimplementasikan dengan memanfaatkan pustaka monaco-editor/react. Pustaka ini beroperasi dengan mesin perender antarmuka yang sama dengan Visual Studio Code, sehingga secara bawaan telah dilengkapi dengan dukungan syntax highlighting spesifik untuk bahasa Python, auto-indentation, serta minimap guna mempermudah navigasi baris kode. Karena fokus penelitian Anda telah diubah ke Pemrograman Terstruktur menggunakan bahasa Python, saya mengadaptasi bagian pendeteksian kode tersebut menjadi "Structure Analyzer" yang mendeteksi konsep pemrograman terstruktur. Seluruh istilah pedagogis (seperti siswa) juga telah diganti dengan terminologi rekayasa perangkat lunak.

Nilai tambah utama dari lingkungan kompilator ini adalah kehadiran fitur Structure Analyzer. Fitur ini berfungsi sebagai asisten otomatis yang memindai struktur sintaksis pengguna akhir secara real-time setiap kali terjadi perubahan karakter di dalam editor (memanfaatkan hook useEffect). Pemindaian ini dieksekusi menggunakan metode Regular Expression (Regex) pada fungsi analyzeCode(). Sistem dirancang untuk mendeteksi penerapan konsep-konsep utama Pemrograman Terstruktur berbasis Python, yaitu:

1. Deklarasi Fungsi, dideteksi dari penggunaan kata kunci def.
2. Struktur Percabangan (Conditionals), dideteksi dari penggunaan kata kunci if, elif, atau else.
3. Struktur Perulangan (Loops), dideteksi dari penggunaan sintaksis for atau while.
4. Operasi Input/Output, dideteksi dari pemanggilan fungsi dasar seperti print() atau input().

Hasil deteksi ini divisualisasikan secara langsung pada panel indikator di sebelah kanan editor. Jika pengguna akhir berhasil mengimplementasikan suatu struktur logika algoritma dengan tepat, indikator pada antarmuka akan secara dinamis berubah warna menjadi hijau sebagai bentuk validasi seketika (real-time validation).

Untuk mengeksekusi kode Python yang telah dirancang, sistem memanfaatkan Application Programming Interface (API) pihak ketiga, yaitu Piston API (dan Wandbox). Antarmuka pemrograman aplikasi ini bertindak sebagai mesin eksekutor komputasi (code execution engine) di dalam lingkungan peladen yang terisolasi (sandbox). Ketika pengguna memicu tombol “Jalankan Kode”, fungsi asinkronus runCode() akan dipanggil.

Sistem kemudian mengemas muatan struktur kode secara utuh ke dalam format JSON yang memuat parameter spesifikasi bahasa (Python) beserta versi kompilatornya lalu mengirimkan muatan (payload) tersebut ke endpoint Piston API. Sistem kemudian mengantrekan proses untuk menunggu respons dari peladen awan dan merender hasil keluarannya secara langsung ke dalam Output Console pada panel kanan bawah. Logika program telah dirancang untuk membedakan jenis respons komputasi yang diterima. Jika API mengembalikan galat sintaksis (Syntax Error) atau kegagalan kompilasi (Runtime Error), teks pada konsol akan dirender dengan warna merah untuk mengindikasikan letak permasalahan kode. Sebaliknya, jika eksekusi program berhasil tanpa hambatan, teks luaran akan ditampilkan dengan warna hijau. Pendekatan antarmuka ini dirancang untuk memfasilitasi proses pelacakan kutu (debugging) secara mandiri layaknya menggunakan perangkat lunak IDE profesional, tanpa memaksa pengguna akhir untuk melakukan konfigurasi instalasi kompilator secara lokal.

5) Modul Profile Showcase: modul ini dirancang untuk mendokumentasikan dan mempublikasikan rekam jejak komputasi (source code) secara sistematis ke dalam bentuk portofolio digital. Setiap pengguna akhir dapat menyimpan riwayat penyelesaian studi kasus pemrograman yang telah dikerjakan, yang meliputi metadata proyek, deskripsi logika pemecahan masalah, kendala (bug) yang dihadapi, serta tautan menuju kode sumber. Peninjau dapat mengakses seluruh rekam jejak penyelesaian algoritma pada workspace yang dikelola melalui halaman monitoring, melakukan tinjauan kode (code review) sebagai umpan balik, dan menandai status repositori tersebut sebagai tervalidasi (resolved). Modul ini secara langsung menjawab permasalahan terfragmentasinya sistem dokumentasi portofolio pemrograman pengguna yang ditemukan pada tahap analisis.

D. Tahap Evaluasi (Evaluation)

Tahap evaluasi dilakukan melalui dua jenis pengujian, yaitu pengujian fungsional menggunakan metode Black Box Testing dan pengujian non-fungsional menggunakan instrumen User Experience Questionnaire (UEQ) untuk mengukur kualitas pengalaman pengguna terhadap LMS Mindlab.

1) Pengujian Fungsional (Black Box Testing)

Pengujian fungsional dilakukan untuk memastikan seluruh fitur LMS Mindlab berjalan sesuai spesifikasi yang telah dirancang, tanpa memperhatikan struktur internal program. Pengujian melibatkan 4 responden yang menguji 54 skenario uji (216 percobaan) mencakup seluruh 17 fungsi utama sistem, mulai dari autentikasi, manajemen pengguna, pembelajaran dan asesmen berbasis PBL, hingga Compiler Online, Forum Grup Diskusi, dan Profile Showcase. Setiap skenario dinyatakan valid apabila hasil eksekusi sesuai dengan hasil yang diharapkan. Ringkasan hasil pengujian per kategori fitur ditunjukkan pada Tabel 4.

Tabel 4 Ringkasan Hasil Pengujian Black Box Testing

Contoh Skenario Pengujian Black Box Testing LMS Mindlab			
Modul / Fitur	Skenario Pengujian (Test Case)	Hasil yang Diharapkan	Hasil
Login (Admin, Guru, Siswa)	Login dengan kredensial valid sesuai peran masing-masing.	Pengguna mengisi email dan password dengan benar sesuai data yang terdaftar dan telah divalidasi admin.	Valid
Kelola Pengguna (Admin)	Tambah pengguna baru (Guru) dengan data lengkap.	Admin mendaftarkan guru secara manual dengan data yang benar dan sesuai ketentuan.	Valid
Profil (Admin, Guru, Siswa)	Pengguna berhasil mengedit data profilnya.	Pengguna mengedit data profil dengan benar dan sesuai ketentuan.	Valid
Pembelajaran & Asesmen (Guru)	Guru dapat menambahkan asesmen baru.	Guru menekan tombol “Tambah Asesmen” dan mengisi semua data dengan benar sesuai ketentuan.	Valid
Compiler Online (Guru)	Guru dapat menggunakan compiler terintegrasi.	Guru menuliskan kode program terstruktur, menekan tombol eksekusi, dan output ditampilkan dengan benar.	Valid
Pembelajaran & Asesmen (Siswa)	Siswa dapat mengerjakan asesmen.	Siswa menekan tombol “Mulai Kerjakan” pada asesmen yang dipilih.	Valid
Compiler Online (Siswa)	Siswa dapat menggunakan compiler untuk praktik mandiri.	Siswa menulis dan mengeksekusi kode program, output ditampilkan secara	Valid

Contoh Skenario Pengujian Black Box Testing LMS Mindlab			
Modul / Fitur	Skenario Pengujian (Test Case)	Hasil yang Diharapkan	Hasil
		akurat di layar.	
Mode Tampilan (Dark/Light)	Pengguna menggunakan fitur dark dan light mode.	Pengguna mengganti tema website menggunakan toggle switcher di navbar.	Valid
Profile Showcase (Siswa)	Siswa dapat melihat profile showcase halaman profil miliknya.	Sistem menampilkan halaman profil portofolio berisi informasi diri, statistik, dan hasil karya/proyek.	Tidak Valid*
Forum Grup Diskusi (Siswa)	Siswa dapat membalas komentar di FGD.	Balasan komentar berhasil muncul secara real-time di bawah komentar yang dituju.	Tidak Valid*

*Tidak valid pada pengujian tahap awal oleh responden 1; telah diperbaiki dan dinyatakan valid pada pengujian responden 2-4.

Berdasarkan Tabel 4, dari 216 percobaan yang dilakukan, 214 percobaan (99,07%) dinyatakan valid dan hanya 2 percobaan (0,93%) yang tidak valid pada pengujian tahap awal, yaitu pada fitur Profile Showcase dan Forum Grup Diskusi Siswa. Ketidakvalidan pada fitur Profile Showcase disebabkan oleh kebijakan akses (bucket policy) pada storage database yang menghalangi siswa mengunggah berkas gambar portofolio akibat Row Level Security (RLS), sedangkan pada fitur Forum Grup Diskusi disebabkan balasan komentar yang tidak muncul secara langsung karena web socket (real-time subscription) belum tersinkronisasi. Kedua permasalahan tersebut telah diperbaiki dan dinyatakan valid pada pengujian ulang oleh responden berikutnya, sehingga secara keseluruhan fungsionalitas LMS Mindlab dinyatakan berjalan sangat baik dan sesuai dengan rancangan.

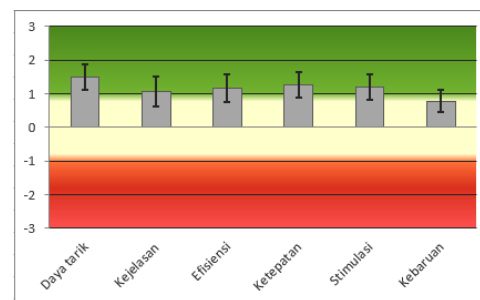
2) Pengujian Non-Fungsional (User Experience Questionnaire/UEQ)

Pengujian non-fungsional dilakukan untuk mengukur kualitas pengalaman pengguna (user experience) terhadap LMS Mindlab menggunakan instrumen UEQ yang melibatkan 31 responden siswa kelas X RPL SMK Antartika 2 Sidoarjo. Instrumen UEQ terdiri atas 26 pasang kata sifat berlawanan yang diukur pada skala 1-7, kemudian ditransformasikan ke rentang nilai -3 (sangat buruk) hingga +3 (sangat baik) dan dikelompokkan ke dalam enam skala kualitas, yaitu Daya Tarik (Attractiveness), Kejelasan (Perspicuity), Efisiensi (Efficiency), Ketepatan (Dependability), Stimulasi (Stimulation), dan Kebaruan (Novelty). Data hasil kuesioner diolah menggunakan UEQ Data Analysis Tool untuk memperoleh nilai rerata per skala serta posisi benchmark terhadap produk digital lain secara global. Hasil pengujian ditunjukkan pada Tabel 5 dan Gbr. 10.

Tabel 5 Hasil Pengujian User Experience Questionnaire (UEQ)

Hasil Pengujian User Experience Questionnaire (UEQ)		
Skala UEQ	Rerata	Kategori Benchmark

Hasil Pengujian User Experience Questionnaire (UEQ)		
Skala UEQ	Rerata	Kategori Benchmark
Daya Tarik (Attractiveness)	1,4892	Above Average
Kejelasan (Perspicuity)	1,0565	Below Average
Efisiensi (Efficiency)	1,1613	Above Average
Ketepatan (Dependability)	1,2661	Above Average
Stimulasi (Stimulation)	1,2016	Above Average
Kebaruan (Novelty)	0,7823	Above Average



Gbr. 10 Grafik rerata skor enam skala UEQ terhadap zona benchmark (hijau: positif, kuning: netral, merah: negatif)

Berdasarkan Tabel 5 dan Gbr. 10, lima dari enam skala UEQ (Daya Tarik, Efisiensi, Ketepatan, Stimulasi, dan Kebaruan) berada pada kategori Above Average, yang berarti kelima skala tersebut dinilai lebih baik dibandingkan mayoritas produk pada dataset benchmark global [12]. Satu skala, yaitu Kejelasan, berada pada kategori Below Average meskipun nilai reratanya tetap positif (1,0565), yang mengindikasikan sebagian siswa masih menghadapi tantangan dalam memahami alur penggunaan sistem pada tahap awal interaksi, kemungkinan dipengaruhi oleh keterbatasan waktu implementasi yang hanya berlangsung dalam tiga pertemuan. Secara keseluruhan, hasil pengujian UEQ menunjukkan bahwa LMS Mindlab memiliki kualitas pengalaman pengguna yang baik dan layak digunakan sebagai media pembelajaran, dengan ruang perbaikan utama pada aspek kemudahan pemahaman sistem (kejelasan) bagi pengguna baru.

E. Pembahasan

Hasil pengembangan sistem menunjukkan bahwa arsitektur full-stack berbasis Next.js memungkinkan integrasi sisi frontend dan logika backend (API Routes) di dalam satu basis kode (codebase). Hal ini secara signifikan mempercepat siklus pengembangan perangkat lunak dan meminimalisasi kompleksitas konfigurasi peladen yang terpisah. Selain itu, pemanfaatan Supabase sebagai Backend-as-a-Service (BaaS) dan Prisma sebagai Object-Relational Mapping (ORM) terbukti sangat efektif dalam menangani autentikasi terpusat, Row Level Security (RLS), dan operasi basis data relasional, tanpa perlu membangun infrastruktur peladen dasar dari awal. Kebutuhan non-fungsional terkait keamanan sistem juga

terbukti valid berdasarkan pengujian parameter batas keamanan (boundary testing).

Pemanfaatan layanan eksternal Piston API dan Wandbox sebagai mesin komputasi (code execution engine) berhasil menjawab permasalahan aksesibilitas dan spesifikasi perangkat keras pada sisi pengguna (client-side) yang diidentifikasi pada tahap analisis. Dengan memindahkan beban komputasi pengeksekusian logika pemrograman Terstruktur (Python) murni ke peladen awan, pengguna akhir dapat menjalankan source code secara real-time langsung melalui peramban. Pendekatan ini sepenuhnya mengeliminasi kebutuhan instalasi Integrated Development Environment (IDE) berat di perangkat lokal, seperti PyCharm atau Visual Studio Code. Hal ini berbeda dengan platform sistem informasi konvensional yang sebelumnya masih terfragmentasi dan mengharuskan pengguna bergantung pada lingkungan komputasi eksternal untuk melakukan kompilasi [13].

Sementara itu, implementasi penganalisis struktur algoritma (Structure Analyzer) di dalam editor dikembangkan menggunakan pendekatan Regular Expression (Regex) karena beban komputasinya yang ringan, sehingga dapat beroperasi secara asinkronus dan real-time di sisi client tanpa membebani performa peladen utama. Namun, pendekatan analitik leksikal ini memiliki keterbatasan karena belum memvalidasi logika kode secara semantik murni. Akibatnya, deteksi berpotensi menjadi kurang akurat apabila terdapat penulisan format kode yang tidak standar. Di sisi lain, integrasi rekam jejak penyelesaian studi kasus menjadi fitur Profile Showcase terbukti memecahkan masalah fragmentasi dokumentasi dengan mengotomatisasi manajemen portofolio digital yang dapat diverifikasi publik.

Secara keseluruhan, evaluasi fungsional menggunakan metode Black Box Testing mencatatkan tingkat kevalidan sebesar 99,07% dari 216 skenario uji. Angka ini menegaskan bahwa arsitektur full-stack Next.js, manajemen basis data Supabase, interaksi Application Programming Interface (API), serta fitur Compiler Online dan Profile Showcase telah berfungsi presisi sesuai rancangan kebutuhan sistem. Selain itu, evaluasi non-fungsional menggunakan User Experience Questionnaire (UEQ) memvalidasi bahwa lima skala matriks usability antarmuka sistem berada pada kategori Above Average terhadap tolok ukur perangkat lunak global.

Hasil pengujian Black Box Testing pada awal iterasi juga mendokumentasikan anomali minor pada fungsionalitas sistem, yakni kendala sinkronisasi balasan pada fitur forum asinkronus (Websocket) dan pemblokiran unggahan muatan (payload) ukuran besar oleh kebijakan keamanan RLS pada storage Supabase. Meskipun permasalahan tersebut berhasil diatasi melalui optimasi komponen event listener dan penyesuaian parameter bucket policy, temuan ini menjadi catatan teknis yang penting mengenai sensitivitas arsitektur real-time terhadap latensi jaringan. Kendala tersebut telah terisolasi dengan baik dan sama sekali tidak mengurangi tingkat kevalidan operasional fungsionalitas sistem secara keseluruhan (99,07%).

IV. KESIMPULAN

Berdasarkan hasil rancang bangun dan pengujian yang telah dilakukan terhadap Learning Management System (LMS) “Mindlab” berbasis website dengan fitur Online Compiler dan Profile Showcase, dapat disimpulkan sebagai berikut. Pertama, tahap analisis kebutuhan berhasil mengidentifikasi tiga peran pengguna (Administrator, Peninjau, dan Pengguna Akhir), 17 kebutuhan fungsional, serta lima aspek kebutuhan non-fungsional yang menjadi acuan perancangan sistem untuk mengatasi permasalahan fragmentasi platform pemrograman dan keterbatasan spesifikasi perangkat keras di sisi client. Kedua, perancangan sistem yang dituangkan ke dalam pemodelan UML, wireframe antarmuka, dan basis data PostgreSQL pada Supabase berhasil direalisasikan menjadi sistem fungsional menggunakan arsitektur full-stack Next.js, dengan integrasi Piston API dan Wandbox sebagai mesin eksekusi kode secara real-time langsung melalui peramban, sehingga menghilangkan kebutuhan instalasi Integrated Development Environment (IDE) di perangkat lokal.

Ketiga, hasil pengujian fungsional dengan metode Black Box Testing pada 54 skenario (216 percobaan) menghasilkan tingkat kevalidan sebesar 99,07% (214 dari 216 percobaan), dengan dua ketidaksesuaian awal pada fitur Profile Showcase dan Forum Grup Diskusi yang disebabkan oleh kebijakan Row Level Security pada storage dan keterlambatan sinkronisasi websocket, dan keduanya telah berhasil diperbaiki. Keempat, hasil pengujian non-fungsional menggunakan User Experience Questionnaire (UEQ) terhadap 31 siswa menunjukkan lima dari enam skala (Daya Tarik, Efisiensi, Ketepatan, Stimulasi, dan Kebaruan) berada pada kategori Above Average dibandingkan tolok ukur global, sedangkan satu skala (Kejelasan) berada pada kategori Below Average meskipun nilai reratanya tetap positif. Secara keseluruhan, LMS Mindlab terbukti andal secara teknis dan fungsional serta mampu memberikan kualitas pengalaman pengguna yang baik sebagai platform eksekusi dan manajemen pemrograman terstruktur berbasis web.

V. SARAN

Berdasarkan hasil penelitian dan kendala yang ditemukan, terdapat beberapa saran yang dapat menjadi bahan pertimbangan untuk pengembangan lebih lanjut. Pertama, mekanisme Structure Analyzer yang saat ini menggunakan pendekatan Regular Expression disarankan untuk dikembangkan menuju analisis berbasis Abstract Syntax Tree (AST) atau pendekatan semantik agar deteksi struktur algoritma menjadi lebih akurat, khususnya pada penulisan kode yang tidak standar. Kedua, mengingat skala Kejelasan (Perspicuity) pada pengujian UEQ berada pada kategori Below Average, pengembangan selanjutnya disarankan untuk menambahkan panduan penggunaan (onboarding guide), tooltip, atau tur interaktif pada antarmuka agar pengguna baru lebih mudah memahami alur penggunaan sistem.

Ketiga, mekanisme sinkronisasi real-time pada Forum Grup Diskusi perlu diperkuat dengan penanganan reconnect dan retry otomatis untuk mengantisipasi latensi jaringan, serta kebijakan bucket policy pada penyimpanan berkas perlu

ditinjau ulang agar tidak menghambat proses unggah data pengguna. Keempat, cakupan bahasa pemrograman pada fitur Compiler Online disarankan untuk diperluas selain Python, memanfaatkan kapabilitas multi-bahasa yang telah disediakan oleh Piston API dan Wandbox. Kelima, pengujian fungsional dan non-fungsional pada penelitian selanjutnya disarankan melibatkan jumlah responden dan variasi institusi pendidikan yang lebih luas agar hasil evaluasi memiliki tingkat generalisasi yang lebih tinggi. Terakhir, bagi institusi pendidikan, LMS Mindlab dapat dijadikan referensi model implementasi platform pembelajaran berbasis web yang terintegrasi dengan fitur praktik pemrograman mandiri dan portofolio digital.

REFERENSI

- [1] A. Z. Putra, A. M. Simarmata, A. M. Husein, and M. Harahap, "Basic pemrograman multi platform dan penerapan inovasi berbasis teknologi," *Dst Dedikasi Sains dan Teknologi*, vol. 4, no. 1, pp. 41–45, 2024, doi: 10.47709/dst.v4i1.4016.
- [2] A. S. Panyili, Sukardi, and E. Alfonsius, "Pengembangan Learning Management System (LMS) berbasis website untuk meningkatkan efektivitas dan efisiensi pembelajaran," *Riau Jurnal Teknik Informatika*, vol. 4, no. 1, pp. 24–34, 2024, doi: 10.30606/rjti.v4i1.3254.
- [3] S. P. Saragih and M. Silalahi, "Pengembangan Learning Management System berbasis Web menggunakan konsep MOOC," *Jurnal Desain dan Analisis Teknologi*, vol. 3, no. 1, pp. 15–21, 2024, doi: 10.58520/jddat.v3i1.42.
- [4] Okpatrioka, "Research and Development (R&D): Penelitian yang inovatif dalam pendidikan," *Nusantara: Jurnal Pendidikan, Bahasa dan Budaya*, vol. 1, no. 1, pp. 86–100, 2023, doi: 10.47861/jdan.v1i1.154.
- [5] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*, 25th ed. Bandung: Alfabeta, 2021.
- [6] A. Rustandi and Rismayanti, "Penerapan model ADDIE dalam pengembangan media pembelajaran di SMPN 22 Kota Samarinda," *FASILKOM (Teknologi Informasi dan Ilmu Komputer)*, vol. 11, no. 2, pp. 57–60, 2021, doi: 10.37859/jf.v11i2.2546.
- [7] M. Safitri and M. R. Azis, "ADDIE, sebuah model untuk pengembangan multimedia," *Jurnal Pendidikan Dasar*, vol. 3, no. 2, pp. 50–58, 2022.
- [8] N. K. Putrianto, Y. M. Widyastuti, and T. Oktiarso, "Perencanaan dan analisis kebutuhan pengguna dalam pengembangan sistem informasi hubungan pelanggan," *Kurawal - Jurnal Teknologi, Informasi dan Industri*, vol. 7, no. 1, pp. 1–11, 2024, doi: 10.33479/kurawal.v7i1.1064.
- [9] E. M. S. Sakti, A. Soderi, and K. Diantoro, "Analisa kebutuhan fungsional dalam perencanaan SIM Bank Sampah studi kasus Bank Sampah Annisa," *IKRA-ITH Informatika: Jurnal Komputer dan Informatika*, vol. 9, no. 3, pp. 71–77, 2025, doi: 10.37817/ikraith-informatika.v9i3.5679.
- [10] E. Nurlailah and K. R. Nova Wardani, "Perancangan website sebagai media informasi dan promosi oleh-oleh khas Kota Pagaralam," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 4, pp. 1175–1185, 2023, doi: 10.29100/jupi.v8i4.
- [11] M. F. R. Alfarizi, P. D. H. Kuncoro, and S. I. Nurfaldini, "Analisis user experience pada aplikasi mobile SIA Universitas Teknologi Yogyakarta dengan metode User Experience Questionnaire," *Teknomatika: Jurnal Informatika dan Komputer*, vol. 17, no. 1, pp. 48–55, 2024, doi: 10.30989/teknomatika.v17i1.1288.
- [12] I. D. Sabukunze and A. Arakaza, "User experience analysis on mobile application design using User Experience Questionnaire," *Indonesian Journal of Information Systems*, vol. 4, no. 1, pp. 15–26, 2021, doi: 10.24002/ijis.v4i1.4646.
- [13] P. M. Silaban, Raudhah, and R. Wanto, "Perancangan aplikasi pembelajaran berbasis web (Studi kasus: SMP Cahaya Pengharapan Abadi)," *Jurnal Armada Informatika*, vol. 7, no. 2, 2021, doi: 10.36520/jai.v7i2.73.