

PENGEMBANGAN WEBSITE PENILAIAN KODE PROGRAM SECARA OTOMATIS BERBASIS *ONLINE JUDGE SYSTEM*

Proposal Penelitian

Diajukan untuk melengkapi tugas-tugas
dan memenuhi syarat-syarat guna memperoleh gelar Sarjana Ilmu Komputer

Oleh:

FAJRY ARIANSYAH
2108107010059



**PROGRAM STUDI INFORMATIKA DEPARTEMEN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA, BANDA ACEH
MEI, 2025**

PENGESAHAN

PENGEMBANGAN WEBSITE PENILAIAN KODE PROGRAM SECARA OTOMATIS BERBASIS *ONLINE JUDGE SYSTEM*

Oleh:

Nama : Fajry Ariansyah
NPM : 2108107010059
Jurusan : Departemen Informatika

Menyetujui:

Pembimbing I

Pembimbing II

Kurnia Saputra, S.T., M.Sc.
p NIP. 198003262014041001

Nazaruddin, S.Si, M.Eng.Sc.
NIP. 197202061997021001

Mengetahui:

Koordinator Prodi S1 Informatika FMIPA
Universitas Syiah Kuala,

Alim Misbullah, S.Si., M.S.
NIP. 198806032019031011

KATA PENGANTAR

Puji dan syukur kami panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya memberikan banyak kesempatan, sehingga penulis dapat menyelesaikan Proposal Penelitian yang berjudul **PENGEMBANGAN WEBSITE PENILAIAN KODE PROGRAM SECARA OTOMATIS BERBASIS ONLINE JUDGE SYSTEM**

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Informatika di Universitas Syiah Kuala. Penelitian ini **dilatarbelakangi** oleh kebutuhan untuk meningkatkan kualitas pembelajaran pemrograman melalui pengembangan suatu platform yang mengintegrasikan latihan soal, kontes pemrograman, dan pengelolaan kelas daring. Harapan saya, hasil penelitian ini dapat memberikan kontribusi positif dalam dunia pendidikan, khususnya di bidang pemrograman.

Saya menyadari bahwa penyelesaian skripsi ini tidak akan terwujud tanpa dukungan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu, izinkan saya mengucapkan terima kasih kepada:

1. Kedua orang tua penulis yang senantiasa selalu mendukung aktivitas dan kegiatan yang penulis lakukan baik secara moral maupun material serta menjadi motivasi terbesar bagi penulis untuk menyelesaikan Tugas Akhir ini.
2. Ibu Rini Deviani, selaku dosen wali, atas bimbingan dan arahan yang telah diberikan semenjak semester satu.
3. Bapak Kurnia Saputra, S.T., M.Sc., selaku dosen pembimbing pertama, atas bimbingan, arahan, dan motivasi yang telah diberikan selama proses penyusunan skripsi ini.
4. Bapak Nazaruddin, S.Si, M.Eng.Sc., selaku dosen pembimbing kedua, atas bimbingan, arahan, dan motivasi yang telah diberikan selama proses penyusunan skripsi ini.
5. Seluruh dosen dan staf di Program Studi Informatika Universitas Syiah Kuala yang telah memberikan ilmu dan inspirasi.
6. Rekan-rekan seperjuangan saya baik dari Informatika FMIPA USK maupun yang tidak, yang telah memberikan dukungan serta masukan yang sangat berharga selama penelitian ini berlangsung.

7. Keluarga tercinta yang selalu memberikan dukungan moral dan material, serta doa agar saya dapat menyelesaikan studi ini dengan baik.

Demikian Tugas Akhir ini penulis buat dengan sebaik-baiknya. Namun penulis menyadari bahwa dalam penulisan Tugas Akhir ini tidaklah sempurna. Oleh karena itu, penulis mengharapkan kritikan dan saran yang bersifat membangun dari pembaca yang bertujuan agar penulis dapat memperbaiki kekurangan tersebut ke depannya. Penulis berharap Tugas Akhir ini dapat bermanfaat bagi banyak pihak dan berguna bagi perkembangan dunia ilmu pengetahuan.

Banda Aceh, 13 Juni 2025

Fajry Ariansyah
NPM. 2108107010059

DAFTAR ISI

	<i>Halaman</i>
Halaman Judul.....	i
Halaman Pengesahan	ii
Kata Pengantar	ii
Daftar Isi	v
Daftar Tabel	vii
Daftar Gambar	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
BAB II TINJAUAN KEPUSTAKAAN.....	5
2.1 Praktikum.....	5
2.2 Pemrograman	5
2.3 Struktur Data dan Algoritma	5
2.4 Online Judge	6
2.5 Next.JS	6
2.6 Tailwind CSS	7
2.7 Application Programming Interface (API)	7
2.8 REST (Representational State Transfer)	8
2.9 FastAPI.....	8
2.10 MySQL	8
2.11 Entity Relational Diagram (ERD)	9
2.12 Rapid Application Development	10
2.13 Pengujian	10
2.13.1 Load Testing	10
2.13.2 System Usability Scale (SUS)	11
2.14 Penelitian Terkait	11
BAB III METODOLOGI PENELITIAN	14
3.1 Waktu dan Lokasi Penelitian	14
3.2 Alat dan Bahan	14
3.2.1 Perangkat Keras (Hardware)	14
3.2.2 Perangkat Lunak (Software)	14
3.3 Metode Penelitian	15
3.3.1 Identifikasi Masalah	15
3.3.2 Studi Literatur	17
3.3.3 Analisis Kebutuhan	18
3.3.4 Perancangan Sistem	20
3.3.5 Pengembangan Sistem	28
3.3.6 Pengujian dan Evaluasi	29

3.3.6.1	Load Testing	29
3.3.6.2	System Usability Scale (SUS)	29
DAFTAR KEPUSTAKAAN.....		32

DAFTAR TABEL

	<i>Halaman</i>
Tabel 2.1	Daftar Pertanyaan <i>System Usability Scale</i> 11
Tabel 2.2	Ringkasan Literatur Terkait Sistem Online Judge 13
Tabel 3.1	Jadwal Pelaksanaan Penelitian 14
Tabel 3.2	Pertanyaan SUS untuk Sistem Penilaian Otomatis 30
Tabel 3.3	Interpretasi Skor SUS 31

DAFTAR GAMBAR

	<i>Halaman</i>
Gambar 3.1 Diagram Alur Penelitian.....	15
Gambar 3.2 Task Asisten Tanpa Bantuan Sistem Penilaian	17
Gambar 3.3 Arsitektur Platform Penilaian Tugas Pemrograman.....	21
Gambar 3.4 Tampilan Halaman Login	22
Gambar 3.5 Dashboard Praktikan	22
Gambar 3.6 Dashboard Admin	23
Gambar 3.7 Form Pembuatan Soal oleh Asisten	23
Gambar 3.8 Daftar Soal	24
Gambar 3.9 Pembuatan Tugas dari Soal yang Tersedia	24
Gambar 3.10 Daftar Tugas yang Dikelola oleh Asisten.....	25
Gambar 3.11 Halaman Tugas Praktikan	25
Gambar 3.12 Hasil Penilaian Otomatis	26
Gambar 3.13 Daftar Kelas Praktikan	26
Gambar 3.14 Kelas yang Diasuh oleh Asisten	27
Gambar 3.15 Rekap Nilai Berdasarkan Tugas	27
Gambar 3.16 Rekap Nilai Satu Kelas	28

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Mata kuliah Pemrograman adalah mata kuliah dasar bagi mahasiswa informatika untuk memahami bagaimana cara memberikan instruksi kepada komputer menggunakan bahasa yang dipahami oleh komputer seperti C, Python, atau Java. Di Program Studi Informatika FMIPA Universitas Syiah Kuala, mata kuliah ini merupakan salah satu fondasi awal mahasiswa dalam membangun kemampuan berpikir komputasional. Dalam mata kuliah ini, mahasiswa mempelajari konsep dasar seperti penggunaan sintaks, variabel, tipe data, operasi matematika, percabangan, perulangan dan lainnya. Pemahaman ini sangat penting untuk mahasiswa agar dapat menyusun kode program yang dapat menyelesaikan berbagai permasalahan. Selain itu, terdapat mata kuliah Struktur Data dan Algoritma, di mana mahasiswa belajar konsep pemrograman lanjutan seperti *array*, *linked list*, *stack*, *queue*, *tree*, hingga *graph*. Dalam mata kuliah ini mahasiswa juga mempelajari algoritma penting seperti pengurutan (*sorting*), pencarian (*searching*), dan rekursi. Tujuan mata kuliah ini adalah untuk melatih cara berpikir yang logis dan terstruktur agar mahasiswa tidak hanya mampu membuat program yang benar, tetapi juga program yang efisien dan optimal.

Untuk membantu pembelajaran mata kuliah tersebut, praktikum menjadi aspek penting yang menghubungkan pemahaman dasar yang telah dipelajari dengan menerapkannya secara langsung. Melalui praktikum, mahasiswa dilatih menerjemahkan teori ke dalam kode program untuk menyelesaikan berbagai persoalan, mulai dari fungsi matematis sederhana hingga membangun algoritma yang lebih kompleks (Thuné & Eckerdal, 2018). Pemberian tugas dalam praktikum juga dirancang bukan sekadar latihan, melainkan sebagai sarana membangun pola pikir pemrograman yang lebih lanjut. Dengan itu, mahasiswa belajar menghadapi berbagai *error*, baik sintaks maupun logika, lalu mencari cara penyelesaiannya. Proses ini juga disebut *debugging*, salah satu kunci yang membantu siswa mengembangkan kemampuannya melalui pemikiran sistematis, analisis, dan pemecahan masalah (Kafai et al., 2020). Dengan memahami konsep, praktik langsung, dan tugas yang menantang, mahasiswa diharapkan mampu membangun pondasi pemrograman yang kuat, melatih logika pemecahan masalah, dan siap menghadapi tantangan lebih besar di dunia *programming*.

Namun, proses penilaian tugas pemrograman secara manual ternyata memiliki tantangan besar, terutama bagi asisten praktikum. Asisten harus mengunduh *file* kode program setiap mahasiswa satu per satu, lalu kemudian menjalankannya, mencoba dengan berbagai kasus uji (*test case*) untuk menguji apakah kode berjalan seperti yang

diharapkan atau tidak, dan terakhir memberikan dan merekap nilai. Proses ini tidak hanya memakan waktu, tetapi juga terkadang rentan terhadap kesalahan penilaian, terutama ketika jumlah praktikan mencapai ratusan. Berdasarkan data Program Studi Informatika Universitas Syiah Kuala pada semester ganjil 2024/2025, terdapat 106 mahasiswa yang terbagi ke dalam 3 kelas (Universitas Syiah Kuala, 2024), sehingga jika setiap asisten harus melayani sekitar 30-46 mahasiswa per kelas, beban penilaian manual akan terasa berat.

Di sisi lain, mahasiswa tidak mendapat umpan balik terkait kesalahan kode mereka, baik dari segi sintaks, logika, maupun ketidaksesuaian *output* dengan *test case* yang diharapkan. Tanpa informasi spesifik tentang letak kesalahan, mahasiswa kesulitan melakukan evaluasi mandiri. Hal ini membuat proses perbaikan kode menjadi lambat dan tidak terarah. Sebagai contoh, seorang mahasiswa mungkin membuat kode program sebuah kalkulator, jika dilihat sekilas, mungkin kodenya berjalan tanpa ada masalah, tetapi ternyata memiliki sebuah *bug*, dimana hasil pembagian dua angka selalu salah atau *error* karena ternyata, ia lupa menangani kasus ketika pembagiannya adalah 0. Jika tidak ada umpan balik langsung, mahasiswa mungkin kesulitan atau bahkan tidak menyadari kesalahan logika ini dan harus mencari penyebabnya secara manual, yang dapat menghambat perkembangan keterampilan pemrogramannya.

Untuk mengatasi tantangan penilaian tugas secara manual tersebut, maka muncul solusi dengan mengotomatisasi proses tersebut agar lebih cepat, akurat, dan efisien. Salah satu pendekatan yang umum digunakan untuk otomatisasi penilaian adalah sistem *online judge* (Ramirez-Echeverry et al., 2018), yaitu sistem yang dapat menilai kode secara otomatis berdasarkan serangkaian *test case* yang ditetapkan. Sistem akan menjalankan kode program yang diberikan lalu melakukan cek dengan setiap *test case* yang diberikan. Bila *test case* gagal maka berarti kode tersebut gagal memenuhi spesifikasi yang ditentukan. Dengan menggunakan sistem tersebut, asisten tidak perlu lagi memeriksa kode satu per satu secara manual, karena sistem akan secara otomatis mengompilasi, menjalankan, dan menguji kode mahasiswa dengan berbagai *test case* yang ditetapkan. Selain itu, mahasiswa juga dapat langsung mengetahui apakah kode mereka sudah benar atau masih memiliki kesalahan berdasarkan *test case* yang diberikan, sehingga mereka bisa lebih cepat memperbaiki dan memahami kekurangan dalam solusi yang dibuat.

Salah satu contoh *online judge* yang sudah banyak digunakan adalah DOM Judge. DOM Judge sering digunakan dalam kompetisi pemrograman seperti ICPC (*International Collegiate Programming Contest*) karena mampu menangani penilaian kode secara cepat dan mendukung banyak bahasa pemrograman. Terdapat juga platform yang mengadopsi sistem *online judge* seperti HackerRank dan LeetCode, keduanya lebih

difokuskan pada latihan pemrograman dan persiapan wawancara kerja, disertai dengan fitur seperti **leaderboard** dan pembahasan solusi. Namun, sistem-sistem ini dirancang untuk kompetisi atau latihan mandiri, tidak menyediakan fitur yang dibutuhkan dalam konteks pembelajaran perkuliahan seperti fitur yang tidak hanya menilai kode, tetapi juga dapat mencatat mahasiswa, membagi kelas, mengelola tugas dan materi, serta menyediakan rekapitulasi nilai secara otomatis.

Oleh karena pertimbangan **diatas**, diperlukan sebuah sistem penilaian otomatis yang menggabungkan konsep **online judge** dengan fitur khusus untuk mendukung proses pembelajaran akademik. Dengan sistem ini, Asisten dapat memberikan tugas dan mahasiswa dapat mengunggah kode jawaban mereka langsung ke sistem, kemudian akan dinilai oleh penilai otomatis (**automated judger**). Kode tersebut akan dikompilasi dan diuji dengan **test case** yang telah disiapkan, lalu sistem akan memberikan hasil dan **feedback** jika terjadi kesalahan, seperti "*Runtime Error*" atau "Gagal pada **test case 3**". Dengan adanya sistem ini, proses penilaian menjadi lebih cepat dan objektif, mahasiswa juga mendapatkan umpan balik terkait kualitas kode **programnya**, sehingga mereka dapat meningkatkan pemahaman dan keterampilan pemrogramannya dengan lebih efektif.

Dengan sistem ini, asisten praktikum akan sangat terbantu dan dapat menghemat waktu banyak waktu. Asisten cukup fokus menyiapkan materi dan beberapa soal, lalu sistem akan secara otomatis menerima dan menilai jawaban mahasiswa secara langsung. Dengan demikian, asisten dapat lebih fokus pada hal-hal lain yang lebih penting tanpa harus menghabiskan banyak waktu untuk melakukan koreksi manual.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam membantu peningkatan waktu penilaian tugas praktikum di lingkungan Prodi Informatika FMIPA USK.

1.2 RUMUSAN MASALAH

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Diperlukan perancangan dan pengembangan sistem penilaian otomatis yang dapat menilai tugas pemrograman mahasiswa secara efisien dan akurat.
2. Identifikasi fitur yang diperlukan dalam sistem penilaian otomatis agar sesuai dengan kebutuhan praktikum pemrograman di lingkungan akademik.
3. Diperlukan pengukuran kepuasan pengguna dan kemampuan sistem dalam membantu mempercepat proses penilaian serta memberikan umpan balik secara instan kepada mahasiswa.

1.3 TUJUAN PENELITIAN

Berdasarkan rumusan masalah yang telah diuraikan, tujuan penelitian ini adalah:

1. Mengembangkan sistem penilaian otomatis berbasis online judge yang dapat menilai tugas pemrograman mahasiswa secara otomatis dan objektif.
2. Menyediakan fitur-fitur tambahan seperti manajemen kelas, pencatatan mahasiswa, serta rekapitulasi nilai untuk mendukung kegiatan praktikum pemrograman.
3. Melakukan pengukuran terhadap kontribusi sistem dalam mempercepat penilaian dan penyediaan umpan balik instan kepada mahasiswa, serta mengevaluasi tingkat kepuasan pengguna melalui pengujian usability.

1.4 MANFAAT PENELITIAN

Adapun manfaat penelitian diharapkan dari penelitian ini antara lain:

1. Sistem penilaian otomatis ini dapat mengurangi beban kerja asisten, sehingga asisten tidak perlu lagi mengunduh, menjalankan, dan memeriksa kode mahasiswa secara manual.
2. Mahasiswa dapat langsung mengetahui kesalahan dalam kode mereka melalui laporan hasil evaluasi otomatis, sehingga dapat segera melakukan perbaikan dan meningkatkan pemahaman mereka terhadap konsep pemrograman.
3. Dengan sistem yang terintegrasi, pencatatan nilai otomatis untuk setiap tugas, memudahkan asisten untuk pencatatan nilai tugas.

BAB II

TINJAUAN KEPUSTAKAAN

2.1 PRAKTIKUM

Praktikum merupakan bagian penting dalam proses pembelajaran di bidang ilmu komputer dan teknik. Pembelajaran berbasis praktikum memungkinkan mahasiswa untuk mempraktikkan langsung teori yang telah dipelajari di kelas, sehingga dapat meningkatkan pemahaman dan keterampilan terkait teori tersebut. Dalam informatika, praktikum menjadi komponen penting untuk meningkatkan kemampuan *problem solving*.

Di Program Studi Informatika Universitas Syiah Kuala, praktikum menjadi fondasi utama dalam pembelajaran berbasis pengalaman (*experiential learning*). Melalui kegiatan praktikum, mahasiswa dapat membangun keterampilan praktis secara langsung, bereksperimen, serta melatih kemampuan *debugging*. Pendekatan ini terbukti lebih efektif dibandingkan dengan pembelajaran yang hanya berfokus pada teori semata.

Penelitian oleh (Abdiansah et al., 2023) menunjukkan bahwa pendekatan pembelajaran pemrograman berbasis praktik mampu meningkatkan motivasi belajar dan pemahaman siswa terhadap materi yang bersifat STEM (*Science, Technology, Engineering, Mathematics*). Hasil pelatihan menunjukkan peningkatan kemampuan siswa sebesar 15% (*N-Gain*) dan tingkat kepuasan peserta mencapai 84,91%. Temuan ini memperkuat bahwa pendekatan praktikum secara umum lebih efektif dalam mendukung pembelajaran pemrograman, dan sangat relevan untuk diterapkan di lingkungan akademik seperti Program Studi Informatika Universitas Syiah Kuala.

2.2 PEMROGRAMAN

Pemrograman merupakan proses menulis kode instruksi yang dapat dipahami dan dijalankan oleh komputer untuk menyelesaikan tugas tertentu. Dalam praktiknya, pemrograman tidak hanya terbatas pada penulisan kode, tetapi juga mencakup kemampuan berpikir kritis, logis, dan sistematis untuk menciptakan solusi digital.

Bahasa pemrograman seperti Python, Java, dan C++ berfungsi sebagai bahasa yang dapat menerjemahkan logika manusia ke dalam bentuk instruksi yang dapat dimengerti oleh mesin. Selain itu, terdapat berbagai paradigma dalam pemrograman seperti pemrograman berorientasi objek (*OOP*), fungsional, dan prosedural yang memberikan kerangka kerja berbeda dalam menyusun struktur program. (Kumar et al., 2023)

2.3 STRUKTUR DATA DAN ALGORITMA

Struktur data dan algoritma merupakan materi penting dalam ilmu komputer yang menentukan efisiensi program. Pemahaman tentang struktur data seperti *array*,

linked list, *stack*, *queue*, dan *tree* sangat penting karena struktur-struktur ini digunakan dalam berbagai aplikasi pemrograman.

Efisiensi suatu program tidak hanya ditentukan oleh pilihan bahasa pemrograman, tetapi juga oleh bagaimana struktur data dan algoritmanya diterapkan dalam menyelesaikan permasalahan. Contohnya, pemrosesan data berskala besar, pemilihan antara struktur data *hash table* dan *binary search tree* dapat sangat mempengaruhi kecepatan pencarian data. Oleh karena itu, pemahaman tentang kompleksitas algoritma menjadi sangat penting dan menjadi kunci yang menentukan kesuksesan mahasiswa IT (Efendi, 2022).

2.4 ONLINE JUDGE

Online Judge adalah sistem otomatis yang digunakan untuk mengevaluasi solusi pemrograman berdasarkan serangkaian kasus uji (*test testcase*). Sistem ini biasanya digunakan di dalam kompetisi pemrograman. *Online Judge* memungkinkan pengujian kode dalam berbagai bahasa pemrograman tanpa perlu konfigurasi lingkungan secara manual.

Menurut (Patel et al., 2021), sistem *Online Judge* memainkan peran penting dalam meningkatkan keterampilan pemecahan masalah mahasiswa. Dengan sistem ini, peserta dapat menguji kode mereka secara langsung dan mendapatkan umpan balik langsung mengenai kesalahan sintaks maupun efisiensi algoritma yang digunakan. Penggunaan *Online Judge* juga membantu dalam pengembangan pemikiran komputasional karena peserta harus memahami dan mengoptimalkan solusi berdasarkan batasan yang diberikan.

Selain itu, penelitian oleh (Xu et al., 2022) menunjukkan bahwa *Online Judge* yang menerapkan sistem peringkat berdasarkan performa pengguna dapat mendorong kompetisi sehat dan meningkatkan motivasi peserta untuk terus mengembangkan keterampilan pemrograman mereka. Beberapa platform terkenal yang menggunakan sistem ini adalah Codeforces, LeetCode, dan HackerRank.

2.5 NEXT.JS

Next.JS adalah sebuah *framework* berbasis React yang dikembangkan oleh Vercel untuk membangun aplikasi web dengan fitur *server-side rendering* (SSR), *static site generation* (SSG), dan *incremental static regeneration* (ISR). Dengan pendekatan ini, Next.JS mampu meningkatkan kecepatan *load* halaman dan performa aplikasinya, terutama dalam konteks optimasi (SEO) dan pengalaman pengguna yang lebih baik.

Menurut (Lee et al., 2022), penggunaan Next.JS dalam pengembangan web modern memberikan berbagai keuntungan, terutama dalam hal performa dan fleksibilitas. Fitur seperti *prefetching* otomatis dan optimalisasi gambar membuat situs web dengan Next.JS lebih responsif dan efisien. Dalam studi tersebut, disebutkan

bahwa dibandingkan dengan aplikasi React tradisional yang menggunakan **client-side rendering (CSR)**, Next.JS mampu mengurangi waktu muat halaman hingga 40%.

2.6 TAILWIND CSS

Tailwind CSS adalah **framework utility-first** yang memberikan kemudahan bagi web *developer* dalam membangun antarmuka dengan cepat dan fleksibel. Dibandingkan dengan **framework** CSS lain seperti Bootstrap yang menawarkan komponen siap pakai, **Tailwind** menyediakan **class** utilitas yang langsung diterapkan ke elemen HTML, memungkinkan pengembang melakukan desain tanpa harus menulis CSS dari nol. Tailwind CSS mempercepat proses pembuatan tata letak **website** karena strukturnya yang fleksibel, sehingga memudahkan penyesuaian gaya antarmuka. Pendekatan **utility-first** pada Tailwind mendukung pengembangan antarmuka yang lebih konsisten dan responsif, terutama dalam proyek berbasis **website** yang membutuhkan kecepatan dan **skalabilitas**. Dengan kemampuannya mengurangi kode CSS berlebihan dan memberikan kebebasan desain, Tailwind CSS menjadi pilihan utama bagi banyak pengembang modern yang mengutamakan efisiensi dan fleksibilitas. (Yudhistira, 2023)

2.7 APPLICATION PROGRAMMING INTERFACE (API)

Application Programming Interface (API) adalah seperangkat aturan dan alat yang memungkinkan komunikasi antara berbagai perangkat lunak. API digunakan secara luas dalam pengembangan perangkat lunak modern untuk memungkinkan integrasi dengan layanan lain tanpa harus mengakses atau mengetahui seperti apa implementasi internal sistem yang terlibat.

Berdasarkan strukturnya, API digolongkan ke dalam beberapa jenis, diantaranya *REST*, *GraphQL*, *SOAP*, dan *WebSocket*. REST API menjadi salah satu standar yang paling banyak digunakan karena fleksibilitas dan kesederhanaannya dalam komunikasi berbasis HTTP. API juga memungkinkan pengembang untuk membangun layanan berbasis **microservices**, yang memungkinkan setiap komponen perangkat lunak bekerja secara independen dan lebih mudah dikembangkan.

Studi oleh (Kim et al., 2021) menunjukkan bahwa pemanfaatan API secara efektif dapat meningkatkan efisiensi pengembangan perangkat lunak, karena tim pengembang tidak perlu membangun fitur dari nol. Sebagai contoh, API Google Maps memungkinkan aplikasi untuk menampilkan peta dan navigasi secara langsung tanpa harus mengembangkan sistem pemetaan sendiri. Dengan berkembangnya teknologi **cloud** dan layanan berbasis SaaS (**Software as a Service**), API semakin menjadi komponen utama dalam pengembangan aplikasi modern.

2.8 REST (REPRESENTATIONAL STATE TRANSFER)

Representational State Transfer (REST) adalah gaya arsitektur perangkat lunak yang digunakan dalam desain API berbasis HTTP. REST diperkenalkan oleh (Fielding, 2000) dalam disertasinya dan telah menjadi standar utama dalam pengembangan layanan web modern. REST menggunakan metode HTTP standar seperti *GET*, *POST*, *PUT*, *DELETE* untuk berkomunikasi dengan server, menjadikannya pilihan utama dalam pengembangan aplikasi berbasis web dan *mobile*. Perusahaan-perusahaan besar seperti Netflix, Twitter dan Uber menggunakan REST untuk membangun *backend* sistem mereka (Golmohammadi et al., 2022).

Salah satu prinsip utama REST adalah *statelessness*, yang berarti setiap permintaan yang dikirimkan oleh klien harus berisi semua informasi yang diperlukan tanpa bergantung pada sesi sebelumnya. Hal ini membuat REST API menjadi lebih mudah diskalakan dan lebih tahan terhadap kegagalan sistem dibandingkan dengan arsitektur lain yang memerlukan penyimpanan status pengguna di server.

2.9 FASTAPI

FastAPI adalah *framework backend* berbasis Python yang dirancang untuk membangun API dengan performa tinggi menggunakan pendekatan *asynchronous programming*. Framework ini dikembangkan oleh Sebastián Ramírez dan telah menjadi salah satu pilihan utama dalam pengembangan layanan berbasis web karena kecepatan dan kemudahannya.

Menurut (Ramirez, 2021), FastAPI menawarkan performa yang setara dengan *framework* berbasis Node.js seperti Express, tetapi dengan keunggulan dalam validasi data berkat dukungan penuh terhadap Python *type hints* dan pustaka *pydantic*. Studi tersebut menunjukkan bahwa API yang dibangun dengan FastAPI memiliki waktu respons yang lebih cepat dibandingkan dengan Flask dan Django REST Framework (DRF), menjadikannya pilihan ideal untuk layanan berskala besar.

Keunggulan lain dari FastAPI adalah kemampuannya dalam menghasilkan dokumentasi API secara otomatis menggunakan OpenAPI dan Swagger UI. Hal ini mempermudah pengembang dalam memahami dan menguji *endpoint* yang telah dibuat tanpa memerlukan konfigurasi tambahan. Oleh karena itu, FastAPI semakin banyak digunakan dalam berbagai industri, termasuk layanan AI, *fintech*, dan aplikasi berbasis *cloud*.

2.10 MYSQL

MySQL merupakan sistem manajemen basis data relasional (*Relational Database Management System*) bersifat *open-source* yang menggunakan bahasa SQL (*Structured Query Language*). MySQL dirancang untuk memungkinkan pengguna menyimpan, mengakses, dan memanipulasi data di sistemnya. Pada awalnya,

MySQL dikembangkan oleh perusahaan Swedia MySQL AB, kini MySQL berada di bawah pengelolaan Oracle Corporation setelah melalui beberapa akuisisi perusahaan sebelumnya (bin Uzayr, 2022).

MySQL dikenal luas karena kemampuannya yang andal, fleksibel, serta didukung di berbagai sistem operasi seperti Linux, Windows, dan macOS. MySQL juga terintegrasi dengan baik dengan berbagai bahasa pemrograman seperti PHP, Python, Java, Javascript, dan C/C++. Hal ini menjadikannya pilihan utama dalam pengembangan aplikasi web, termasuk dalam arsitektur XAMPP (X (*Cross-Platform*), Apache, MySQL, PHP, Perl).

MySQL hadir dalam dua versi utama: **Community Edition**, yang bebas digunakan di bawah lisensi GPL, dan **Enterprise Edition**, yang menawarkan dukungan tambahan dari Oracle. Dengan berbagai kelebihan tersebut, MySQL telah digunakan secara luas oleh organisasi kecil maupun besar dalam pengelolaan data berskala tinggi.

2.11 ENTITY RELATIONAL DIAGRAM (ERD)

Entity Relationship Diagram (ERD) adalah alat bantu visual yang digunakan dalam tahap awal perancangan basis data. ERD menggambarkan struktur data secara konseptual, termasuk entitas, atribut, dan relasi antar entitas, sehingga memudahkan perancang memahami alur dan keterkaitan data dalam sistem. Komponen ERD:

- **Entitas**

Entitas merupakan objek nyata atau konsep yang datanya ingin disimpan, seperti Mahasiswa, Dosen, atau Mata Kuliah. Disimbolkan dengan persegi panjang.

- **Atribut**

Atribut adalah karakteristik dari entitas, seperti Nama, NIM, atau Alamat. Setiap entitas wajib memiliki atribut **primary key** sebagai identitas yang unik. Disimbolkan dengan elips.

- **Relasi**

Relasi menunjukkan hubungan antar entitas, misalnya mahasiswa mengambil Mata Kuliah. Relasi disimbolkan dengan belah ketupat dan dapat memiliki derajat **kardinalitas** sebagai berikut:

1. **One to One**

Satu entitas hanya berelasi dengan satu entitas lain.

2. **One to Many**

Satu entitas hanya berelasi dengan banyak entitas lain.

3. **Many to Many**

Banyak entitas hanya berelasi dengan banyak entitas lain.

Dengan ERD yang baik, proses implementasi database menjadi lebih terstruktur dan minim kesalahan (Afifah et al., 2022).

2.12 RAPID APPLICATION DEVELOPMENT

Rapid Application Development (RAD) adalah salah satu metode dalam pengembangan aplikasi. Metode ini menekankan kecepatan dalam iterasi dan pengembangannya. Sehingga pengerjaannya dapat dilakukan dalam iterasi yang lebih cepat (Aryanti et al., 2021). Metode ini sering digunakan karena dapat melakukan pengembangan secara cepat sehingga cocok digunakan untuk pengembangan perangkat lunak skala kecil (Sondang, 2024). Dalam pengembangannya RAD terdiri dari 4 tahap, perencanaan kebutuhan, desain sistem, pengembangan, implemementasi. Setiap siklus dalam metode ini dilakukan dengan iterasi yang cepat. (Sondang, 2024).

2.13 PENGUJIAN

Setiap sistem perlu diadakan pengujian, hal ini berguna untuk memastikan bahwa semua komponen dapat berfungsi dan berjalan sesuai dengan spesifikasi yang diharapkan. Tujuan utama pengujian ini adalah menemukan bug, memastikan kinerja sistem optimal, dan menjamin bahwa sistem siap digunakan oleh pengguna akhir.

2.13.1 Load Testing

Load Testing merupakan salah satu metode dalam pengujian performa (*performance testing*) yang bertujuan untuk mengetahui bagaimana sistem bekerja di bawah beban pengguna yang sesuai dengan kondisi nyata. Tujuan utama dari *load testing* adalah mengukur kecepatan respon, kapasitas maksimum pengguna, serta stabilitas sistem saat digunakan secara bersamaan

Load testing dilakukan dengan cara mensimulasikan sejumlah pengguna (virtual users) yang mengakses sistem dalam waktu yang bersamaan, untuk melihat apakah sistem mampu menangani beban tersebut tanpa menurunnya performa atau terjadinya kesalahan layanan. Pengujian ini sangat penting dilakukan pada aplikasi yang memiliki potensi trafik tinggi, seperti sistem layanan berbasis API, e-commerce, atau sistem keuangan (Hendayun et al., 2023).

Menurut (Erinle, 2017), *load testing* dapat membantu:

- Mengidentifikasi bottleneck performa sejak dini.
- Menjamin aplikasi dapat berjalan sesuai ekspektasi saat digunakan oleh banyak pengguna.
- Memberikan dasar bagi optimasi konfigurasi sistem dan infrastruktur pendukung.

2.13.2 System Usability Scale (SUS)

System Usability Scale adalah metode evaluasi yang digunakan untuk mengukur seberapa berguna suatu sistem, aplikasi, atau produk. SUS dikembangkan oleh John Brooke pada tahun 1986 dan masih banyak digunakan karena sederhana, cepat, dan efektif dalam memberikan gambaran umum tentang pengalaman pengguna (Nuriman & Mayesti, 2020).

SUS terdiri dari 10 pernyataan yang dijawab oleh pengguna dengan skala 1 sampai 5 (1 = sangat tidak setuju, 5 = sangat setuju). Pertanyaannya mencakup aspek seperti kemudahan penggunaan, fungsi aplikasi dan kenyamanan pengalaman seperti pada 2.1

Tabel 2.1: Daftar Pertanyaan *System Usability Scale*

item	Pertanyaan	Skor
Q1	Saya berpikir akan menggunakan aplikasi ini lagi	1-5
Q2	Saya menemukan bahwa aplikasi ini tidak dibuat serumit ini	1-5
Q3	Saya berpikir aplikasi ini mudah untuk digunakan	1-5
Q4	Saya berpikir saya perlu bantuan orang teknis dalam menggunakan aplikasi ini	1-5
Q5	Saya menemukan berbagai fungsi di aplikasi ini terintegrasi dengan baik	1-5
Q6	Saya pikir terlalu banyak ketidakkonsistenan dalam aplikasi ini	1-5
Q7	Saya berpikir bahwa kebanyakan orang akan belajar dengan mudah dalam mempelajari aplikasi ini	1-5
Q8	Saya menemukan aplikasi ini sangat tidak praktis	1-5
Q9	Saya merasa sangat percaya diri dalam menggunakan aplikasi ini	1-5
Q10	Saya perlu banyak belajar sebelum menggunakan aplikasi ini	1-5

2.14 PENELITIAN TERKAIT

Beberapa penelitian dan sistem terkait telah dikembangkan sebelumnya yang menjadi dasar dalam pembuatan platform ini:

1. Perancangan Sistem Penilaian Otomatis Program Java Berbasis Web pada Perkuliahan Pemrograman

Penelitian ini menekankan pentingnya tugas pemrograman dalam membantu siswa memahami elemen bahasa pemrograman yang sedang dipelajari. Sistem yang dirancang memungkinkan siswa mengunggah jawaban tugas pemrograman ke aplikasi web, yang kemudian secara otomatis menilai program tersebut melalui proses kompilasi dan pengujian kasus uji yang telah ditentukan (Sukendar, 2023).

2. Penerapan Sistem Evaluasi Pembelajaran Pemrograman Terintegrasi dengan Online Judge di SMK

Penelitian ini membahas pengembangan sistem evaluasi pembelajaran pemrograman berbasis Online Judge di Sekolah Menengah Kejuruan (SMK). Penelitian ini **dilatarbelakangi** oleh proses evaluasi pembelajaran pemrograman yang masih bersifat manual, di mana guru harus memeriksa jawaban kode program siswa satu per satu, yang cenderung tidak efisien dan rawan kesalahan. Sistem Online Judge yang diterapkan mampu mengotomatiskan proses penilaian dengan cara mengompilasi dan mengevaluasi kode program siswa, termasuk mendeteksi **error** kompilasi, waktu eksekusi, serta kesalahan **sintaks**. (Mustamiin et al., 2022)

3. A Designed Platform for Programming Courses

Penelitian ini membahas pengembangan platform pembelajaran pemrograman yang dirancang khusus untuk mendukung pengajaran di kelas. Platform ini bertujuan mengurangi beban pengajaran dan meningkatkan efisiensi belajar pemrograman. Dibandingkan dengan sistem **"online judge"** yang banyak tersedia di internet yang biasanya hanya memeriksa kebenaran kode platform ini menawarkan fitur lebih lengkap yang mendukung pengelolaan kelas, manajemen siswa, serta pengaturan soal pemrograman. Sistem ini juga mampu memberikan umpan balik terkait format kode yang dikumpulkan agar siswa mengembangkan kebiasaan pemrograman yang baik.

Selain fitur-fitur tersebut, platform ini dilengkapi dengan fungsi siaran perangkat lunak untuk membantu guru dalam mendemonstrasikan materi secara langsung, serta menyediakan data statistik guna mengevaluasi performa belajar siswa di kelas. Hasil pengujian selama satu semester menunjukkan bahwa platform ini sangat efektif sebagai alat bantu pengajaran. Ini membuktikan bahwa pendekatan yang dirancang sesuai kebutuhan pembelajaran di kelas dapat meningkatkan keterlibatan siswa dan membantu guru mengelola pembelajaran dengan lebih efisien (Liang et al., 2020).

Daftar penelitian terkait **dikatas**, dapat dilihat juga pada tabel 2.2

Tabel 2.2: Ringkasan Literatur Terkait Sistem Online Judge

No	Judul	Tahun	Peneliti	Tujuan
1	Perancangan Sistem Penilaian Otomatis Program Java Berbasis Web pada Perkuliahan Pemrograman	2023	Ade Sukendar	Merancang sistem penilaian otomatis kode program Java mata kuliah pemrograman jurusan Teknik Informatika, Univeristas Pasundan
2	Penerapan Sistem Evaluasi Pembelajaran Pemrograman Terintegrasi dengan Online Judge di SMK	2022	Muhamad Mustamiin, Adi Suheryadi, Alifia Puspaningrum	Merancang sistem sistem evaluasi pembelajaran pemrograman terintegrasi dengan <i>online judge</i> pada mata pelajaran pemrograman dasar di SMKN 2 Indramayu
3	A Designed Platform for Programming Courses	2020	Jia-Chi Liang, Jia-Yin Wang, Shen-Hung Chou	Merancang platform kursus pemrograman dengan mengintegrasikan sistem <i>online judge</i> yang support C/C++ dan Python

BAB III METODOLOGI PENELITIAN

3.1 WAKTU DAN LOKASI PENELITIAN

Penelitian ini dilaksanakan selama lima bulan, mulai dari April 2025 hingga Agustus 2025, bertempat di Laboratorium Rekayasa Perangkat Lunak (RPL), Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Universitas Syiah Kuala.

Tabel 3.1: Jadwal Pelaksanaan Penelitian

No	Kegiatan	Bulan (Minggu Ke-)																			
		April				Mei				Juni				Juli				Agustus			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Penyusunan Proposal																				
2	Identifikasi Masalah																				
3	Studi Literatur																				
4	Analisis Kebutuhan																				
5	Perancangan Sistem																				
6	Pengembangan Sistem																				
7	Pengujian dan Evaluasi																				

3.2 ALAT DAN BAHAN

3.2.1 Perangkat Keras (Hardware)

Perangkat keras yang digunakan dalam penelitian ini adalah satu unit laptop Lenovo V14 G4 dengan spesifikasi sebagai berikut:

- Prosesor: AMD Ryzen 3 7320U (4 core / 8 thread, 2,4 GHz hingga 4,1 GHz, 2 MB L2 / 4 MB L3)
- Layar: 14 inci FHD (1920 × 1080) TN 250 nits
- Memori: 8 GB LPDDR5-5500
- Penyimpanan: SSD 256 GB M.2 2242 PCIe 4.0x4 NVMe
- Kartu Grafis: AMD Radeon 610M

3.2.2 Perangkat Lunak (Software)

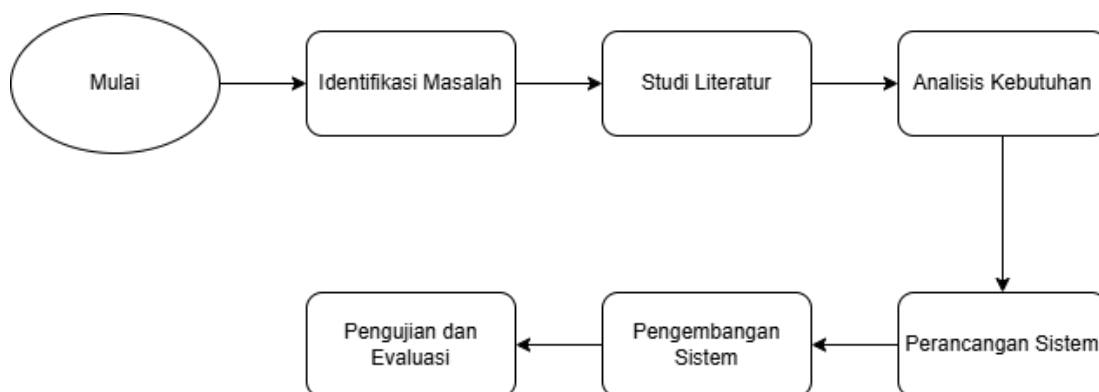
Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Google Chrome versi 135.0.7049.116
2. Visual Studio Code versi 1.92.8

3. Node JS versi 20.11
4. Typescript versi 5.8.2
5. Next JS versi 15
6. Python versi 3.12.6
7. Figma
8. Draw.io
9. Miro
10. Postman versi 11.23.3

3.3 METODE PENELITIAN

Metode penelitian yang dilakukan pada penelitian terdiri dari beberapa tahapan berikut, sebagaimana pada Gambar 3.1



Gambar 3.1: Diagram Alur Penelitian

3.3.1 Identifikasi Masalah

Identifikasi masalah dalam proses penilaian tugas pada praktikum Pemrograman dan Struktur Data dan Algoritma (SDA) dilakukan untuk memahami berbagai tantangan yang dihadapi dalam pelaksanaan pembelajaran berbasis latihan pemrograman. Kedua praktikum ini dipilih karena memiliki karakteristik tugas yang sangat bergantung pada logika algoritmik dan ketepatan implementasi kode. Analisis dilakukan terhadap alur kerja asisten laboratorium dalam menyusun soal, mendistribusikan tugas, serta melakukan proses penilaian terhadap kode yang dikumpulkan oleh mahasiswa.

Secara umum, asisten laboratorium memiliki tanggung jawab untuk membuat soal, menentukan platform distribusi, memastikan mahasiswa dapat mengunggah

tugas dengan benar, menjalankan kode mahasiswa menggunakan sejumlah *test case*, dan memberikan penilaian yang sesuai dengan kriteria. Selain itu, identifikasi juga mencakup pengalaman mahasiswa dalam menerima tugas, memahami instruksi, menyelesaikan permasalahan pemrograman, hingga mengunggah hasil tugas untuk dinilai.

Untuk memperkuat pemahaman terhadap permasalahan tersebut, peneliti melakukan wawancara *semi-terstruktur* dengan sembilan asisten laboratorium dari praktikum Pemrograman dan SDA yang aktif pada tahun 2023 hingga 2025. Hasil wawancara menunjukkan bahwa sebagian besar asisten mengalami kesulitan dalam menjaga efisiensi waktu penilaian, konsistensi dalam pemberian nilai, serta menghadapi beban kerja yang cukup tinggi, khususnya saat menangani kelas dengan jumlah mahasiswa besar.

Secara umum, proses pembuatan dan distribusi tugas dilakukan melalui platform Google Classroom. Mahasiswa biasanya mengunggah kode dalam format *.c*, *.java*, atau *.zip*, yang kemudian diunduh secara manual oleh asisten untuk diuji secara lokal. Proses pengujian dilakukan dengan cara menjalankan kode dan memasukkan input secara manual, lalu mencocokkan *output* dengan hasil yang diharapkan. Durasi penilaian bervariasi antara 1 hingga 5 menit per mahasiswa, tergantung kompleksitas tugas. Sebagian asisten menggunakan *grader* sederhana untuk mempercepat proses, namun sebagian besar penilaian tetap dilakukan secara manual. Nilai akhir biasanya direkap secara manual melalui *spreadsheet* atau langsung dimasukkan ke Google Classroom.

Berdasarkan temuan tersebut, terdapat tiga permasalahan utama yang secara konsisten muncul dalam pelaksanaan penilaian praktikum:

1. Proses Penilaian Manual yang Memakan Waktu

Asisten harus memeriksa setiap tugas mahasiswa secara manual dengan mengunduh, menjalankan, dan menguji kode menggunakan beberapa *test case*. Proses ini sangat memakan waktu, terutama jika jumlah mahasiswa banyak dan tugas bersifat kompleks.

2. Tingginya Beban Kerja Asisten Praktikum

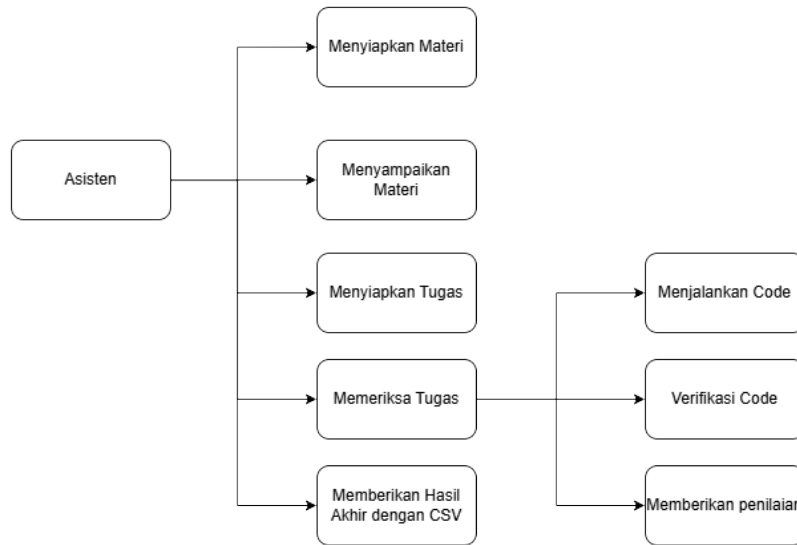
Banyaknya tugas yang harus diperiksa dalam waktu terbatas menyebabkan beban kerja tinggi bagi para asisten. Hal ini berisiko menimbulkan keterlambatan dalam pemberian nilai maupun umpan balik, serta menurunkan akurasi dan konsistensi penilaian.

3. Kurangnya Transparansi dan Umpan Balik kepada Mahasiswa

Mahasiswa seringkali tidak mengetahui secara rinci bagaimana penilaian

dilakukan atau letak kesalahan dalam kode mereka. Minimnya umpan balik menghambat proses belajar dan refleksi diri terhadap kesalahan pemrograman yang terjadi.

Permasalahan-permasalahan tersebut menunjukkan perlunya sistem yang dapat mengotomatisasi proses penilaian tugas pemrograman, meningkatkan efisiensi kerja asisten, serta menyediakan umpan balik yang cepat dan transparan kepada mahasiswa.



Gambar 3.2: Task Asisten Tanpa Bantuan Sistem Penilaian

3.3.2 Studi Literatur

Tahap studi literatur dilakukan untuk mencari dan mengkaji referensi yang relevan guna mencapai tujuan penelitian. Studi ini mencakup pencarian sumber dari berbagai media seperti situs web, dokumentasi resmi, buku, serta jurnal nasional dan internasional yang membahas sistem penilaian otomatis dalam pemrograman. Fokus utama dalam studi literatur adalah memahami konsep dan implementasi sistem **online judge** yang dapat digunakan untuk menilai tugas pemrograman secara otomatis.

Dalam penelitian ini, peneliti mencari referensi yang sesuai untuk membangun sistem **judge** yang optimal dengan menggunakan *Google Scholar* serta jurnal ilmiah terkait. Beberapa sistem yang menjadi acuan dalam penelitian ini adalah DOMJudge, LeetCode, dan TLX TOKI. Sistem-sistem ini telah dikenal digunakan secara luas dalam kompetisi pemrograman dan pendidikan. Studi terhadap sistem tersebut akan membantu dalam memahami mekanisme penilaian, eksekusi kode, serta metode pengujian otomatis.

3.3.3 Analisis Kebutuhan

Pada tahap ini, peneliti melakukan identifikasi terhadap berbagai aspek yang diperlukan dalam pengembangan sistem **online judge** untuk penilaian tugas pemrograman. Antara lain adalah:

- **Identifikasi Kebutuhan Pengguna**

Peneliti mengidentifikasi peran dan kebutuhan pengguna dalam sistem, antara lain :

1. **Laboran**

Laboran berperan sebagai administrator sistem yang mengelola data kelas, asisten dan praktikan. Tugas utama laboran antara lain:

- (a) Membuat dan mengelola data kelas.
- (b) Menetapkan dan mengatur peran asisten dalam setiap kelas
- (c) Menambahkan dan mengelola data praktikan kelas.

2. **Asisten**

Asisten berperan sebagai pengajar di dalam kelas, asisten dapat memberikan tugas dan soalnya di dalam kelas, lalu mendapatkan rekap hasil tugasnya. Tugas utama asisten antara lain:

- (a) Membuat dan mengelola tugas.
- (b) Membuat dan mengelola soal yang ada di dalam tugas.
- (c) Mengunduh laporan hasil tugas dalam format **csv**.

3. **Praktikan**

Praktikan adalah mahasiswa yang mengikuti suatu kelas. Praktikan dapat melihat tugas di dalam kelas dan menjawab soalnya. Tugas utama praktikan antara lain:

- (a) Melihat tugas dan soal kelas.
- (b) Mengupload kode jawaban.
- (c) Melihat hasil jawaban bila diperbolehkan oleh asisten.

4. **Dosen Pengampu**

Dosen pengampu adalah dosen yang bertanggung jawab terhadap mata kuliah dan praktikumnya. Dosen pengampu dapat mengetahui apa saja materi dan penilaian yang diberikan dalam praktikum. Beberapa hal yang dapat dilakukan dosen antara lain:

- (a) Melihat isi kelas

- (b) Melihat tugas yang diberikan
- (c) Melihat nilai mahasiswa

- **Identifikasi Kebutuhan Fungsional dan Non-Fungsional**

Peneliti akan menentukan fitur utama yang dibutuhkan dalam sistem. Identifikasi fitur ini terdapat dalam dua kategori utama, yaitu kebutuhan fungsional dan non-fungsional.

1. **Kebutuhan Fungsional**

Kebutuhan fungsional adalah fitur-fitur utama yang harus tersedia dalam sistem agar dapat digunakan sesuai dengan alur penilaian tugas pemrograman. Fitur-fitur tersebut antara lain:

- (a) **Unggahan Soal dan Test Case**

Asisten dapat mengunggah soal beserta *test case* dalam format yang telah ditentukan. *Test case* ini digunakan untuk menilai kebenaran program mahasiswa secara otomatis.

- (b) **Pengumpulan Kode Jawaban**

Mahasiswa dapat mengunggah kode program mereka untuk setiap soal yang tersedia dalam sistem.

- (c) **Penilaian Otomatis Berdasarkan Test Case**

Sistem menjalankan kode di server dan membandingkan hasil eksekusi dengan *output* yang telah ditentukan dalam *test case*, lalu menentukan status jawaban (Accepted, Wrong Answer, Runtime Error, dll.).

- (d) **Umpan Balik Langsung**

Setelah penilaian, mahasiswa menerima umpan balik secara langsung mengenai hasil jawabannya baik berupa status, waktu, dan memori yang digunakan.

- (e) **Manajemen Pengguna dan Kelas**

Laboran dapat membuat kelas, menetapkan asisten, serta mengelola daftar mahasiswa dan memasukkan mahasiswa ke dalam kelas.

2. **Kebutuhan Non-Fungsional** Kebutuhan *non-fungsional* adalah aspek-aspek kualitas sistem yang tidak berhubungan langsung dengan fitur, namun penting untuk mendukung kinerja, keandalan, dan kenyamanan pengguna. Kebutuhan *non-fungsional* pada penelitian ini antara lain :

- (a) **Keamanan Eksekusi Kode dalam Server**

Menjalankan kode yang diberikan oleh mahasiswa secara langsung di dalam server bisa sangat berbahaya. Ada kemungkinan mahasiswa

secara tidak sengaja atau sengaja menjalankan perintah-perintah berbahaya, seperti `rm -rf /` yang dapat menghapus seluruh isi direktori, atau mencoba mengakses data dan `file` yang tidak seharusnya. Oleh karena itu, diperlukan solusi untuk mengeksekusi kode dalam lingkungan yang aman dan terisolasi, yang disebut juga *sandbox*. Pada penelitian ini, eksekusi kode akan dilakukan menggunakan Docker sebagai teknologi *sandboxing*. Docker memungkinkan pembatasan sumber daya dan akses *file*, dari sistem utama.

(b) **Penjaminan Evaluasi Jawaban**

Sistem harus memastikan bahwa setiap jawaban yang diunggah oleh mahasiswa akan selalu diproses dan dinilai. Hal ini penting agar tidak ada mahasiswa yang dirugikan dan seluruh proses penilaian berjalan dengan benar. Untuk itu, sistem dirancang agar dapat menangani antrian tugas secara teratur sehingga tidak melewatkan setiap tugas.

3. Kebutuhan Format Soal

Agar proses penilaian dapat dilakukan secara otomatis oleh komponen *judge*, ada beberapa kriteria soal yang harus dipenuhi agar bisa dilakukan penilaian otomatis disebut. Soal-soal tersebut memiliki karakteristik sebagai berikut:

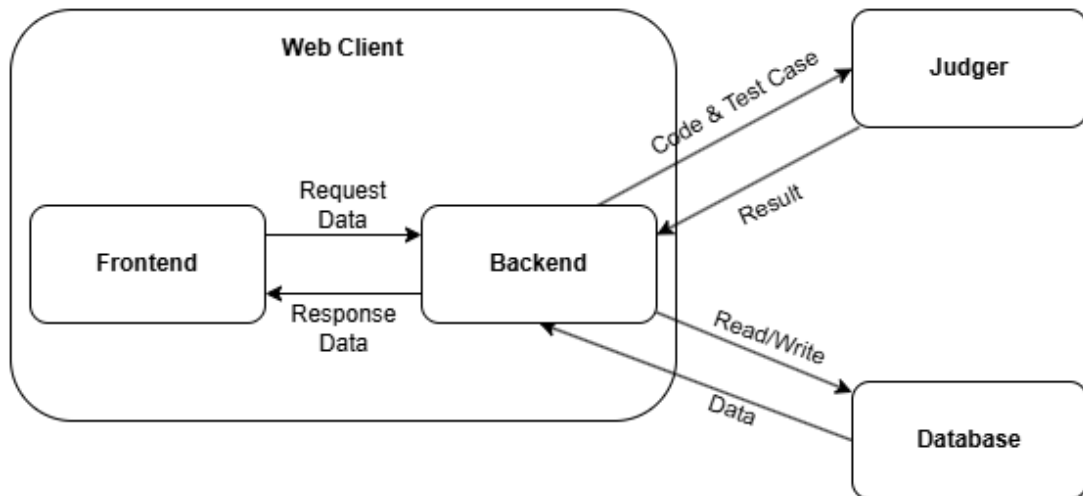
- (a) Masukan (input) dibaca langsung melalui *standard input* (`stdin`), tanpa antarmuka pengguna atau *file*.
- (b) Keluaran (*output*) ditulis ke *standard output* (`stdout`), tanpa ada tambahan kata atau kalimat selain jawaban yang diminta.
- (c) Penilaian dilakukan dengan membandingkan hasil *output* dari program dengan *expected output* yang tersedia dalam *test case*.
- (d) Soal dapat memiliki beberapa *test case* untuk menguji berbagai skenario.
- (e) Tidak boleh ada interaksi pengguna selama program berjalan, seperti penggunaan `input()` dengan `prompt` atau `print()` berisi penjelasan.

Dengan menggunakan format ini, sistem dapat menjalankan dan menilai program secara otomatis tanpa campur tangan manusia. Soal dengan format ini juga bisa disebut sebagai *Standar Input/Output Problem*.

3.3.4 Perancangan Sistem

Pada tahap perancangan sistem, peneliti merancang struktur dan arsitektur dari sistem *online judge* yang akan dikembangkan. Arsitektur platform secara keseluruhan

digambarkan **sebagaimana** pada Gambar 3.3.



Gambar 3.3: Arsitektur Platform Penilaian Tugas Pemrograman

Terdapat empat komponen utama dalam sistem ini, yaitu **frontend**, **backend**, **judger**, dan **database**. Komponen **database** termasuk ke dalam bagian **backend**. Berikut adalah gambaran umum dari masing-masing komponen:

- **Perancangan Judger**

Judger merupakan komponen yang bertanggung jawab untuk menjalankan dan menilai kode yang dikumpulkan oleh praktikan. Sistem ini menerima kode dan **test case** terkait lalu dapat mengeksekusi kode yang diberikan dan melakukan penilaian. Proses penilaian dilakukan dengan membandingkan **output** dari kode praktikan dengan **output** yang ditetapkan berdasarkan **test case** yang diberikan.

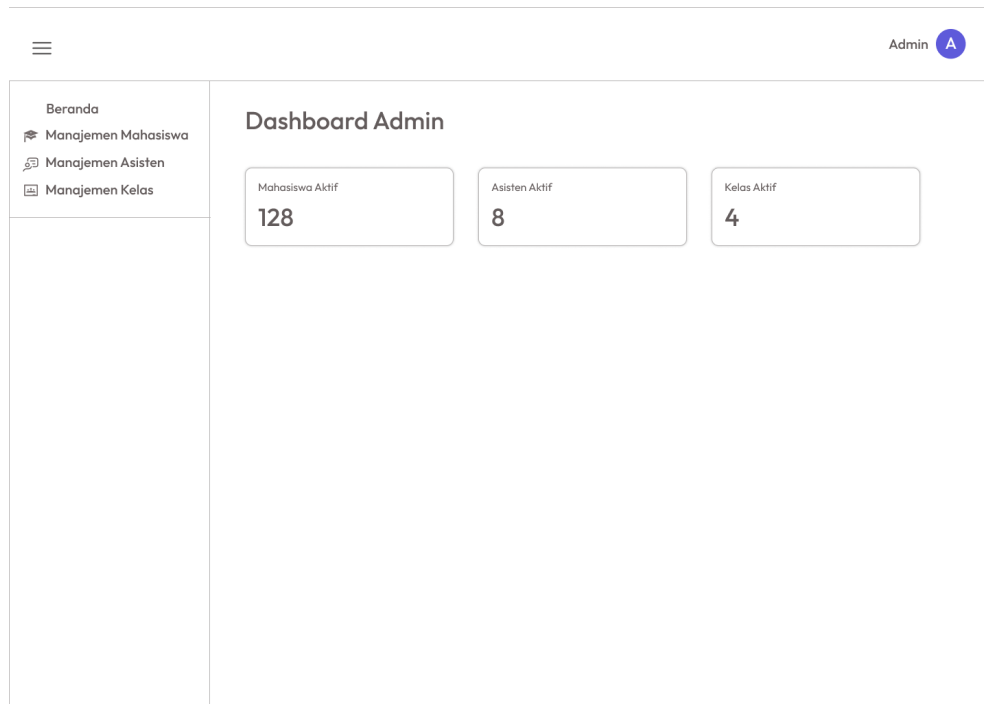
- **Perancangan Frontend**

Frontend merupakan antarmuka yang digunakan oleh pengguna seperti praktikan, asisten, dan laboran. Pada tahap ini dilakukan perancangan **prototype** yang menjelaskan bagaimana alur aplikasi nantinya. Pengguna dapat melakukan berbagai aktivitas seperti **login**, memilih soal, mengunggah kode jawaban, serta melihat hasil penilaian.

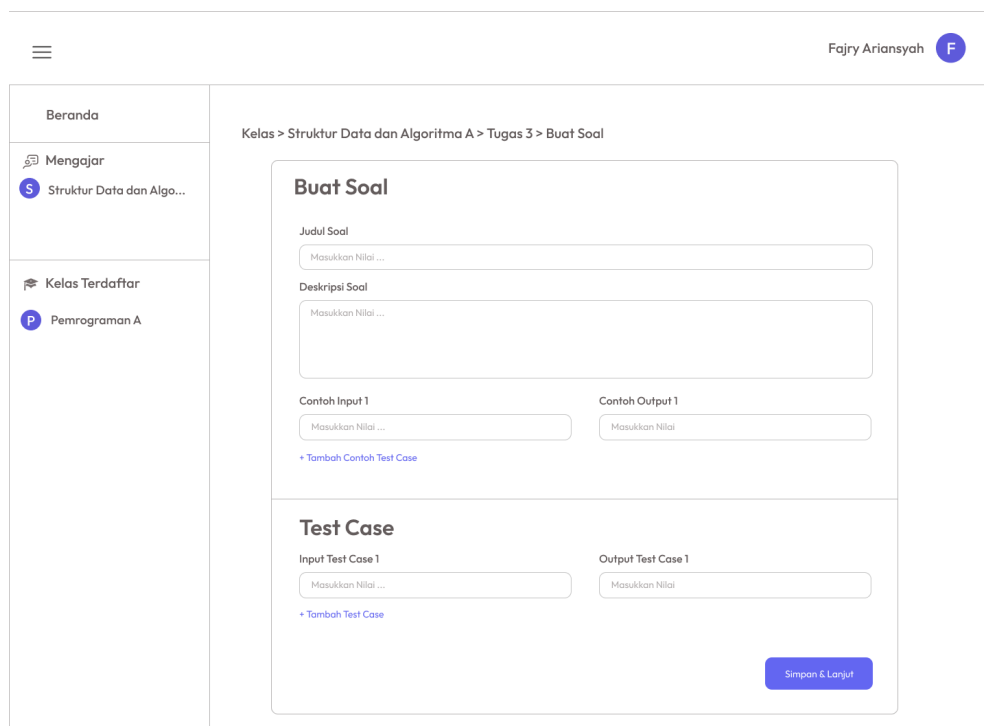
Untuk menggambarkan fitur utama dan alur interaksi pengguna, berikut adalah beberapa tampilan rancangan antarmuka pengguna sistem:

Gambar 3.4: Tampilan Halaman Login

Gambar 3.5: Dashboard Praktikan



Gambar 3.6: Dashboard Admin



Gambar 3.7: Form Pembuatan Soal oleh Asisten

Fajry Ariansyah

F

Beranda

Mengajar

S

Struktur Data dan Algo...

Kelas Terdaftar

P

Pemrograman A

Kelas > Pemrograman A > Tugas 3 > Soal 1

Soal 1 - Perkenalan if else

Deskripsi Soal

Seorang dosen ingin membuat program sederhana yang dapat mengkonversi nilai ujian mahasiswa (0-100) menjadi nilai huruf sesuai kriteria berikut:

- Jika nilai 91 - 100 maka A
- Jika nilai 81 - 90 maka B
- Jika nilai 71 - 80 maka C
- Jika nilai 61 - 70 maka D
- Jika 60 kebawah maka E

Bantulah dosen tersebut membuat program yang dapat mengkonversi nilai ujiannya. Jika rentang nilai yang diinput tidak sesuai, tampilkan "Nilai tidak valid".

Contoh Input 1

79

Contoh Output 1

C

Contoh Input 2

999

Contoh Output 2

Nilai tidak valid

Submit Jawaban

Source code

Browse

Language

C

Submit

Gambar 3.8: Daftar Soal

Fajry Ariansyah

F

Beranda

Mengajar

S

Struktur Data dan Algo...

Kelas Terdaftar

P

Pemrograman A

Kelas > Struktur Data dan Algoritma A > Buat Tugas

Buat Tugas

Judul Tugas

Masukkan Nilai ...

Deskripsi Tugas

Masukkan Nilai ...

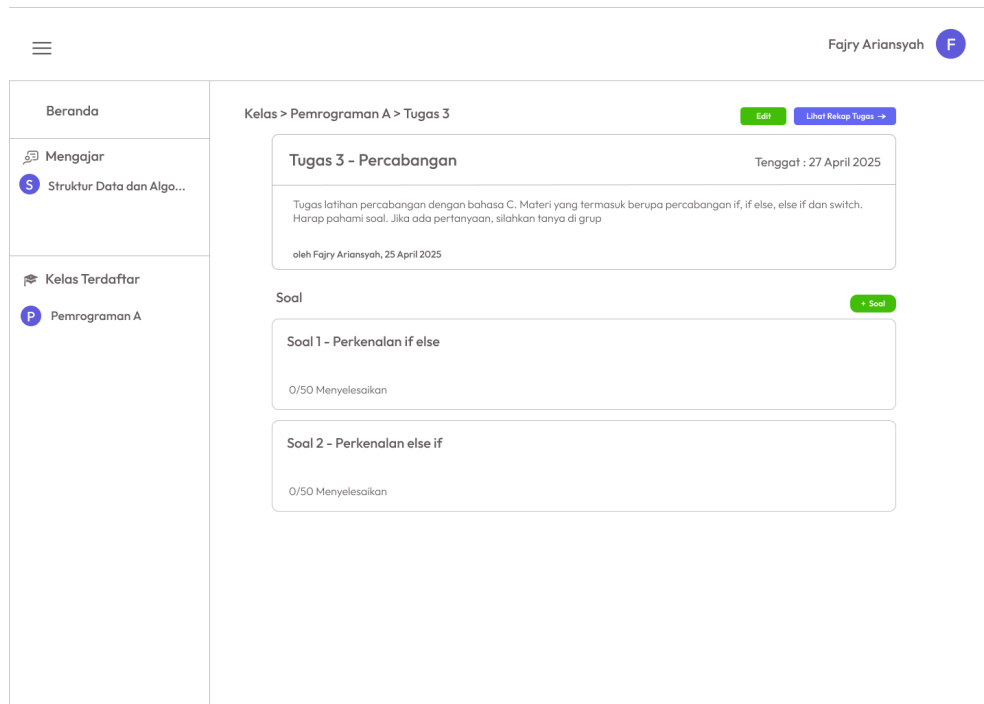
Batas Kumpul

mm/dd/yyyy --:-- --

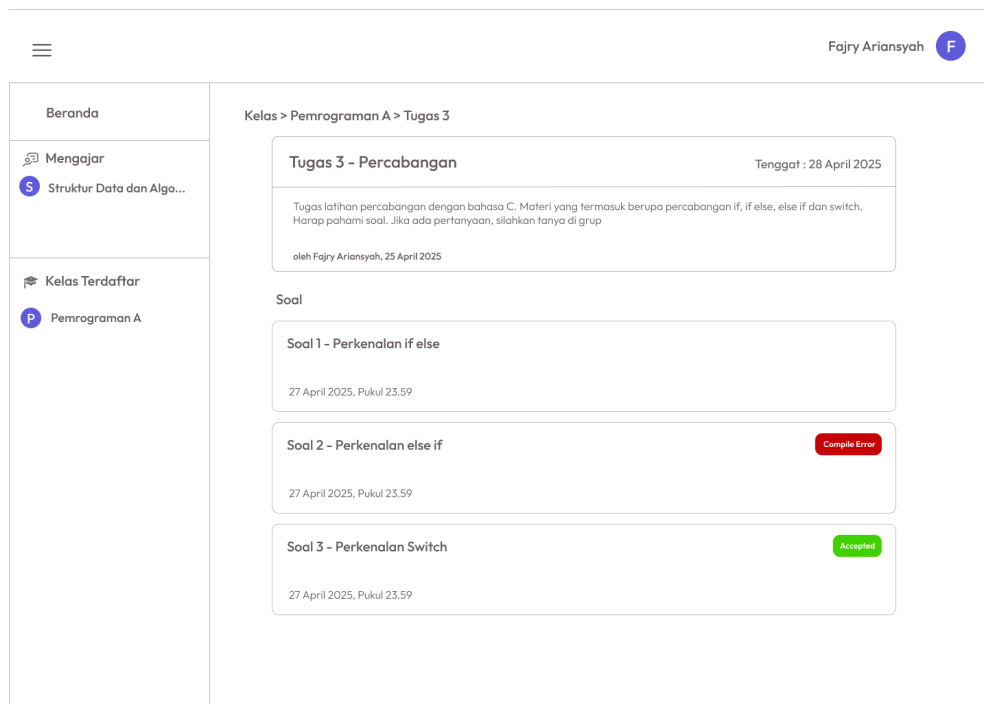
Simpan & Lanjut

Gambar 3.9: Pembuatan Tugas dari Soal yang Tersedia

24



Gambar 3.10: Daftar Tugas yang Dikelola oleh Asisten



Gambar 3.11: Halaman Tugas Praktikan

Fajri Ariansyah

F

Beranda

Mengajar

Struktur Data dan Alga...

Kelas Terdaftar

Pemrograman A

Kelas > Pemrograman A > Tugas 3 > Soal 1

Submission

<- Kembali

Hasil

S soal : Soal 1 - Pengenalan If Else

S skor : 90

Waktu Rata-rata : 3 ms

Memori Rata-rata : 512 kb

User ID Submission : Fajri Ariansyah

Waktu : 3823923

Bahasa : 26 April 2025, pukul 10:32:29

C :

Detail Test Case

No	Status	Nilai
1	Accepted	10
2	Accepted	10
3	Accepted	10
4	Accepted	10
5	Accepted	10
6	Accepted	10
7	Accepted	10
8	Accepted	10
9	Accepted	10
10	Wrong Answer	0
Skor		90

Top Users

Top Score

#	Nama	Skor
1	Ariansyah	100
2	Ariansyah	100
3	M. Fajry	100

Esekusi Tercepat*

#	Nama	Skor
1	Ariansyah	3 ms
2	Fajry Ariansyah	4 ms
3	M. Fajry	4 ms

Memori Tersedikit*

#	Nama	Skor
1	Fajry Ariansyah	128 kb
2	Ariansyah	128 kb
3	M. Fajry	128 kb

*Note :
Esekusi tercepat dan memori tersedikit hanya diambil dari yang mendapatkan skor 100

Fajry Ariansyah

Beranda

Mengajar

Struktur Data dan Algo...

Kelas Terdaftar

Pemrograman A

Kelas > Pemrograman A

Pemrograman

Kelas A

Semester Ganjil 2024/2025

Anggota : 50

Tugas : 1

Tugas

Tugas 3 - Percabangan

Dihugaskan

27 April 2025, Pukul 23.59

Tugas 2 - Perulangan

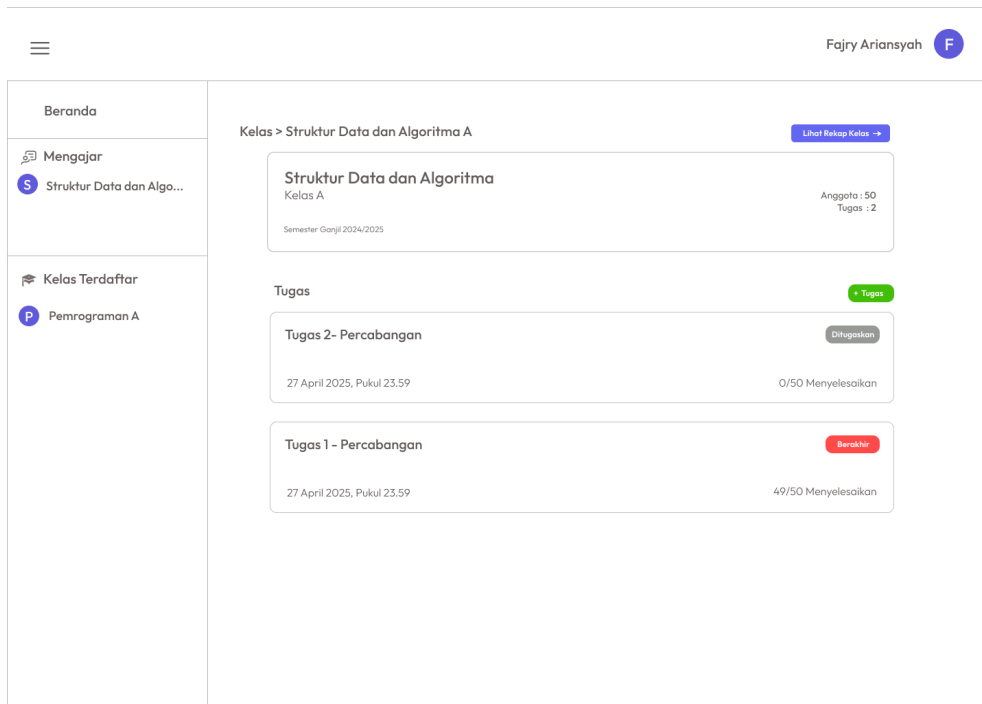
Tidak Dikumpulkan

28 April 2025, Pukul 23.59

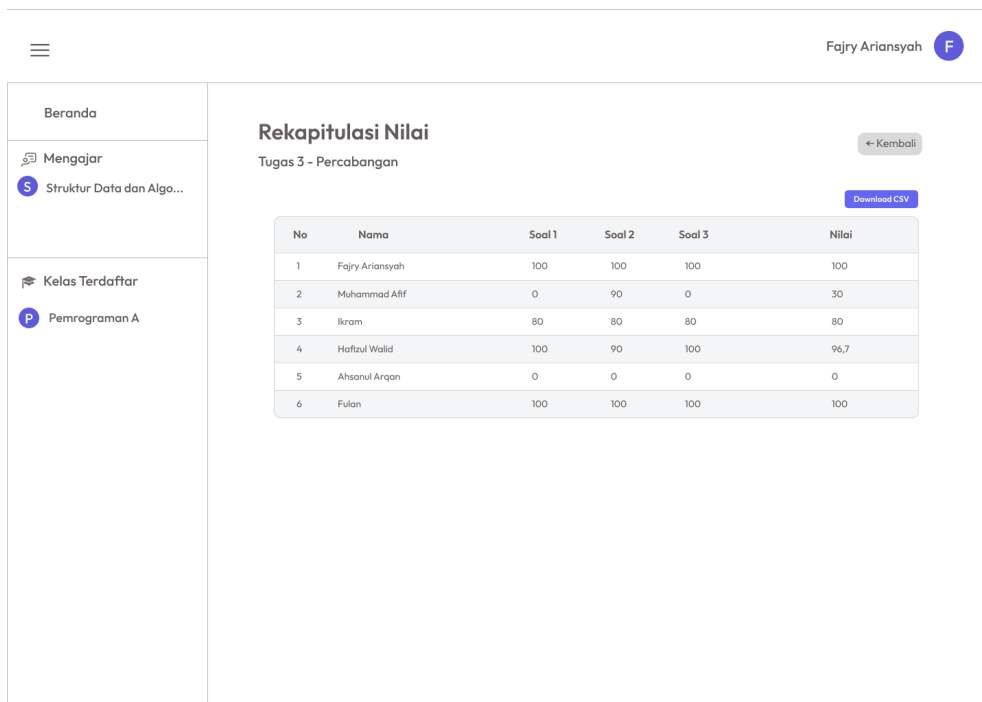
Tugas 1 - Variable dan Tipe Data

Dikumpulkan

26 April 2025, Pukul 23.59



Gambar 3.14: Kelas yang Diasuh oleh Asisten



Gambar 3.15: Rekap Nilai Berdasarkan Tugas

Fajry Ariansyah

F

Beranda

Mengajar

S Struktur Data dan Algo...

Kelas Terdaftar

P Pemrograman A

Rekapitulasi Kelas

Semua tugas

← Kembali

Download CSV

No	Nama	Tugas 1	Tugas 2	Tugas 3	Rata-rata
1	Fajry Ariansyah	100	100	100	100
2	Muhammad Alif	0	90	0	30
3	Ikrum	80	80	80	80
4	Haftzul Walid	100	90	100	96,7
5	Ahsanul Arqan	0	0	0	0
6	Fulan	100	100	100	100

Gambar 3.16: Rekap Nilai Satu Kelas

Desain ini bersifat sederhana dan bertujuan untuk memberikan gambaran awal alur interaksi dan fitur inti dalam sistem. Tampilan dapat dikembangkan lebih lanjut nantinya.

• Perancangan Backend

Backend bertugas menangani permintaan dari *frontend*, mengelola data, dan mengirimkan kode dan *test case* kepada *judge*. Pada tahap ini dilakukan perancangan *Entity Relationship Diagram* (ERD) agar mengetahui struktur entitas yang terdapat di dalam sistem nantinya. *Backend* akan bertugas sebagai pusat pengelola data, *dimana* akan *berinteraksi* langsung dengan *database*.

3.3.5 Pengembangan Sistem

Pengembangan sistem dilakukan berdasarkan hasil dari tahap perancangan sebelumnya. Metode pengembangan yang digunakan adalah *Rapid Application Development* (RAD), yang terdiri dari empat tahapan utama: perencanaan kebutuhan, desain sistem, pengembangan, dan implementasi.

Tahapan pengembangan mencakup pembuatan antarmuka pengguna, pengelolaan data melalui *backend*, serta sistem *judge* otomatis untuk mengevaluasi tugas pemrograman mahasiswa. Secara umum, sistem ini terdiri atas tiga komponen utama: *frontend*, *backend*, dan *online judge*.

- **Pengembangan Judger**

Sistem penilaian otomatis atau *online judger* dikembangkan menggunakan bahasa pemrograman *Python* untuk menjalankan dan mengevaluasi kode program mahasiswa secara otomatis berdasarkan *test case* yang telah ditentukan. Untuk mendukung integrasi dengan komponen *backend*, sistem *judger* juga dilengkapi dengan layanan *API* yang dibangun menggunakan *FastAPI*.

- **Pengembangan Frontend**

Antarmuka pengguna dikembangkan menggunakan *Next.js* dengan *TypeScript*. *Framework* ini dipilih karena kemampuannya dalam mendukung *server-side rendering (SSR)*, performa tinggi, serta ekosistem yang luas yang memungkinkan integrasi dengan berbagai pustaka dan layanan. Komponen-komponen pada antarmuka pengguna dirancang menggunakan *Tailwind CSS*.

- **Pengembangan Backend**

Backend sistem dikembangkan menggunakan *Next.js* dengan memanfaatkan fitur *API Routes* untuk membangun layanan *backend*. Untuk penyimpanan data, digunakan basis data *MySQL*. *Backend* ini menangani autentikasi pengguna, pengelolaan data tugas, dan komunikasi dengan sistem *judger* melalui *API* yang telah disediakan.

3.3.6 Pengujian dan Evaluasi

Pengujian dan evaluasi dilakukan untuk memastikan bahwa sistem yang dikembangkan berfungsi dengan baik, memiliki performa yang optimal, serta memenuhi kebutuhan pengguna. Pada tahap ini, pengujian difokuskan dengan menggunakan *Load Testing* dan *System Usability Scale (SUS)*.

3.3.6.1 Load Testing

Load testing digunakan untuk mengukur kinerja sistem saat menerima beban tinggi dan eksekusi tugas pemrograman. Pengujian ini akan menggunakan *Apache JMeter* pada server dengan mensimulasikan 100, 300, dan 500 pengguna bersamaan melakukan *login*, mengunggah tugas, dan melihat hasil. Pengukuran akan dilakukan dengan melihat *Response time*, *throughput*, *error rate*, dan penggunaan sumber daya (*CPU*, *RAM*).

3.3.6.2 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan metode evaluasi standar yang digunakan untuk mengukur kegunaan sistem berdasarkan persepsi pengguna. Pada

tahap ini, evaluasi dilakukan terhadap 10–20 partisipan yang terdiri dari asisten dan mahasiswa. Setiap partisipan akan menggunakan sistem selama 15–20 menit, kemudian diminta untuk mengisi kuesioner SUS.

Kuesioner SUS terdiri dari 10 pernyataan yang harus dijawab menggunakan skala Likert 5 poin, yaitu:

- 1 = Sangat Tidak Setuju
- 2 = Tidak Setuju
- 3 = Netral
- 4 = Setuju
- 5 = Sangat Setuju

Daftar pertanyaan SUS yang digunakan untuk mengevaluasi sistem dapat dilihat pada Tabel 3.2.

Tabel 3.2: Pertanyaan SUS untuk Sistem Penilaian Otomatis

No	Pernyataan
1	Sistem ini mudah digunakan untuk mengunggah dan menilai kode.
2	Sistem ini terlalu rumit.*
3	Saya merasa percaya diri menggunakan sistem ini tanpa bantuan.
4	Saya membutuhkan bantuan teknis untuk dapat menggunakan sistem ini.*
5	Fungsi-fungsi dalam sistem ini terintegrasi dengan baik.
6	Antarmuka sistem ini terasa tidak konsisten.*
7	Kebanyakan orang akan dengan cepat dapat mempelajari penggunaan sistem ini.
8	Sistem ini terasa tidak praktis untuk digunakan dalam praktikum.*
9	Kecepatan respon sistem ini memuaskan.
10	Sistem ini memerlukan waktu lama untuk dipelajari.*

Catatan: *Pernyataan negatif memerlukan pembalikan skor dalam perhitungan.

Perhitungan Skor SUS:

Untuk menghitung skor SUS, langkah-langkah berikut dilakukan:

1. Untuk pernyataan bernomor ganjil (positif), skor dikurangi dengan 1.

2. Untuk pernyataan bernomor genap (negatif), skor dihitung dengan cara 5 dikurangi skor jawaban.
3. Semua skor dijumlahkan.

Secara matematis, skor SUS dapat dihitung dengan rumus:

$$\text{Skor SUS} = (\text{Total skor setelah penyesuaian}) \times 2,5$$

Interpretasi dari skor SUS dapat dilihat pada Tabel 3.3.

Tabel 3.3: Interpretasi Skor SUS

Skor SUS	Grade	Keterangan
0–25	F	Sangat Buruk, Tidak Dapat Diterima
25–39	F	Buruk, Tidak Dapat Diterima
39–52	D	Cukup, Marginal
52–62	C	Cukup, Marginal
62–73	B	Baik, Dapat Diterima
73–85	A	Sangat Baik, Dapat Diterima
85–100	A	Luar Biasa, Dapat Diterima

Dengan menggabungkan metode *Load Testing* dan evaluasi *System Usability Scale (SUS)*, sistem diuji tidak hanya dari aspek teknis tetapi juga dari aspek kepuasan dan persepsi pengguna. Hal ini memastikan bahwa sistem memiliki performa yang optimal dan tingkat kegunaan yang tinggi dalam lingkungan akademik.

DAFTAR KEPUSTAKAAN

- Abdiansah, A., Utami, A. S., Yusliani, N., Miraswan, K. J., & Wedhasmara, A. (2023). Peningkatan motivasi belajar siswa sma melalui pendekatan pemrograman komputer. *Jurnal Pengabdian Kolaborasi dan Inovasi IPTEKS*, 1(4), 319–328. <https://doi.org/10.59407/jpki2.v1i4.56>
- Afiifah, K., Azzahra, Z., & Anggoro, A. (2022). Analisis teknik entity-relationship diagram dalam perancangan database sebuah literature review. *INTECH*, 3, 18–22. <https://doi.org/10.54895/intech.v3i2.1682>
- Aryanti, R., Fitriani, E., Ardiansyah, D., & Saepudin, A. (2021). Penerapan metode rapid application development dalam pengembangan sistem informasi akademik berbasis web. *Paradigma: Jurnal Komputer dan Informatika*, 23, 249–254. <https://doi.org/10.31294/p.v23i2.11170>
- bin Uzayr, S. (Ed.). (2022). *Mastering mysql for web: A beginner's guide* (1st). CRC Press. <https://doi.org/10.1201/9781003229629>
- Efendi, Y. (2022). Buku ajar pendidikan algoritma dan struktur data.
- Erinle, B. (2017). Performance testing with jmeter 3 [Book reference on JMeter for load testing].
- Fielding, R. (2000). Rest: Principles, properties, and constraints. *Proceedings of the International Conference on Web Services*, 1(1), 50–65.
- Golmohammadi, A., Zhang, M., & Arcuri, A. (2022). Testing restful apis: A survey. *ACM Transactions on Software Engineering and Methodology*, 33, 1–41. <https://doi.org/10.1145/3617175>
- Hendayun, M., Ginanjar, A., & Ihsan, Y. (2023). Analysis of application performance testing using load testing and stress testing methods in api service. *Jurnal Sisfotek Global*, 13(1), 28–34. <https://doi.org/10.38101/sisfotek.v13i1.2656>
- Kafai, Y., et al. (2020). Turning bugs into learning opportunities: Understanding debugging processes, perspectives, and pedagogies. *International Society of the Learning Sciences (ISLS) Annual Conference*, 374–375. <https://doi.org/https://doi.dx.org/10.22318/icls2020.374>
- Kim, J., et al. (2021). Api economy: The role of web services in modern software development. *Journal of Software Engineering and Applications*, 14(3), 267–290.
- Kumar, R., Rawat, R. S., Prajapat, V., & Choudhary, V. (2023). Software engineering and programming. *Journal of Nonlinear Analysis and Optimization*, 14(1), 102–108. <https://doi.org/10.36893/JNAO.2023.V14I1.102-108>
- Lee, M., et al. (2022). Performance optimization in web development using next.js. *Journal of Web Technologies*, 15(2), 88–105.
- Liang, J.-C., Wang, J.-Y., & Chou, S.-H. (2020). A designed platform for programming courses. *2020 9th International Congress on Advanced Applied Informatics (IIAI-AAI)*, 290–293.

- Mustamiin, M., Suheryadi, A., & Puspaningrum, A. (2022). Penerapan sistem evaluasi pembelajaran pemrograman terintegrasi dengan online judge di smk. 2, 61–67. <https://doi.org/10.46961/jpk.v2i2.636>
- Nuriman, M., & Mayesti, N. (2020). Evaluasi ketergunaan website perpustakaan universitas indonesia menggunakan system usability scale. *BACA: JURNAL DOKUMENTASI DAN INFORMASI*, 41, 253. <https://doi.org/10.14203/j.baca.v4i1i2.622>
- Patel, N., et al. (2021). Online judge systems and their impact on competitive programming. *International Journal of Computer Science Education*, 18(2), 112–130.
- Ramirez, S. (2021). Fastapi: A modern, fast (and asynchronous) web framework for python. *International Journal of Software Engineering*, 18(4), 135–150.
- Ramirez-Echeverry, J., Restrepo-Calle, F., & González, F. (2018). Uncode: Interactive system for learning and automatic evaluation of computer programming skills. <https://doi.org/10.21125/edulearn.2018.1632>
- Sondang. (2024). Penerapan metode rad dalam pengembangan sistem informasi pemesanan jasa percetakan berbasis web pada percetakan karya sehat jaya. *Riset dan E-Jurnal Manajemen Informatika Komputer*, 8, 871–879.
- Sukendar, A. (2023). Perancangan sistem penilaian otomatis program java berbasis web pada perkuliahan pemrograman. *Device*, 13, 216–222. <https://doi.org/10.32699/device.v13i2.5841>
- Thuné, M., & Eckerdal, A. (2018). Analysis of students' learning of computer programming in a computer laboratory context. *European Journal of Engineering Education*, 44, 1–3. <https://doi.org/10.1080/03043797.2018.1544609>
- Universitas Syiah Kuala. (2024). Data jumlah mahasiswa program studi informatika pada semester ganjil 2024/2025 [Diakses April 2025].
- Xu, L., et al. (2022). Analyzing the effectiveness of online judge systems in learning algorithms. *ACM Transactions on Computing Education*, 20(1), 75–98.
- Yudhistira, A. (2023). Perancangan dan implementasi sistem informasi perpustakaan berbasis web. *Jurnal Sistem Informasi dan Komputerisasi*. <https://doi.org/https://doi.org/10.56291/jsk.v7i1.95>