

Sistem Pengelolaan Data Program Studi Independen Berbasis Web di Infinite Learning

Web-Based Independent Study Program Data Management System at Infinite Learning

Olivia Ristami Nainggolan¹, Metta Santiputri²

^{1,2}Program Studi Teknik Informatika, Politeknik Negeri Batam

E-mail: ¹oliviaristamin@gmail.com, ²metta@polibatam.ac.id

Abstrak

Pendataan yang masih manual dalam pelaksanaan studi independen program Web Development di Infinite Learning telah menjadi masalah signifikan yang menyulitkan pengguna, sehingga menghambat efektivitas pengelolaan data dan pemantauan kinerja peserta program. Oleh karena itu, penelitian ini mengusulkan pengembangan sistem pengelolaan data berbasis digital untuk mengatasi masalah tersebut. Topik penelitian ini dipilih karena kebutuhan mendesak untuk meningkatkan digitalisasi dalam pengelolaan data pada program studi independen. Dengan adanya sistem pengelolaan data yang terintegrasi, diharapkan dapat mempersingkat waktu pengguna dalam mengelola dan mengevaluasi perkembangan akademik, serta menyediakan informasi. Metode penelitian yang digunakan adalah metode *Waterfall*. Model *Waterfall* dipilih karena strukturnya yang sistematis dan terstruktur, memungkinkan pengembangan sistem dilakukan secara bertahap mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian dan pemeliharaan. Hasil penelitian menunjukkan bahwa sistem pengelolaan data berhasil dikembangkan dengan menggunakan teknologi yang mencakup penggunaan MySQL sebagai *database*, Bahasa pemrograman JavaScript, ReactJs sebagai *library* dan ExpressJs sebagai *framework* untuk *back-end*. Kesimpulannya, seluruh fitur yang direncanakan telah berhasil diimplementasikan sesuai dengan perancangan yang telah dibuat. Hasil penelitian ini diharapkan dapat menjadi referensi bagi pengembangan sistem serupa di masa depan.

Kata kunci: pengelolaan data, waterfall, informasi, digitalisasi

Abstract

The manual data management in the implementation of the independent study program in Web Development at Infinite Learning has become a significant problem, making it difficult for users and hindering the effectiveness of data management and participant performance monitoring. Therefore, this study proposes the development of a digital data management system to address this issue. The topic of this research was chosen due to the urgent need to enhance digitalization in data management within the independent study program. With an integrated data management system in place, it is expected to shorten the time users spend managing and evaluating academic progress, as well as providing information more efficiently. The research method used is the Waterfall model. The Waterfall model was chosen because of its systematic and structured approach, which allows for the development of the system to be carried out in stages, from requirements analysis, design, implementation, to testing and maintenance. The research results show that the data management system has been successfully developed using technologies that include MySQL as the database, JavaScript programming language, ReactJS as the library, and ExpressJS as the framework for the back-end. In conclusion, all planned features have been successfully implemented according to the design. The findings of this study are expected to serve as a reference for the development of similar systems in the future.

Keywords: data management, waterfall, information, digitalization

1. PENDAHULUAN

Infinite Learning, sebagai sub-unit PT Kinema Systrans Multimedia yang bergerak di bidang teknologi dan pendidikan memainkan peran penting sebagai mitra Kampus Merdeka dalam melaksanakan programnya, salah satunya yaitu Studi Independen Merdeka Belajar Kampus Merdeka yang memberikan kesempatan bagi mahasiswa dari berbagai universitas untuk mengikuti program studi independen secara mandiri, memungkinkan mereka untuk belajar sesuai dengan minat khususnya pada program *web development*. Namun, dalam implementasi program ini Infinite Learning dihadapkan pada sejumlah tantangan krusial yang memerlukan penyelesaian menggunakan teknologi, berada dalam posisi yang strategis untuk menghadirkan inovasi dalam ranah pendidikan melalui solusi pembelajaran berbasis teknologi. Dalam pelaksanaan studi independen pada program *web development*, Infinite Learning memiliki 800 *mentee* yang didukung oleh 13 mentor *technical*, 11 mentor desain dan 10 mentor profesional. Dalam 1 *batch*, *mentee* memiliki beberapa tugas yang dibagi menjadi tugas profesional, tugas desain, dan tugas *technical*. Tugas yang dikumpulkan akan dinilai oleh mentor sesuai bagian, sehingga setiap mentor dapat mengakses data seluruh *mentee*.

Permasalahan utama yang dihadapi berkaitan dengan pengelolaan data. Saat ini, Infinite Learning masih menggunakan Airtable, sebuah aplikasi basis data berbasis *cloud* yang memberikan alat untuk mengumpulkan dan mengelola data. Seluruh proses, seperti pengelolaan data *mentee*, absensi, pengajuan pengunduran diri, ketidakhadiran, dan pengumpulan tugas masih dilakukan secara manual menggunakan Airtable. Situasi ini mengakibatkan rendahnya tingkat organisasi dalam manajemen data program studi independen. Selain itu, terdapat risiko hilangnya data karena setiap mentor memiliki akses ke Airtable ini.

Salah satu penelitian serupa dilakukan oleh [1] dalam judul “Analisa dan Perancangan Sistem Informasi Sekolah SMK Bina Putra Mandiri berbasis Website Menggunakan Framework Laravel”. Dalam penelitian disebutkan bahwa semua catatan administratif siswa SMK Bina Putra Mandiri masih mengandalkan proses manual. Oleh karena itu diperlukan sebuah sistem untuk membantu staf sekolah yang bertugas dalam manajemen data siswa dan staf sekolah dengan cepat dan tepat.

Dalam [2] juga dilakukan penelitian terhadap manajemen data dengan judul “Analisis Perancangan Sistem Informasi Perpustakaan Menggunakan Model Waterfall dalam Peningkatan Inovasi Teknologi.” Dalam penelitian ini disebutkan bahwa pada perpustakaan manajemen data yang masih menggunakan metode konvensional dapat menyebabkan berbagai permasalahan seperti proses pencarian data buku dilakukan dengan membuka kembali buku persediaan per halaman sehingga membutuhkan waktu yang cukup lama. Selain itu, proses pembuatan laporan rekapitulasi data buku dan daftar laporan peminjam buku juga harus memeriksa per halaman dari dalam buku rekapitulasi dan menyalinnya kembali. Sehingga membutuhkan waktu yang lama dalam melakukan rekap dan menyusun laporan. Maka dirancanglah pembuatan Analisis Perancangan Sistem Informasi Perpustakaan Menggunakan Model *Waterfall* dalam Peningkatan Inovasi Teknologi. Penggunaan metode pengembangan *waterfall* dapat mengatasi masalah pengelolaan data perpustakaan yang masih menggunakan cara konvensional karena telah memiliki fitur-fitur yang disesuaikan.

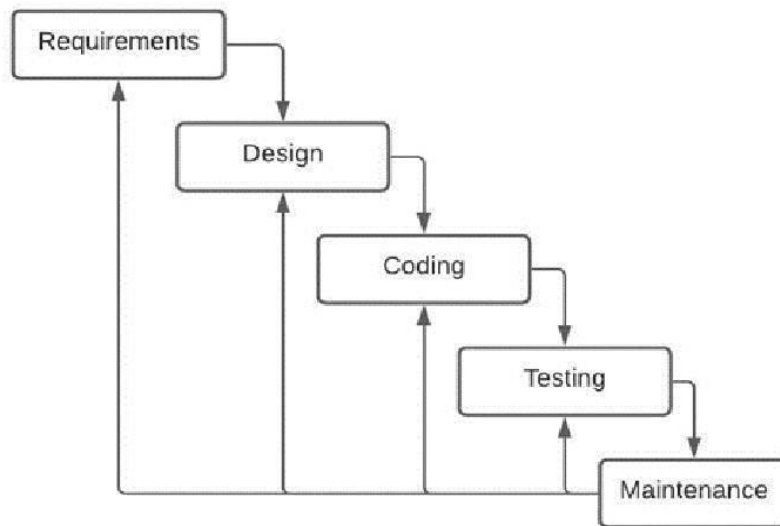
Penelitian yang serupa juga dilakukan oleh [3] yang berjudul “Pembangunan Aplikasi Pengelolaan Data pada Rumah Sakit Berbasis Web”. Dalam penelitian ini disebutkan bahwa masalah yang terjadi di Rumah Sakit Kopo Bandung adalah masih manualnya penyimpanan data-data Rumah Sakit menggunakan Microsoft Excel, dan seiring berjalannya waktu data-data semakin banyak bertambah. Hal tersebut menyulitkan para pegawai ketika mengelola data-data tersebut dan mencari arsip yang dibutuhkan karena *file* Excel yang digunakan belum terorganisir dengan baik. Dari permasalahan yang ada maka dirancanglah pembuatan web

sistem pengelolaan data rumah sakit yang diharapkan dapat memudahkan pegawai rumah sakit.

Berdasarkan temuan-temuan di atas, kesimpulan yang dapat diambil adalah pendataan yang masih manual cukup menyulitkan para pengguna dan memakan waktu yang cukup lama karena prosesnya yang masih manual. Sehingga tugas akhir ini mengusulkan pengembangan sistem pengelolaan data program Studi Independen berbasis web di Infinite Learning dengan tujuan mempermudah pengguna serta meningkatkan efisiensi pelaksanaan studi independen yang dapat membantu memonitor dan mengevaluasi studi akademik, mengelola data *mentee*, mencatat absensi, dan menyediakan informasi nilai. Pendekatan metode pengembangan seperti *Software Development Life Cycle* (SDLC) atau model *waterfall* yang telah terbukti berhasil dalam penelitian terdahulu dapat menjadi dasar untuk merancang sistem ini.

2. METODE PENELITIAN

Metode yang akan digunakan dalam membuat sistem pengelolaan data ini yaitu *waterfall*. Metode ini menyusun langkah-langkah secara berurutan mulai dari analisis, desain, implementasi, pengujian, hingga perbaikan yang memungkinkan pengembang untuk memahami secara mendalam setiap tahapan dalam pengembangan sistem dengan lebih terstruktur [4]. Alasan penggunaan metode *waterfall* adalah karena pendekatannya yang berurutan, dimulai dari wawancara dengan admin, *mentee*, dan mentor. Pendekatan ini cocok untuk proyek seperti sistem manajemen data program studi independen yang memiliki kebutuhan yang stabil dan jelas, memungkinkan pemantauan yang terstruktur terhadap kemajuan proyek dan pengelolaan yang efektif dari awal hingga akhir proyek. Metode *waterfall* dapat dilihat pada Gambar 1.



Gambar 1. Metode *waterfall*

2.1 Requirements

Proses dimulai dengan menyusun konteks masalah yang akan diselidiki, merumuskan permasalahan dan mengidentifikasi batasan-batasan yang akan ditetapkan, serta melakukan penelitian terhadap objek penelitian [5]. Dalam studi ini, awalnya adalah mencari permasalahan yang muncul dalam perencanaan Sistem Pengelolaan Data Program Studi Independen berbasis Web di Infinite Learning melalui wawancara dengan admin, mentor dan *mentee* untuk mengumpulkan informasi terkait apa saja kebutuhan *website* yang akan dibuat. Metode yang digunakan adalah wawancara dengan tiap *role*, sehingga menghasilkan data-data.

2.2 Design

Langkah berikutnya adalah tahap *design*. *Design* dilaksanakan sebelum memulai proses pengkodean dengan tujuan menyajikan gambaran komprehensif mengenai tugas yang harus diselesaikan dan bagaimana sistem yang diinginkan akan terlibat. Ini membantu dalam menentukan secara spesifik kebutuhan perangkat keras dan sistem, serta meurmuskan secara menyeluruh arsitektur sistem yang akan dibuat [6]. Pada tahap ini, peneliti melakukan proses *design* sistem meliputi *design interface* dan perancangan basis data serta menghasilkan luaran berupa, Gambaran Umum Sistem, Diagram *Use Case*, dan *Class Diagram*, ERD dan desain antarmuka pengguna.

2.3 Development

Setelah proses desain selesai, Langkah selanjutnya adalah menerapkan *website* yang telah dibuat. Dalam menggunakan *website* ini, perhatian utama terletak pada konten atau informasi yang akan disampaikan melalui *website* tersebut. Tahap ini melibatkan pengkodean, yang merupakan proses mengubah desain menjadi format yang dapat dipahami oleh mesin [7]. Pada tahap ini, peneliti mulai mengimplementasikan desain sistem dengan melakukan pengkodean menggunakan Bahasa pemrograman Javascript dengan framework ReactJS dan Express, serta MySQL sebagai *database* dan menghasilkan pembuatan *website*.

2.4 Testing

Setelah sistem berhasil dikembangkan, sistem tersebut menjalani serangkaian pengujian untuk mengevaluasi kinerjanya serta memastikan tingkat optimalitasnya, memeriksa apakah sudah sesuai dengan standar yang ditetapkan atau masih memerlukan penyesuaian lebih lanjut [8]. Pada tahap ini, dilakukan *functional testing* untuk memeriksa apakah semua fitur bekerja sesuai fungsinya. Setelah dilakukan *testing*, tidak ditemukan permasalahan dan semua fitur berjalan sesuai fungsinya.

2.5 Maintenance

Saat pengguna mulai memanfaatkan sistem yang telah dirancang, tantangan yang sebenarnya muncul dan memerlukan penanganan yang teratur. Produk yang dikembangkan dipelihara dengan cermat. Dalam mengikuti perkembangan situasi atau teknologi pengguna, perangkat lunak secara berkala diperbaharui. Tujuannya adalah untuk memastikan bahwa sistem yang telah dibuat berfungsi dengan baik sambil terus mengembangkan aplikasi tersebut. Tahap pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tetapi tidak untuk membuat perangkat lunak baru [9]. Luaran dari tahap ini adalah sistem yang diperbarui secara konsisten dan disesuaikan agar tetap relevan dengan kebutuhan pengguna.

3. HASIL DAN PEMBAHASAN

3.1 *Functional Requirement*/Kebutuhan fungsional

Functional Requirements atau kebutuhan fungsional memuat gambaran perilaku yang diinginkan dari sistem, yang dapat dijelaskan sebagai layanan, tugas, atau fungsi yang harus dilaksanakan oleh sistem itu sendiri [10]. Berikut adalah hasil analisis mengenai *Functional Requirement* pada Sistem Pengelolaan Data Program Studi Independen berbasis Web di Infinite Learning:

Tabel 1. Kebutuhan Fungsional

NO	Kebutuhan Fungsional
F001	Pengguna (admin, mentor dan <i>mentee</i>) dapat melakukan <i>login</i> dengan menggunakan <i>username</i> dan <i>password</i>
F002	Mentor dan <i>mentee</i> dapat melakukan edit profil
F003	Mentor dan <i>mentee</i> dapat mengakses informasi kelas
F004	<i>Mentee</i> dapat melihat mentor pribadi
F005	<i>Mentee</i> dapat mengajukan izin
F006	<i>Mentee</i> dapat mengajukan pengunduran diri
F007	<i>Mentee</i> dapat mengumpulkan tugas
F008	<i>Mentee</i> dapat mengedit dan menghapus pengumpulan tugas
F009	<i>Mentee</i> dapat melihat absensi kehadiran
F010	Mentor dapat melihat pengumpulan tugas <i>mentee</i>
F011	Mentor dapat melihat data <i>mentee</i>
F012	Mentor dapat mengelola nilai <i>mentee</i>
F013	Mentor dapat mengelola absensi <i>mentee</i>
F014	Admin dapat mengelola data <i>mentee</i> , mentor dan universitas
F015	Admin dapat memvalidasi pengajuan pengunduran diri <i>mentee</i>
F016	Sistem dapat menampilkan data <i>mentee</i> , mentor dan universitas
F017	Sistem dapat menampilkan ajuan pengunduran diri dari <i>mentee</i>
F018	Sistem dapat menampilkan informasi kelas
F009	Sistem dapat menampilkan pengumpulan tugas <i>mentee</i>
F020	Sistem dapat menampilkan data <i>mentee</i>
F021	Sistem dapat menampilkan rekap absensi <i>mentee</i>
F022	Sistem dapat menampilkan nilai <i>mentee</i>
F023	Sistem dapat menampilkan mentor pribadi
F024	Sistem dapat menampilkan tempat pengajuan izin
F025	Sistem dapat menampilkan tugas dan tempat pengumpulan tugas

3.2 Non-functional Requirement/Kebutuhan non-fungsional

Kriteria kualitas atau kinerja yang harus dipenuhi oleh sistem perangkat lunak, dikenal sebagai kebutuhan nonfungsional (NFR), merupakan elemen kunci yang memerlukan perhatian selama proses pengembangan.

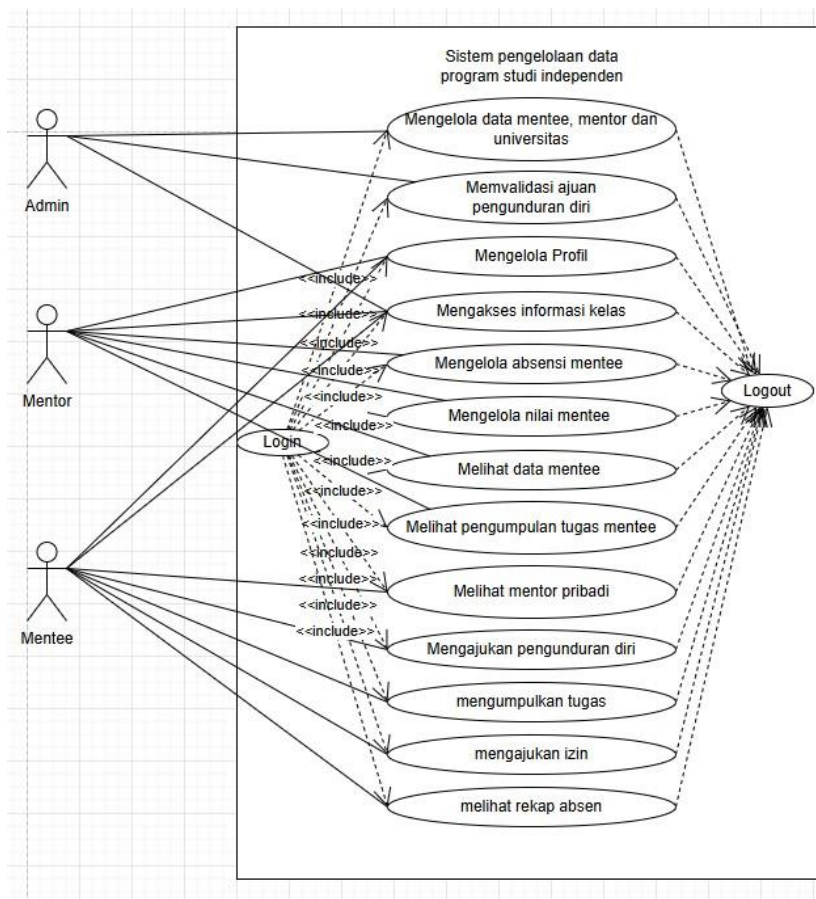
Tabel 2. Kebutuhan Non-fungsional

Kriteria	Parameter
<i>Availability</i>	Sistem dapat digunakan kapan saja
<i>Ergonomy</i>	Sistem mudah dipahami dan digunakan
Bahasa	Sistem menggunakan Bahasa Indonesia

3.3 Use Case Diagram

Dalam tahap ini, akan dijelaskan mengenai interaksi antara pengguna dan aplikasi dengan menggunakan *Unified Modeling Language* (UML). *Unified Modeling Language* (UML) digunakan dalam rekayasa perangkat lunak untuk mengembangkan suatu sistem. UML merupakan bahasa visual yang digunakan untuk mendefinisikan dan mendokumentasikan sistem. Dengan UML, persyaratan dalam skenario yang menjelaskan bagaimana pengguna menggunakan sistem dapat diungkapkan, dan batasan sistem juga dapat dinyatakan dengan jelas [11].

Dalam Gambar 2, *user* berinteraksi langsung dengan sistem yaitu dapat melakukan *login*, mengelola profil dan melihat informasi kelas, dan *logout*. *Mentee* dapat melihat mentor pribadi, juga dapat melakukan pengajuan seperti izin, pengunduran diri, dan mengumpulkan tugas. Mentor dapat mengelola absensi *mentee*, mengelola nilai *mentee*, melihat data *mentee*, dan melihat pengumpulan tugas *mentee*. Kemudian admin dapat mengelola data mentor dan *mentee* serta melakukan validasi ajuan pengunduran diri *mentee*.



Gambar 2. Use Case Diagram

3.4 ER Diagram

Model data *Entity Relationship* (ER) diagram memungkinkan penggambaran visual dari entitas dan hubungan data yang ada di lingkungan dunia nyata perusahaan. Penerapan model ini umumnya digunakan dalam perancangan awal basis data, menyajikan konsep-konsep yang berguna untuk mengubah deskripsi informal kebutuhan pengguna terhadap basis data mereka menjadi deskripsi yang lebih terperinci dan tepat untuk Untuk menggambarkan desain sistem [12]. ER Diagram Sistem Pengelolaan Data Program Studi Independen dapat dilihat pada Gambar 3.

3.6 Hasil Implementasi

1. Halaman Utama

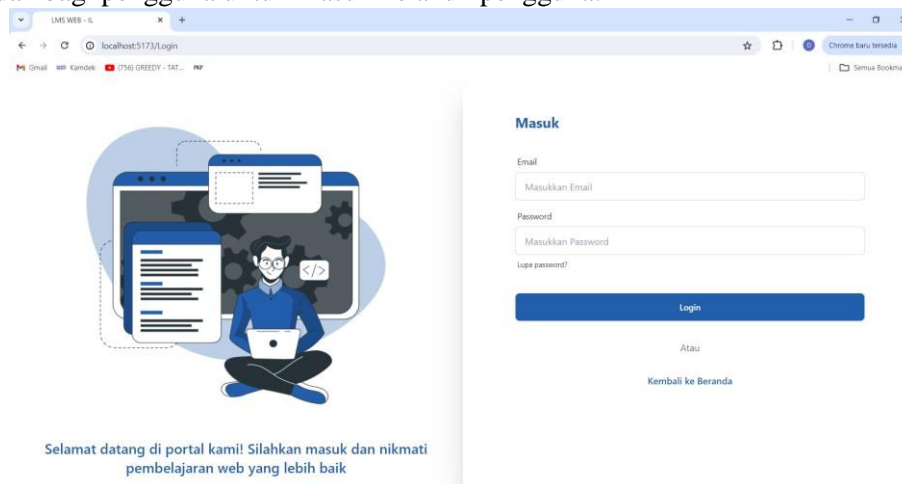
Halaman utama merupakan halaman pertama yang dilihat oleh pengguna ketika mengunjungi situs web. Pada halaman ini, berbagai elemen penting telah dirancang dengan tujuan menarik perhatian pengguna, memberikan informasi yang jelas, dan memandu mereka ke bagian lain dari situs.



Gambar 5. Halaman Utama

2. Halaman Login

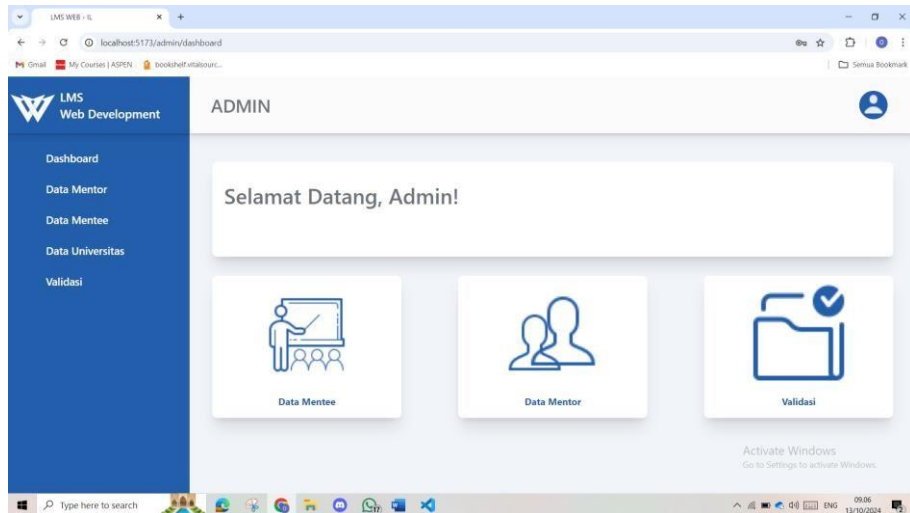
Halaman *login* adalah salah satu komponen kunci dari situs web yang memerlukan autentikasi pengguna. Halaman ini dirancang untuk memberikan akses yang aman dan mudah bagi pengguna untuk masuk ke akun pengguna.



Gambar 6. Halaman Login

3. Halaman Dashboard Admin

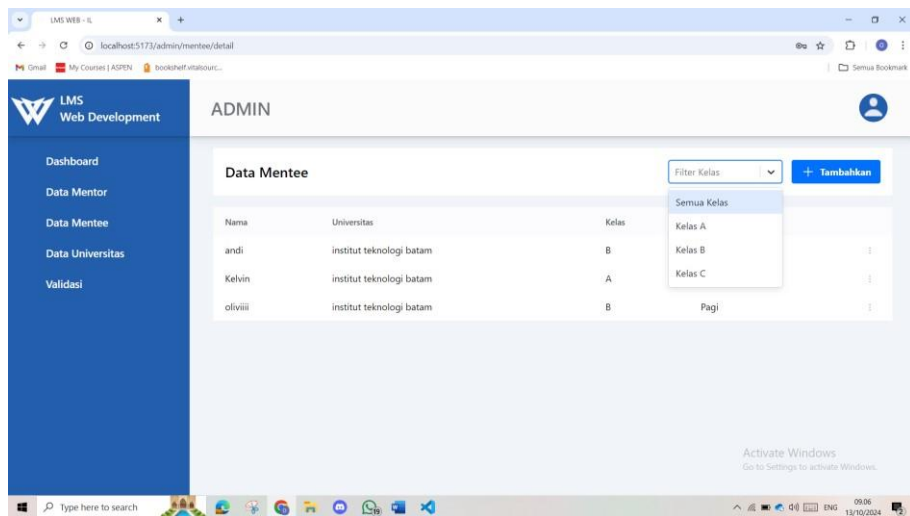
Halaman *dashboard* admin adalah pusat kendali utama bagi administrator situs web untuk mengelola dan memantau berbagai aspek operasional. *Dashboard* ini dirancang untuk memberikan akses mudah ke data penting dan alat administrasi yang diperlukan. Adapun fitur-fiturnya adalah data *mentee*, data *user*, data universitas dan validasi pengunduran diri.



Gambar 7. Halaman *Dashboard Admin*

4. Halaman *Data Mentee*

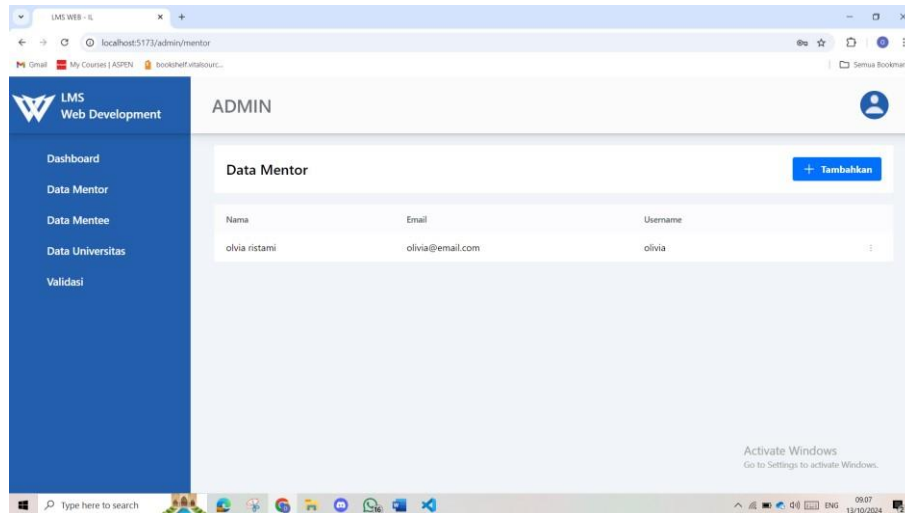
Halaman data *mentee* adalah fitur yang memungkinkan administrator untuk mengelola dan memantau informasi semua *mentee* yang terdaftar dalam sistem. Fitur ini dirancang untuk memberikan akses yang komprehensif dan mudah terhadap data *mentee*.



Gambar 8. Halaman *Data Mentee*

5. Halaman *Data Mentor*

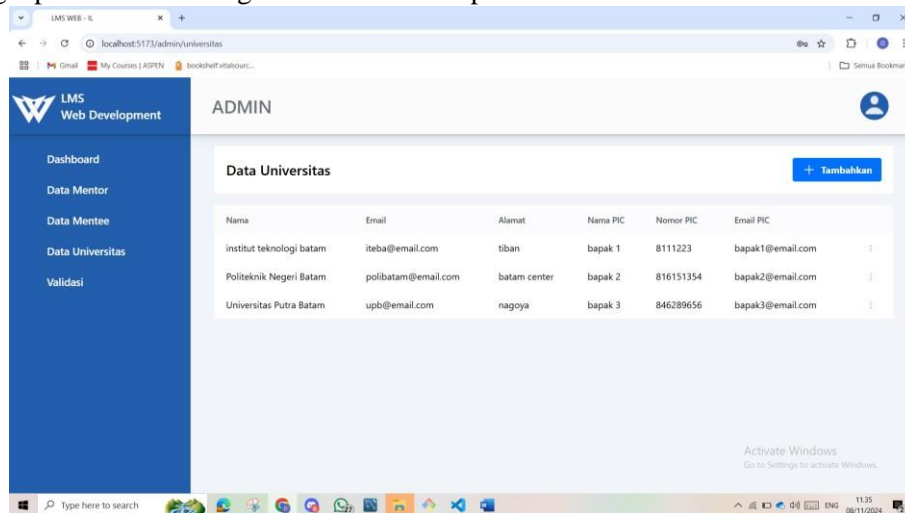
Halaman data mentor adalah fitur yang memungkinkan admin dengan mudah untuk melihat daftar lengkap mentor yang terdaftar dalam sistem. Selain itu admin juga dapat menambahkan, mengedit, dan menghapus data mentor. Ini akan mendukung proses manajemen dan pemeliharaan data mentor dengan lebih efektif.



Gambar 9. Halaman Data Mentor

6. Halaman Universitas

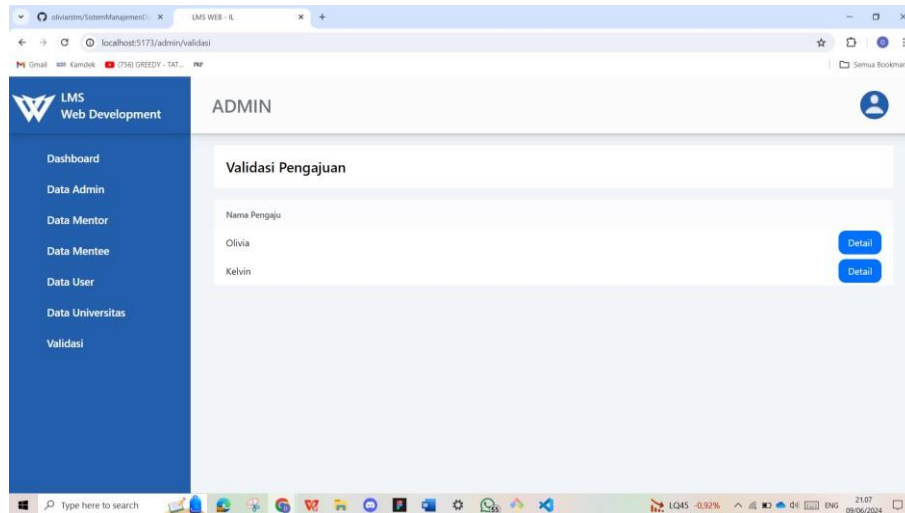
Fitur detail universitas memainkan peran penting dalam memberikan informasi lengkap dan akurat mengenai universitas kepada administrator.



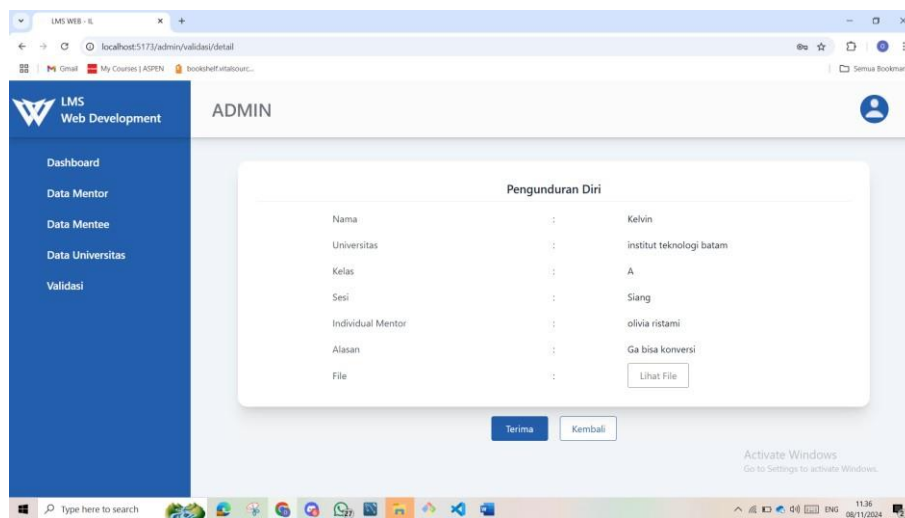
Gambar 10. Halaman Universitas

7. Halaman Validasi Pengunduran diri *mentee*

Halaman validasi pengunduran diri mentee adalah komponen penting dalam manajemen mentee, memastikan bahwa semua permohonan pengunduran diri ditangani dengan adil dan efisien.



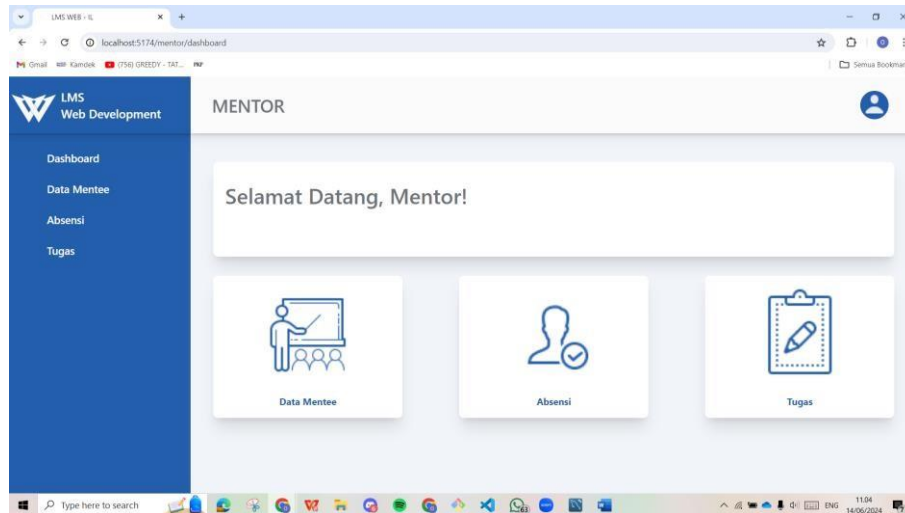
Gambar 11. Halaman Validasi Pengunduran Diri



Gambar 12. Halaman Validasi Pengunduran Diri *mentee*

8. *Dashboard* Mentor

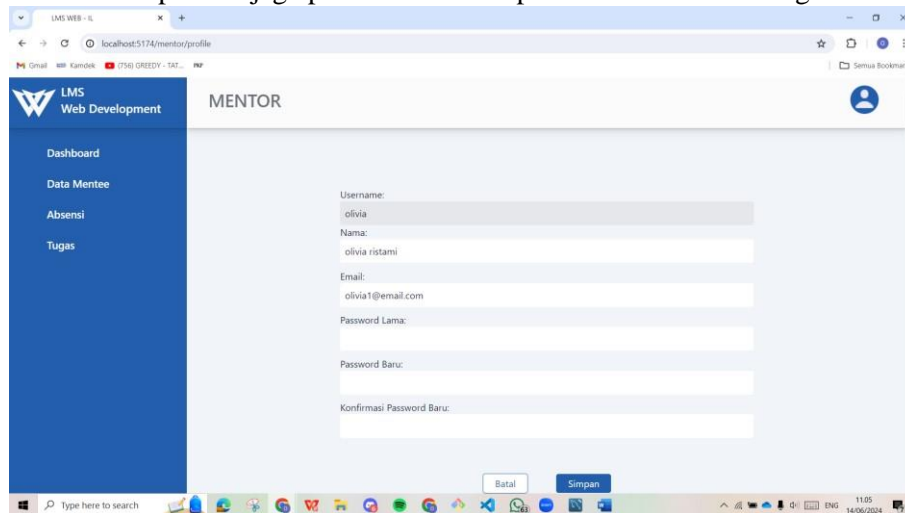
Dashboard mentor adalah fitur yang dirancang untuk memberikan akses mudah dan terorganisir bagi mentor dalam mengelola kegiatan mentoring. *Dashboard* ini menyediakan informasi penting dan alat yang dibutuhkan oleh mentor untuk menjalankan tugas mereka dengan efektif.



Gambar 13. Halaman *Dashboard* Mentor

9. Halaman Edit Profil Mentor

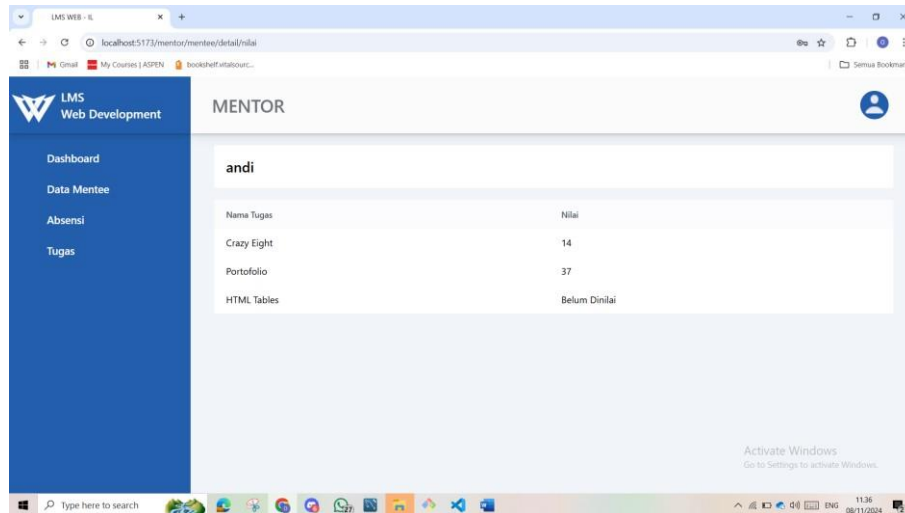
Halaman edit profil mentor adalah fitur yang memungkinkan mentor untuk memperbarui informasi pribadi dalam sistem. Fitur ini dirancang untuk memastikan bahwa mentor dapat menjaga profil mereka tetap akurat dan terkini dengan mudah.



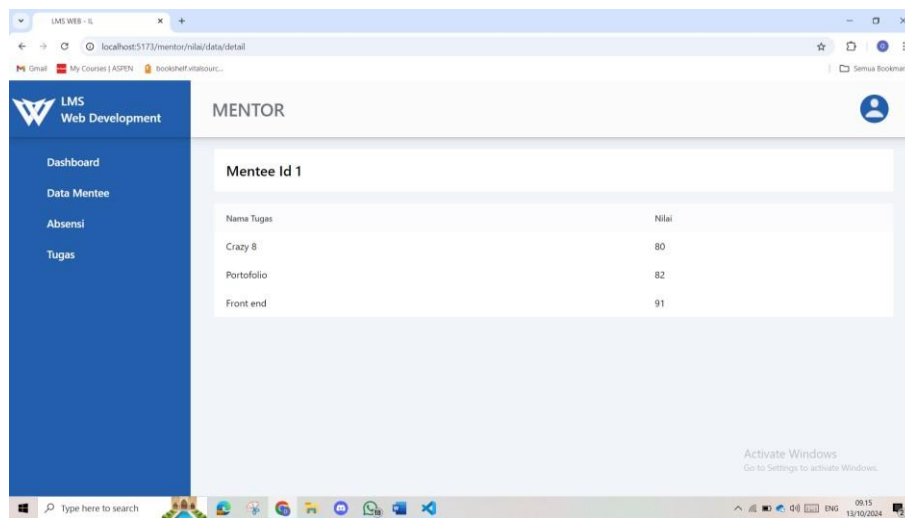
Gambar 14. Halaman Edit Profil Mentor

10. Halaman Data Nilai

Halaman Data Nilai adalah fitur yang dirancang untuk memungkinkan mentor untuk melihat nilai *mentee* dalam pembelajaran. Fitur ini menyediakan akses mudah ke data akademik, membantu mentor dalam memantau perkembangan *mentee*.



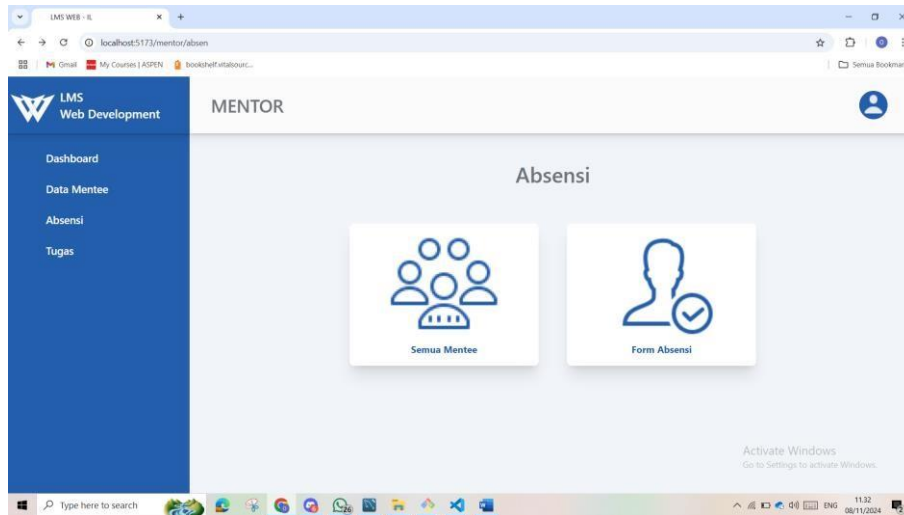
Gambar 15. Halaman Data Nilai *Mentee*



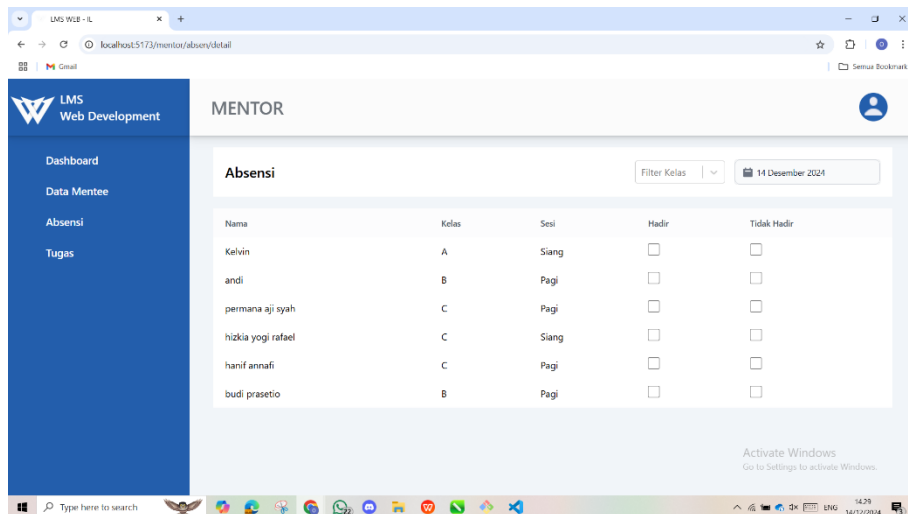
Gambar 16. Halaman Data Nilai *Mentee*

11. Halaman Absensi

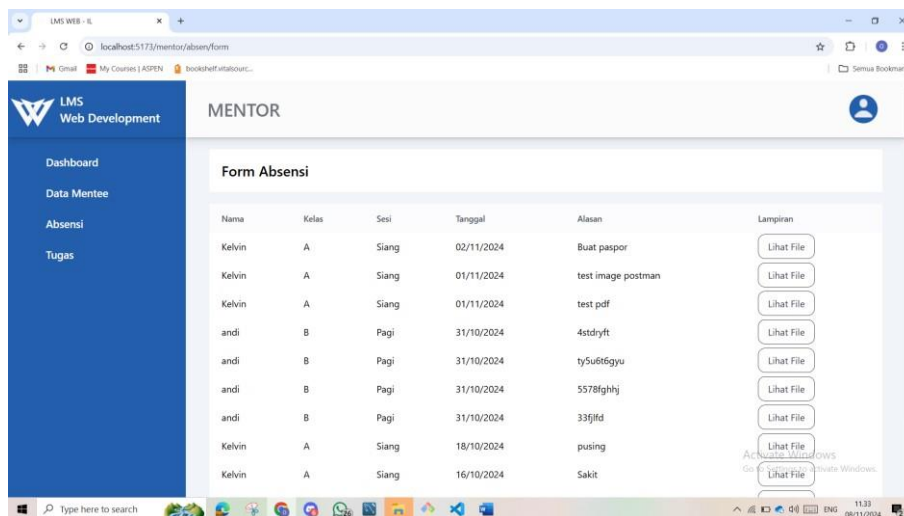
Halaman absensi adalah fitur yang dirancang untuk memungkinkan mentor memantau dan mengelola kehadiran *mentee* dalam setiap kegiatan pembelajaran. Fitur ini menyediakan akses mudah ke catatan kehadiran, membantu mentor dalam memastikan partisipasi *mentee* dan mengelola jadwal pertemuan dengan lebih efektif.



Gambar 17. Halaman Absensi



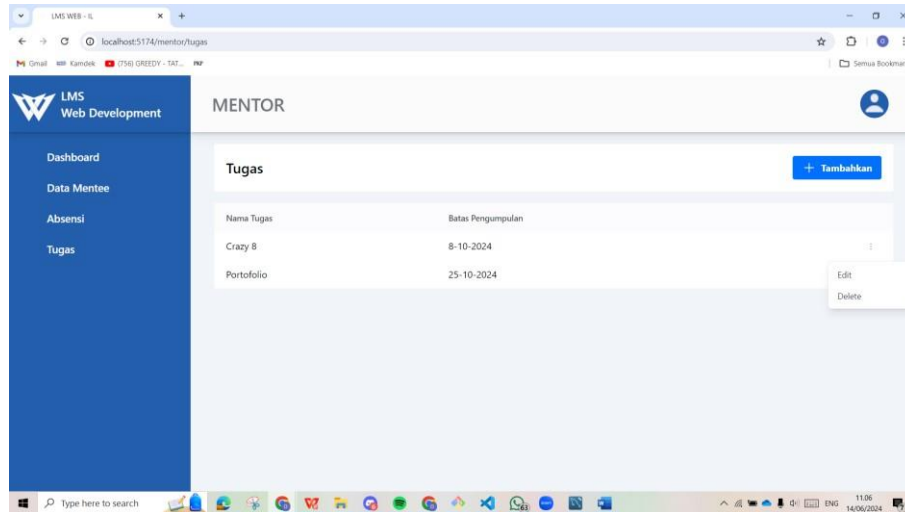
Gambar 18. Halaman Absensi



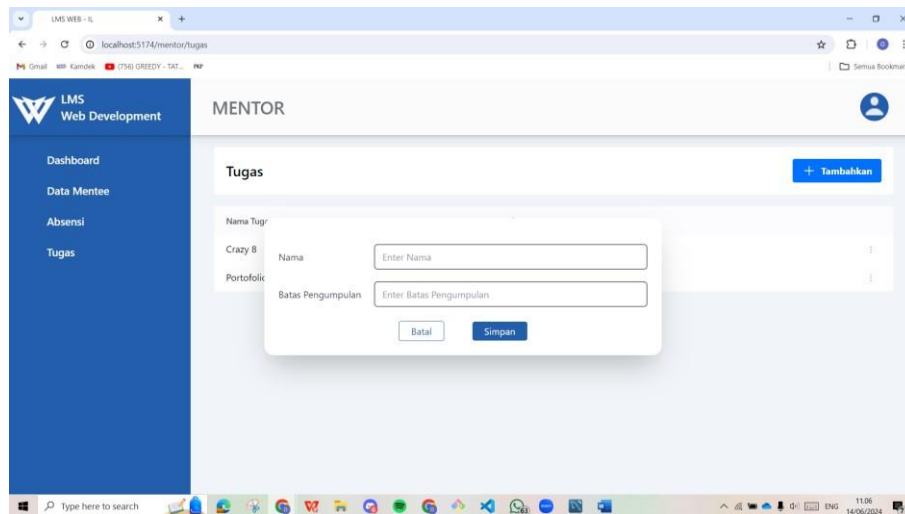
Gambar 19. Halaman Absensi

12. Halaman Tugas

Halaman tugas adalah fitur yang dirancang untuk memungkinkan mentor mengelola tugas. Fitur ini menyediakan alat yang dibutuhkan mentor untuk memantau proses tugas *mentee* dan memberikan umpan balik berupa nilai.



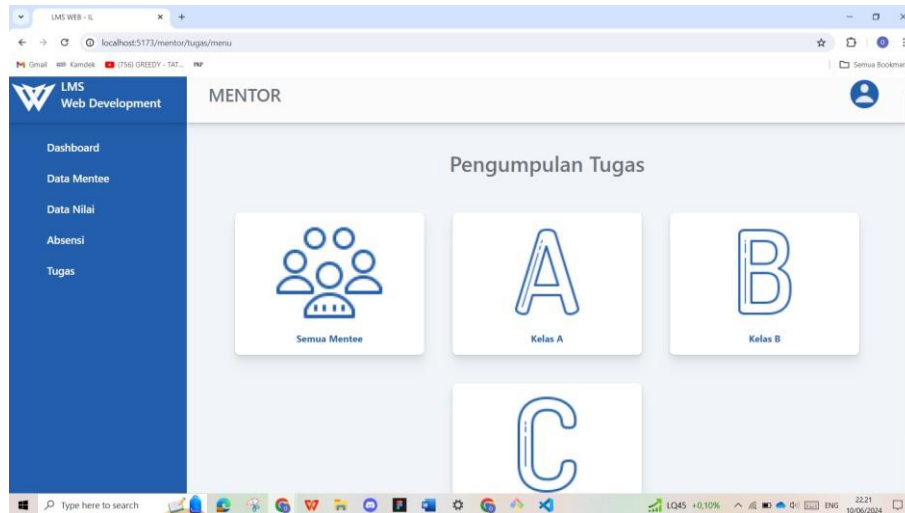
Gambar 20. Halaman Tambahkan Tugas



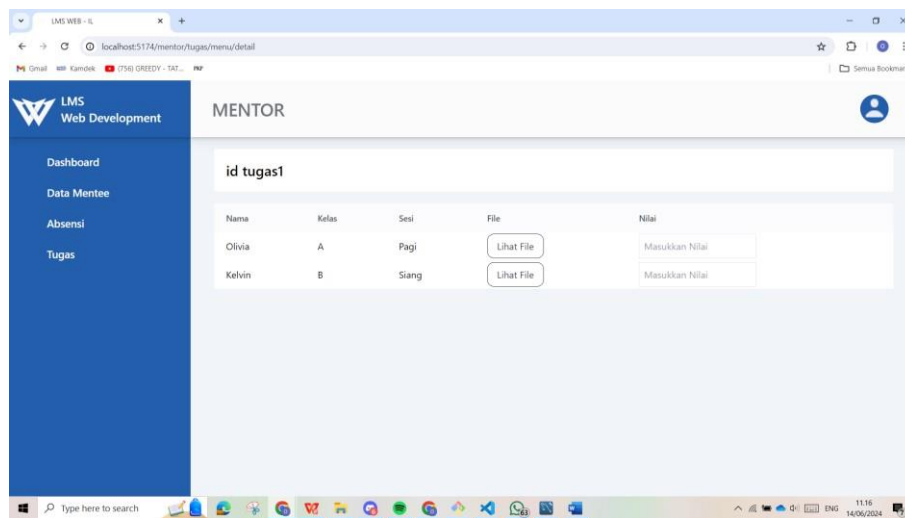
Gambar 21. Halaman Edit Tugas

13. Halaman Pengumpulan Tugas

Halaman pengumpulan tugas adalah halaman yang memungkinkan mentor untuk melihat dan mengelola pengumpulan tugas yang diajukan oleh *mentee*. Halaman ini memberikan mentor akses untuk memantau pengumpulan tugas serta mengelola proses penilaian.



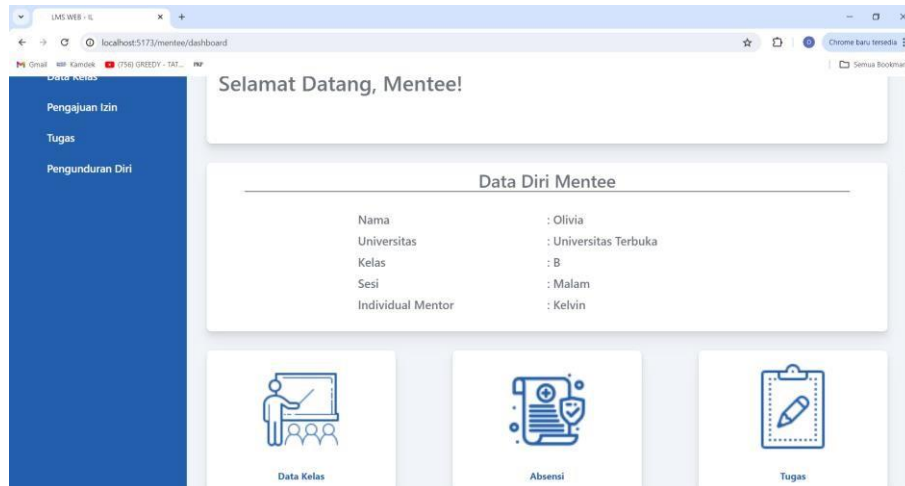
Gambar 22. Halaman Pengumpulan Tugas



Gambar 23. Halaman Pengumpulan Tugas

14. Halaman Dashboard *Mentee*

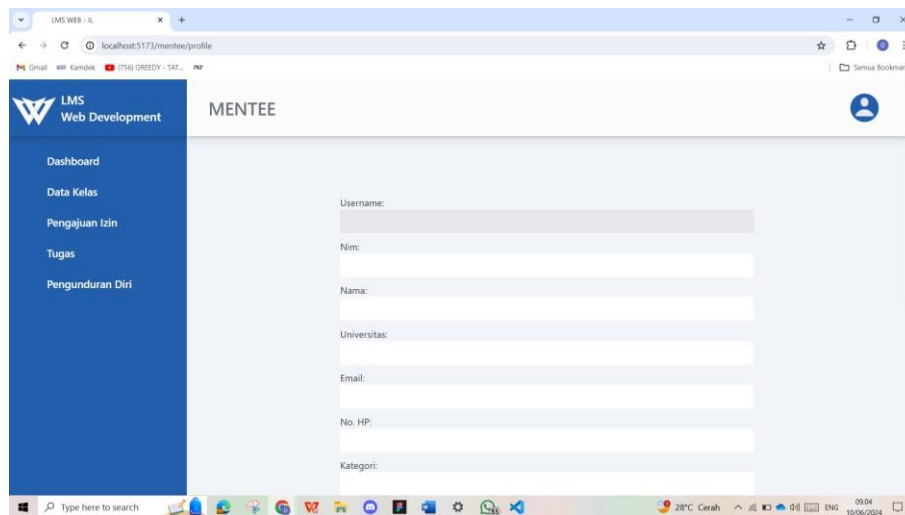
Halaman dashboard *mentee* adalah halaman utama yang disajikan kepada *mentee* ketika *mentee* masuk ke dalam sistem. Fitur ini dirancang untuk memberikan *mentee* akses cepat dan terorganisir ke informasi penting, serta memungkinkan mereka untuk melacak dan mengelola aktivitas mereka dalam proses pembelajaran.



Gambar 24. Halaman Dashboard *Mentee*

15. Halaman Ubah Profil *Mentee*

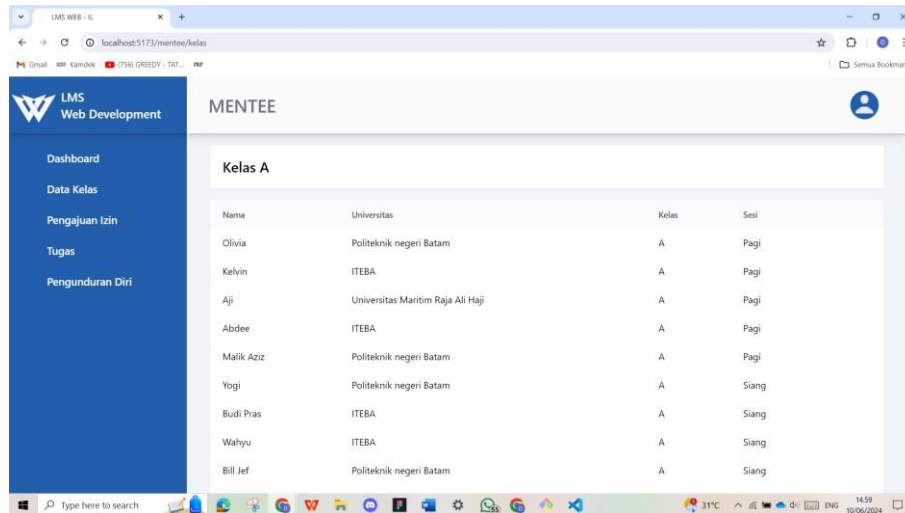
Halaman ubah profil *mentee* adalah bagian penting dari pengalaman pengguna dalam sistem, untuk memungkinkan *mentee* untuk memperbarui informasi mereka sesuai kebutuhan.



Gambar 25. Halaman Profil *Mentee*

16. Halaman Data Kelas

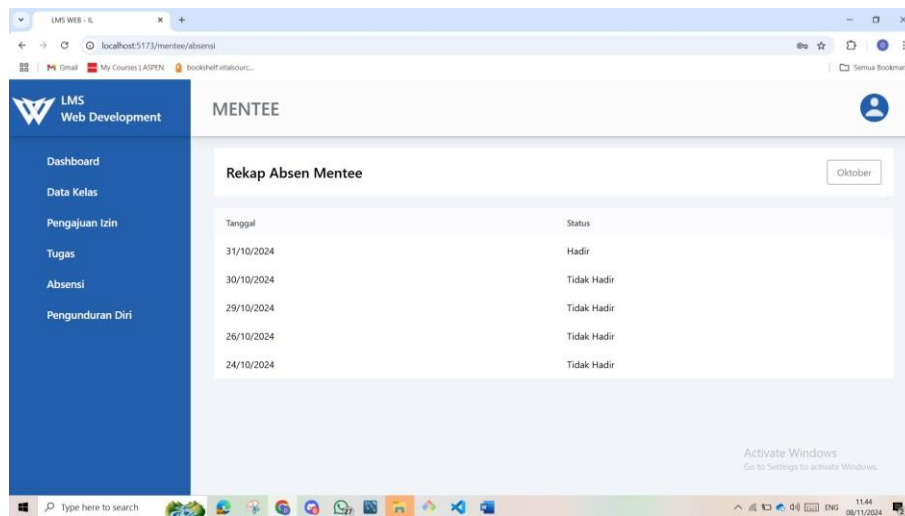
Halaman data kelas adalah fitur yang memungkinkan *mentee* untuk melihat informasi anggota kelas yang diikuti oleh *mentee*. Fitur ini memberikan akses kepada *mentee* untuk memantau rekan-rekan kelasnya.



Gambar 26. Halaman Data Kelas

17. Halaman Rekapitulasi Absensi *Mentee*

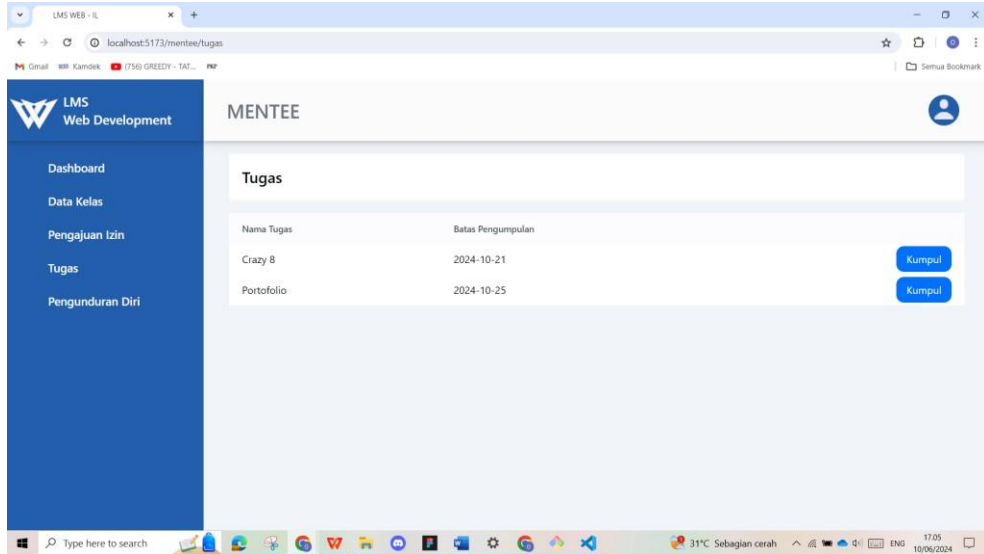
Halaman absensi mentee adalah fitur yang memungkinkan *mentee* untuk melihat rekapitulasi kehadiran *mentee* dalam sesi pembelajaran yang sudah dijadwalkan. Fitur ini memberikan akses kepada *mentee* untuk memantau dan memeriksa absensi mereka.



Gambar 27. Halaman Rekapitulasi Absensi

18. Halaman Tugas *Mentee*

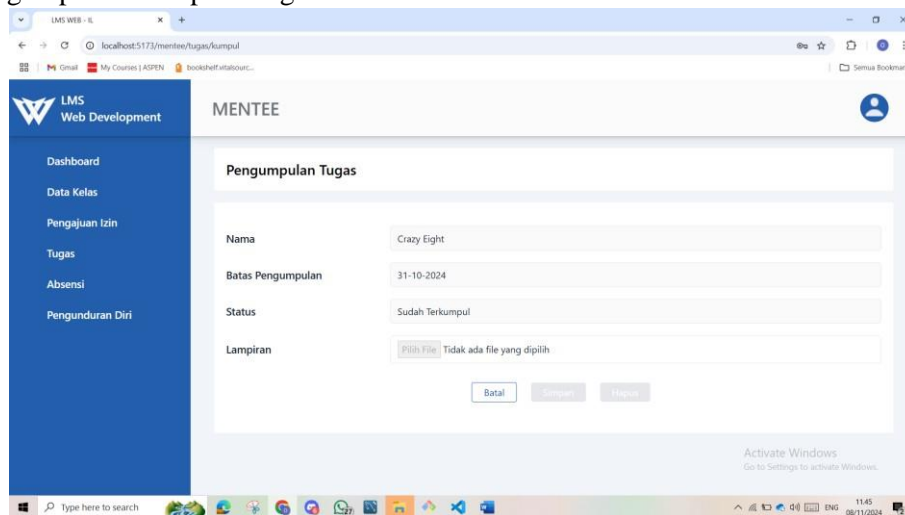
Halaman tugas pada *mentee* adalah fitur yang memungkinkan *mentee* untuk melihat daftar tugas yang diberikan oleh mentor, serta mengumpulkan tugas yang telah selesai dikerjakan. Fitur ini memberikan akses kepada *mentee* untuk melihat batas pengumpulan tugas dan mengumpulkan tugas.



Gambar 28. Halaman Tugas *Menteer*

19. Halaman Pengumpulan Tugas *Menteer*

Halaman pengumpulan tugas adalah halaman yang memungkinkan *mentee* untuk mengumpulkan lampiran tugas.



Gambar 29. Halaman Pengumpulan Tugas *Menteer*

3.7 Hasil Pengujian

Tujuan dari pengujian sistem adalah untuk mengetahui sistem yang dibangun dapat berjalan baik sesuai dengan kebutuhan fungsionalitasnya. Pengujian sistem ini menggunakan metode *black box* yaitu teknik pengujian perangkat lunak untuk menentukan fungsionalitas aplikasi berdasarkan *input* yang tersedia dan keluaran yang diharapkan tanpa mengetahui cara kerja *internal*. Pengujian sistem untuk hak *user* admin telah dilakukan oleh tim internal dari Infinite Learning yaitu Aulia, hak *user* mentor telah dilakukan oleh mentor dari Infinite Learning yaitu Kelvin dan Bill, serta salah satu *mentee* studi independen *batch 7* pada hari Senin, 4 Oktober 2024 dan 14 Desember 2024. Hasil pengujian ini diperlihatkan pada Tabel 3.

Tabel 3. Pengujian

No	Fungsional	Skenario Pengujian	Role	Kondisi Akhir	Hasil
1	<i>Login</i>	Pengguna memasukkan <i>email</i> dan <i>password</i> benar	1. Admin 2. Mentor 3. <i>Mentee</i>	<i>Login</i> Berhasil	Sesuai
		Pengguna memasukkan <i>email</i> dan <i>password</i> salah	1. Admin 2. Mentor 3. <i>Mentee</i>	<i>Login</i> gagal	Sesuai
2	<i>Dashboard</i>		1. Admin 2. Mentor 3. <i>Mentee</i>	Menampilkan halaman <i>dashboard</i> yang berisi data <i>mentee</i> , data <i>user</i> , data universitas, dan validasi pengajuan pengunduran diri	Sesuai
3	Edit Profil	Pengguna memasukkan nama, <i>password</i> lama benar, <i>password</i> baru dan konfirmasi <i>password</i> baru benar	1. Mentor 2. <i>Mentee</i>	Profil berhasil di edit	Sesuai
		Pengguna memasukkan nama, <i>password</i> lama salah, <i>password</i> baru dan konfirmasi <i>password</i> baru benar	1. Mentor 2. <i>Mentee</i>	Profil gagal di edit	Sesuai
4	Mengelola data <i>mentee</i>	1. Tambah <i>mentee</i> dengan mengisi nama, universitas, kelas, sesi dan <i>personal</i> mentor 2. Edit data <i>mentee</i> dengan memasukkan kelas dan sesi yang ingin diubah 3. Menghapus <i>mentee</i> dengan cara klik ikon hapus pada aksi	Admin	1. Data yang dimasukkan berhasil disimpan 2. Data yang diedit telah berubah 3. Data yang dipilih untuk dihapus telah terhapus pada halaman utama	Sesuai
5	Mengelola data Mentor	1. Tambah mentor dengan mengisi nama, <i>email</i> dan <i>username</i> 2. Edit data <i>user</i> dengan memasukkan nama baru 3. Hapus <i>user</i> dengan cara klik pilihan hapus pada aksi	Admin	1. Data yang dimasukkan berhasil disimpan 2. Data yang di edit telah berubah 3. Data yang dipilih untuk dihapus telah terhapus pada halaman utama	Sesuai
6	Mengelola data universitas	1. Tambah univ dengan mengisi nama, alamat, <i>email</i> , nama PIC, <i>email</i> PIC dan nomor hp PIC	Admin	1. Data yang dimasukkan berhasil disimpan 2. Data yang di edit telah berubah	Sesuai

		2. Edit data universitas dengan memasukkan nama PIC, <i>email</i> PIC, dan nomor hp PIC 3. Hapus data universitas dengan memilih aksi hapus		3. Data yang telah dipilih untuk dihapus telah terhapus pada halaman utama	
7	Validasi pengajuan pengunduran diri <i>mentee</i>	Admin memilih menu validasi untuk melihat pengajuan, kemudian melihat data pengajuan pengunduran diri, lalu memvalidasi pengajuan dengan memilih aksi terima	Admin	Ajuan pengunduran diri berhasil di validasi	Sesuai
8	Melihat data <i>mentee</i>	Mentor memilih menu data <i>mentee</i> untuk melihat data <i>mentee</i> baik secara keseluruhan, per kelas, ataupun <i>mentee individual</i> .	Mentor	Sistem akan menampilkan informasi <i>mentee</i> secara keseluruhan, per kelas ataupun <i>mentee individual</i>	Sesuai
9	Melihat data nilai <i>mentee</i>	Mentor memilih aksi detail pada nama <i>mentee</i> untuk melihat data nilai <i>mentee</i> .	Mentor	Sistem akan menampilkan informasi nilai setiap <i>mentee</i> yang dipilih	Sesuai
10	Melakukan absensi <i>mentee</i>	Mentor memilih absensi dan melakukan absensi <i>mentee</i> secara per kelas atau keseluruhan	Mentor	Absen <i>mentee</i> berhasil dilakukan	Sesuai
11	Melihat form absensi	Mentor membuka form absensi untuk melihat data yang izin dan pindah sesi	Mentor	Sistem menampilkan <i>mentee-mentee</i> yang izin setiap harinya	Sesuai
12	Menambahkan Tugas	Mentor memilih menu tugas dan menambahkan tugas dengan mengisi nama tugas dan batas pengumpulan	Mentor	Tugas berhasil ditambahkan	Sesuai
13	Mengedit tugas	Mentor memilih aksi edit pada tugas yang ingin diedit lalu mengisi nama atau batas pengumpulan tugas	Mentor	Tugas berhasil diedit	Sesuai
14	Menghapus tugas	Mentor memilih aksi hapus pada tugas yang ingin dihapus	Mentor	Tugas berhasil dihapus	Sesuai
15	Melihat Pengumpulan Tugas	Mentor memilih tugas yang mau dilihat pengumpulannya	Mentor	Sistem menampilkan daftar pengumpulan tugas oleh <i>mentee</i>	Sesuai

16	Memberikan nilai tugas	Mentor memberikan nilai untuk tugas <i>mentee</i> yang telah dikumpulkan	Mentor	Nilai berhasil ditambahkan	Sesuai
17	Mengedit nilai tugas	Mentor mengedit nilai tugas <i>mentee</i> dengan menginput nilai baru pada kolom nilai	Mentor	Nilai berhasil diperbarui	Sesuai
18	Melihat data kelas	<i>Mentee</i> memilih menu data kelas untuk melihat anggota kelasnya	<i>Mentee</i>	Sistem menampilkan anggota kelas sesuai kelas <i>mentee</i>	Sesuai
19	Melihat rekapitulasi absensi	<i>Mentee</i> memilih menu absensi untuk melihat rekapitulasi absen pribadi selama pembelajaran	<i>Mentee</i>	Sistem menampilkan rekapitulasi absen pribadi <i>mentee</i>	Sesuai
20	Melihat tugas	<i>Mentee</i> memilih menu tugas untuk melihat daftar tugas dari yang terbaru dan batas pengumpulannya	<i>Mentee</i>	Sistem menampilkan daftar tugas dari yang terbaru dan batas pengumpulannya	Sesuai
21	Mengumpulkan tugas	<i>Mentee</i> mengklik tombol aksi kumpul pada tugas yang ingin dikumpulkan lalu mengunggah <i>file</i> yang ingin dikumpulkan kemudian mengklik tombol kumpul	<i>Mentee</i>	Lampiran tugas berhasil dikumpulkan	Sesuai
22	Mengedit lampiran pengumpulan tugas	<i>Mentee</i> membuka kembali tugas yang sudah dikumpulkan, lalu mengunggah kembali file terbaru kemudian menekan tombol edit untuk memperbarui file pengumpulan	<i>Mentee</i>	Lampiran tugas berhasil diperbarui	Sesuai
23	Menghapus lampiran pengumpulan tugas	<i>Mentee</i> membuka kembali tugas, kemudian menekan tombol hapus	<i>Mentee</i>	Lampiran tugas berhasil dihapus	Sesuai
24	Mengajukan Izin	<i>Mentee</i> membuka pengajuan izin lalu mengisi <i>form</i> izin untuk melakukan perizinan	<i>Mentee</i>	Izin berhasil diajukan	Sesuai
25	<i>Logout</i>	Menekan tombol keluar	1. Admin 2. Mentor 3. <i>Mentee</i>	<i>User</i> berhasil keluar dan kembali ke halaman masuk	Sesuai

4. KESIMPULAN DAN SARAN

Dari hasil pembahasan mengenai Sistem Pengelolaan Data Program Studi Independen berbasis Web di Infinite Learning dapat diambil kesimpulan sebagai berikut:

1. Website Sistem Manajemen Data Studi Independen Program Web Development di Infinite Learning berhasil dikembangkan menggunakan bahasa pemrograman Javascript dengan framework ReactJS dan Express, serta MySQL sebagai *database*.
2. Seluruh fitur yang direncanakan telah berhasil diimplementasikan sesuai dengan perancangan yang telah dibuat sebelumnya.
3. Berdasarkan pengujian dengan metode *BlackBox*, sistem ini sudah sesuai dan dapat diterima dengan baik. Meskipun demikian, sistem ini belum sepenuhnya sempurna, sehingga memerlukan pengembangan lebih lanjut untuk menambahkan fitur-fitur yang masih belum tersedia.

Namun mengingat keamanan data sangat penting karena sistem ini menyimpan informasi penting terkait dengan data pribadi mentee, tugas, dan nilai-nilai akademik. Disarankan untuk menambahkan lapisan keamanan yang lebih kuat seperti enkripsi data, autentikasi dua faktor, serta proteksi terhadap akses tidak sah. Dengan cara ini, data yang disimpan dalam sistem dapat lebih terjaga dan mencegah potensi kebocoran data.

DAFTAR PUSTAKA

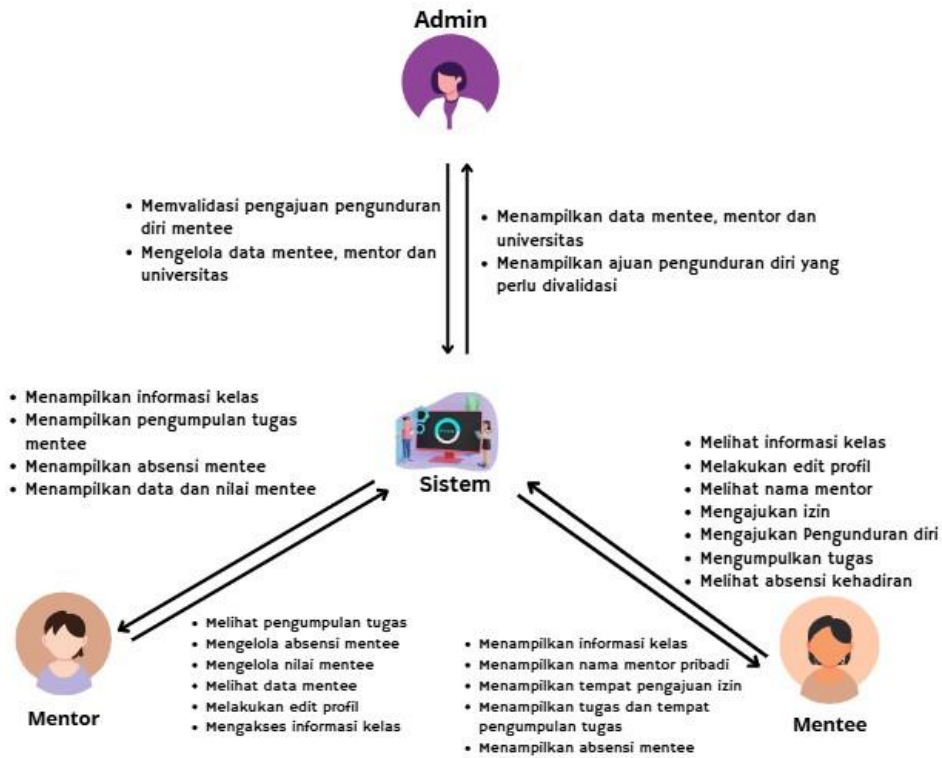
- [1] M. A. Siswanto, B. A. Nafsah, H. Susanto, and S. Juani, "Analisa dan Perancangan Sistem Informasi Sekolah SMK Bina Putra Mandiri Berbasis Website Menggunakan Framework Laravel," vol. 1, no. 1, pp. 107–112, 2023.
- [2] R. D. Rusdian Yusron and M. M. Huda, "Analisis Perancangan Sistem Informasi Perpustakaan Menggunakan Model Waterfall Dalam Peningkatan Inovasi Teknologi," *J. Autom. Comput. Inf. Syst.*, vol. 1, no. 1, pp. 26–36, 2021, doi: 10.47134/jacis.v1i1.4.
- [3] S. A. Rosadi, A. Wirawan, and S. F. Sujadi, "Pembangunan Aplikasi Pengelolaan Data pada Rumah Sakit Berbasis Web," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 2, pp. 416–434, 2021, doi: 10.28932/jutisi.v7i2.3696.
- [4] M. K. Kodmelwar, P. R. Futane, S. D. Pawar, S. A. Lokhande, and S. P. Dhanure, "A Comparative Study of Software Development Waterfall, Spiral and Agile Methodology," *J. Posit. Sch. Psychol.*, vol. 6, no. 3, pp. 7013–7017, 2022, [Online]. Available: <https://www.journalppw.com/index.php/jpsp/article/view/4267>
- [5] A. Wahid Abdul, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, pp. 1–5, 2020.
- [6] F. N. Hasanah, *Buku Ajar Rekayasa Perangkat Lunak*. 2020. doi: 10.21070/2020/978623-6833-89-6.
- [7] B. Acharya and K. Sahu, "Software Development Life Cycle Models: A Review Paper," *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 12, pp. 169–176, 2020, doi: 10.34218/IJARET.11.12.2020.019.
- [8] J. A. Ramadhan, D. T. Haniva, and A. Suharso, "Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid," *JIEET J. Inf. Eng. Educ. Technol.*, vol. 07, no. 01, pp. 36–42, 2023, [Online]. Available: <https://journal.unesa.ac.id/index.php/jieet/article/view/21941>
- [9] N. Dwivedi, D. Katiyar, and G. Goel, "A Comparative Study of Various Software Development Life Cycle (SDLC) Models," *Int. J. Res. Eng. Sci. Manag.*, vol. 5, no. 3, pp. 141–145, 2022, [Online]. Available: <https://www.ijresm.com>
- [10] R. Malan, D. Bredemeyer, and B. Consulting, "Functional requirements and use cases," *functreq. pdf*, 39k June, pp. 1–8, 1999, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.436.4773&rep=rep1&type=pdf>
<http://www.qxzyzz.cn/dzs/reslib/400/180/110/020/030/050/060/1000000000240572.pdf>

- [11] H. Koç, A. M. Erdoğan, Y. Barjakly, and S. Peker, “UML Diagrams in Software Engineering Research: A Systematic Literature Review,” p. 13, 2021, doi: 10.3390/proceedings2021074013.
- [12] Ravi, S. Yadav, R. Jindal, and S. Anand, “A Study on Comparison of UML and ER Diagram,” *Int. Res. J. Eng. Technol.*, vol. 7, no. 5, pp. 7575–7580, 2020, [Online]. Available: <https://irjet.com/archives/V7/i5/IRJET-V7I51425.pdf>
- [13] M. H. L. Vo and Q. Hoang, “Transformation of uml class diagram into owl ontology,” *J. Inf. Telecommun.*, vol. 4, no. 1, pp. 1–16, 2020, doi: 10.1080/24751839.2019.1686681.

LAMPIRAN

1 Gambaran umum sistem

Sistem Pengelolaan Data Program Studi Independen Berbasis Web dirancang untuk memudahkan pengelolaan data program studi independen. Sistem ini juga dirancang untuk dapat memudahkan admin, mentor dan *mentee* dalam pelaksanaan studi independen. Gambaran umum sistem pengelolaan data program studi independen dapat dilihat pada Gambar 2.



Gambar 30. Gambaran umum sistem