

**Rancang Bangun Repositori Konten Digital
Berbasis Web Sebagai *Platform Review* di PT
Labtech Penta International**

PROYEK AKHIR

Disusun Oleh:

Wildy Johannes Simanjuntak

3312011076

**Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Teknik Informatika**



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
2026**

HALAMAN PENGESAHAN

Rancang Bangun Repositori Konten Digital Berbasis Web Sebagai *Platform Review* di PT Labtech Penta International

Oleh :

Wildy Johanes Simanjuntak 3312011076

Telah diuji dan dipertahankan di depan Tim Penguji

Dalam sidang akhir

Pada tanggal 6 July 2026

Dan dinyatakan **LULUS**

Batam, 3 August 2026

Disetujui dan disahkan oleh:

Pembimbing,



TTE oleh:

Maidel Fani, S.Pd., M.Kom.
3rd August 2026

Perihal oleh:

Persetujuan untuk halaman pengesahan , tertanggal
3 August 2026

Untuk:

WILDY JOHANES SIMANJUNTAK
3312011076

Pindai QRCode untuk Verifikasi.

Pastikan URL di if.polibatam.ac.id/tugas-akhir

Maidel Fani, S.Pd., M.Kom

NIK. 117192

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga laporan Proyek Akhir yang berjudul “Pengembangan Sistem Repositori Aset Digital Berbasis Web” ini dapat diselesaikan dengan baik dan tepat pada waktunya.

Penulisan laporan ini disusun sebagai salah satu syarat untuk menyelesaikan program studi Master dalam bidang Pengembangan Perangkat Lunak. Selama proses penyusunan laporan ini, penulis menyadari bahwa keberhasilan ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang tulus kepada:

- Maidel Fani, S.Pd., M.Kom, Selaku Dosen Pembimbing yang telah memberikan arahan, waktu, dan ilmu yang sangat berharga dalam membimbing penulis menyelesaikan proyek ini.
- Para Dosen Penguji yang telah memberikan saran dan kritik membangun demi kesempurnaan laporan ini.
- Labtech Penta Internasional, yang telah memberikan saya kesempatan, dukungan fasilitas, serta data yang diperlukan selama proses penelitian dan pengembangan sistem ini berlangsung.
- Keluarga tercinta, terutama ibu saya, yang selalu memberikan dukungan moral, doa, dan motivasi tanpa batas terhadap penulis dalam menempuh pendidikan.
- Rekan-Rekan/Teman, yang telah membantu saya memberikan semangat dan diskusi yang bermanfaat dalam penyelesaian laporan ini.

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI.....	iv
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
ABSTRAK	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang	2
1.2. Rumusan Masalah	3
1.3. Tujuan	3
1.4. Batasan Masalah.....	3
1.5. Manfaat	4
1.6. Kolaborasi dan Sinergi dalam Pelaksanaan Proyek	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Penelitian Terkait	5
2.2. Landasan Teori.....	7
2.3. Metode Pengembangan Produk.....	9
BAB III ANALISIS DAN PERANCANGAN	12
3.1. Analisis Kebutuhan	12
3.1.1 Proses Bisnis Berjalan.....	12
3.1.2 Gambaran Umum Sistem	12
3.2. Perancangan	13
3.2.3 Kebutuhan Fungsional	13
3.2.4 Kebutuhan Non-Fungsional	14
3.2.5 <i>Use Case Diagram</i>	15
3.2.6 Skenario <i>Use Case</i>	15
3.2.7 Sequence diagram	38
3.2.8 ER Diagram.....	46
3.2.9 Perancangan Antarmuka (<i>Wireframe</i>).....	47
BAB IV IMPLEMENTASI DAN PENGUJIAN	59
4.1. Hasil Implementasi.....	59

4.1.1 Deskripsi Proses Implementasi	59
4.1.2 Tampilan Antarmuka Sistem	60
4.2. Pengujian.....	64
BAB V KESIMPULAN.....	68
5.1 Kesimpulan	68
5.2 Saran.....	68
DAFTAR PUSTAKA	69
DAFTAR LAMPIRAN	70

DAFTAR TABEL

Tabel 1. Perbandingan Penelitian Sebelumnya.....	5
Tabel 2. Skenario Use Case Diagram - Login.....	16
Tabel 3. Skenario Use Case Diagram – User Management	17
Tabel 4. Skenario Use Case Diagram – Category Management.....	22
Tabel 5. Skenario Use Case – Module Management, Sign Task and Upload Landing page	25
Tabel 6. Skenario Use Case Diagram - Logout.....	29
Tabel 7. Skenario Use Case Diagram - Update Profile.....	30
Tabel 8. Skenario Use Case Diagram - Comment	31
Tabel 9. Skenario Use Case - View Projects.....	32
Tabel 10. Skenario Use Case Diagram – View Log Activity	34
Tabel 11. Skenario Use Case Diagram - Change Password.....	35
Tabel 12. Skenario Use Case Diagram - Upload Content.....	37
Tabel 13. Pengujian Sistem.....	65

DAFTAR GAMBAR

Gambar 1. Waterfall.....	10
Gambar 2. Gambaran Umum Sistem	13
Gambar 3. Use Case Diagram.....	15
Gambar 4. Sequence Diagram Authentication, Profile, Change Password and Monitoring	40
Gambar 5. Sequence Diagram Category Management.....	42
Gambar 6. Sequence Diagram Module Management	44
Gambar 7. Sequence Diagram Upload Content	45
Gambar 8. Sequence Diagram Users Management.....	46
Gambar 9. ER Diagram.....	47
Gambar 10. Wireframe Login.....	48
Gambar 11. Wireframe User Management - List.....	48
Gambar 12. Wireframe User Management - Create	49
Gambar 13. Wireframe User Management - Delete	49
Gambar 14. Wireframe User Management - Update	49
Gambar 15. Wireframe Category Management - List	50
Gambar 16. Wireframe Category Management – Create.....	50
Gambar 17. Wireframe Category Management - Delete	51
Gambar 18. Wireframe Category Management - Update	51
Gambar 19. Wireframe Module Management - List.....	52
Gambar 20. Wireframe Module Management - Create	52
Gambar 21. Wireframe Module Management - Remove.....	53
Gambar 22. Wireframe Module Management - Update	53
Gambar 23. Wireframe Profile - View.....	54
Gambar 24. Wireframe profile - Edit Photo.....	54
Gambar 25. Wireframe Profile - Edit Bio.....	54
Gambar 26. Wireframe Dashboard	55
Gambar 27. Wireframe View Projects - Modules List	55
Gambar 28. Wireframe View Projects – Preview & Comment.....	56
Gambar 29. Wireframe Log Activity	57
Gambar 30. Wireframe Change Password	57
Gambar 31. Wireframe Upload Content	58
Gambar 32. Antarmuka - Login	60
Gambar 33. Antarmuka - CRUD Users	61
Gambar 34. Antarmuka - CRUD Category.....	61
Gambar 35. Antarmuka - CRUD Module.....	62
Gambar 36. Antarmuka - Update Profile	62
Gambar 37. Antarmuka - View Projects & Comment.....	63
Gambar 38. Antarmuka - Log Activity	63
Gambar 39. Antarmuka - Change Password.....	64
Gambar 40. Antarmuka - Upload Content	64

DAFTAR LAMPIRAN

Lampiran A. Link Produk	70
Lampiran B. Link List Of Digital Content.....	70

ABSTRAK

PT Labtech Penta International menghadapi tantangan dalam kolaborasi dan peninjauan konten digital secara *real-time* karena keterbatasan penyimpanan data lokal. Penelitian ini bertujuan untuk merancang dan membangun aplikasi repositori konten digital berbasis web yang berfungsi sebagai platform peninjauan. Metode pengembangan yang digunakan adalah metode waterfall. Sistem ini dikembangkan menggunakan *framework Nuxt.js* sebagai klien, *Node.js* untuk server, dan *MongoDB* sebagai basis data. Implementasi aplikasi ini telah terbukti mampu menyederhanakan manajemen aset digital dan mempercepat koordinasi antar pihak dalam melakukan peninjauan konten *online*.

Kata Kunci: repositori, konten digital, review

BAB I PENDAHULUAN

Perkembangan teknologi informasi berkembang cepat yang membuat setiap perusahaan berlomba-lomba dalam mengembangkan produk mereka menjadi produk yang lebih modern. Perkembangan teknologi tentunya memiliki sisi yang baik dan buruk dalam kehidupan manusia, yang mana membuat pekerjaan menjadi lebih mudah atau menjadi ketergantungan.

Perkembangan teknologi tidak hanya berefek perusahaan tertentu saja tapi ke perusahaan yang di bidang sektor lainnya seperti rumah sakit atau bahkan perusahaan yang bergerak dibidang pendidikan seperti PT Labtech Penta International, perusahaan ini adalah perusahaan yang bergerak dibidang pendidikan yang telah berdiri sejak tahun 1990 di sekupang, Batam.

Produk yang dihasilkan oleh perusahaan ini adalah konten digital berbasis *WebGL*, *Virtual Reality*, *Augmented Reality* serta *Learning Management System* dan juga alat peraga pendidikan. Aset digital seperti konten digital selalu mengalami pembaruan, revisi serta melibatkan kalaborasi antar berbagai pihak selama proses pengembangan nya. Pengelolaan yang tidak terorganisir dengan baik akan menyebabkan kesulitan dalam penyimpanan dan pencarian hingga proses evaluasi dan pemberian umpan balik (*feedback*).

Seiring meningkatnya jumlah aset digital yang telah dikelola, penggunaan media penyimpanan lokal menjadi kurang efektif dan memadai. Kondisi tersebut dapat membatasi akses kepada pihak terkait lainnya (*Subject Matter Expert*) dalam meninjau (*review*) perkembangan sebuah konten digital. Berbagai penelitian menunjukan bahwa digitalisasi pengelolaan aset melalui sistem repositori berbasis web dapat meningkatkan efisiensi penyimpanan, mempermudah dalam mengelola aset digital, serta dapat juga dalam melakukan kolaborasi antar pihak. [1]-[3].

Salah satu implementasi digitalisasi tersebut adalah pengembangan repositori konten digital berbasis web yang tidak hanya berfungsi sebagai penyimpanan tapi juga sebagai sarana kolaborasi melalui fitur peninjau dan pemberian umpan balik (*feedback*). Dengan adanya sistem tersebut, proses pengembangan produk menjadi lebih mudah, terorganisir dan efektif.[4]

Berdasarkan kebutuhan tersebut, penelitian ini mengembangkan repositori konten digital berbasis web menggunakan *framework Nuxt JS* sebagai *client* dan juga menggunakan *Node JS* sebagai *server*. Kombinasi dari kedua *framework* tersebut membantu dalam mengembangkan sistem repositori menjadi lebih terstruktur dan membuat performa sistem menjadi lebih stabil.

1.1. Latar Belakang

PT Labtech Penta International adalah usaha patungan antara ECI Equipment LTD di Amerika Serikat dan PT Andes Reko Nusantara Indonesia yang didirikan di Jakarta. PT Labtech Penta International mulai beroperasi pada tahun 1990, di kawasan industri Sekupang, plot 34 Batam, dan telah beroperasi hingga sekarang. Lokasinya di kawasan perdagangan bebas Batam dekat dengan Singapura dan dunia. Labtech memiliki kantor pemasaran di kota-kota besar seperti Singapura, Malaysia, Jakarta, dan Yordania untuk membantu perusahaan dalam proses komunikasi dan pemasaran.

PT Labtech Penta International adalah salah satu pemain besar dalam desain dan produksi alat peraga untuk sistem pendidikan keterampilan dan teknis, termasuk konten 3D tentang cara kerja mesin mobil, juga menjadi salah satunya. Labtech dikembangkan dalam kolaborasi dengan pengalaman berdasarkan bantuan dari pendidik profesional terlatih Amerika, Eropa, dan Asia. "Membuat Teknologi Terlihat" adalah salah satu filosofi perusahaan dalam menciptakan sistem pendidikan yang dapat diakses oleh orang lain.

Bekerja sama dengan perusahaan besar lainnya, Labtech, yang merupakan perusahaan yang telah hadir selama 32 tahun, bermitra dalam peluncuran produk bersama Microsoft Certified Partner, Worlddidac, ISTE, Intel, MSC, CompTia, IVETA, GenYes, Zspace, dan lainnya. Selain itu, Labtech telah melakukan proyek pendidikan berbasis permainan pembelajaran virtual di negara-negara seperti Peru, Australia, Afrika, dan Malaysia. Pembelajaran Virtual adalah departemen perangkat lunak yang mengembangkan modul pembelajaran permainan dari kerangka kerja *Unity*. Ada tim dalam pemodelan dan kemudian tim pemrograman dalam membangun modul dalam pembelajaran virtual.

Berdasarkan itu, pemrogram masih menyimpan data konten di server lokal mereka (artinya pemrogram hanya dapat mengerjakan modul di dalam perusahaan), sehingga sulit bagi pihak eksternal untuk dapat memantau pengembangan konten secara *real-time*. Sebelumnya, pihak eksternal tidak memiliki akses *real-time* untuk meninjau konten digital secara *real-time* karena pengembang perangkat lunak hanya dapat meninjaunya di antara pertemuan dan pertemuan Zoom atau lainnya untuk menunjukkan pengembangan konten. Ini juga mengakibatkan pemrogram menawarkan umpan balik melalui email dan *WhatsApp* saja.

Saat ini, Pembelajaran Virtual terdiri dari 137 modul atau sekitar 1097 potongan konten, yang sulit untuk dikelola oleh Admin (dan harus melakukan banyak unggahan untuk mempertahankan pembaruan ketika pemrogram menambahkan konten baru). Oleh karena itu, dalam penelitian ini, aplikasi repositori dikembangkan untuk memfasilitasi pertukaran, kerjasama, dan pertukaran di antara para pemrogram serta memberikan umpan balik. Untuk memastikan pekerjaan, metode *waterfall* diadopsi untuk mengembangkan aplikasi ini.

1.2. Rumusan Masalah

Dari latar belakang yang telah diuraikan diperoleh rumusan masalah yaitu Bagaimana membangun repositori konten digital sebagai *platform review*.

1.3. Tujuan

Tujuan dari penelitian ini adalah merancang bangun repositori konten digital sebagai platform review.

1.4. Batasan Masalah

Untuk memfokuskan tujuan, maka batasan masalah pada penelitian ini adalah sebagai berikut :

1. Aplikasi ini hanya bertujuan untuk sebagai tempat repository dan platform review
2. Aplikasi hanya digunakan oleh pihak perusahaan dan pihak yang berkaitan (SME)

1.5. Manfaat

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Penelitian ini dapat membuat repositori konten digital
2. Hasil dari penelitian ini dapat bermanfaat bagi pihak yang terkait dalam menyimpan data ataupun melakukan *review* konten digital.

1.6. Kolaborasi dan Sinergi dalam Pelaksanaan Proyek

Bentuk kolaborasi dan dukungan yang saya dapatkan dalam pelaksanaan proyek ini adalah sebagai berikut :

1. Konsultasi dengan dosen pembimbing, Ibu Maidel Fani S.Pd., M.Kom, dilakukan secara *luring/offline* setiap satu minggu sekali untuk memastikan laporan ilmiah sesuai dengan standar akademik.
2. Diskusi dengan pembimbing lapangan, Bapak Ferry Sugianto selaku *Manager* di *Virtual Learning Department*, dilakukan secara *luring/offline* yang mana pertemuan tatap muka ini bertujuan untuk membedah permasalahan teknis yang ditemukan, menyelaraskan ekspektasi terhadap fungsional sistem yang diinginkan.
3. Diskusi rutin dengan pembimbing lapangan, Bapak Yossie Dharma Putra selaku *Leader* di *section Software Web Developer*, dilakukan setiap hari kerja untuk memantau progres pengerjaan serta pemberian solusi pada permasalahan yang didapat saat pengerjaan selama proses pengembangan sistem repositori.

Bentuk kolaborasi ini menunjukkan bahwa dalam pengerjaan proyek ini dilakukan secara individu, dan juga terdapat sinergi yang kuat dari pihak yang terkait dengan proyek ini. Keterlibatan dari pihak seperti dosen pembimbing, dosen lapangan dan manajer yang menjadi bagian penting dalam proses pengerjaan proyek ini.

BAB II TINJAUAN PUSTAKA

Bab ini memperkenalkan latar belakang teoritis dan tinjauan literatur yang membentuk dasar ilmiah untuk menyelesaikan masalah manajemen aset di PT Labtech Penta International. Seperti yang dinyatakan dalam pengantar, tantangan paling umum yang kami hadapi adalah rendahnya efisiensi kerja sama akibat sistem penyimpanan lokal dan masalah umpan balik yang tertunda.

Oleh karena itu, bab ini bertujuan untuk meninjau studi penelitian yang tersedia tentang digitalisasi repositori dan dampak umpan balik terhadap kinerja sistem. Selain itu, bagian ini juga membahas dukungan teoritis dari teknologi yang dipilih, yaitu kerangka kerja *Nuxt.js* dan *Node.js*, yang stabilitas dan kinerjanya telah diuji secara komparatif terhadap teknologi lain seperti Bun. Wawasan teoritis ini disusun sepanjang proses desain sistem, dan kami berencana untuk menerapkan metode *waterfall* untuk menciptakan platform tinjauan konten yang kolaboratif dan andal.

2.1. Penelitian Terkait

Penelitian terkait dilakukan sebagai dasar untuk menempatkan proyek akhir ini dalam studi relevan sebelumnya. Tujuan dari tinjauan literatur ini adalah untuk membahas tiga aspek: manajemen aset digital yang transparan, efisiensi sistem repositori berbasis web, dan analisis kinerja teknologi modern (*runtime JavaScript* dan kerangka kerja terbaru). Dengan merujuk pada studi-studi ini, diharapkan sistem yang dibangun di PT Labtech Penta International akan memiliki standar kinerja yang optimal dan fitur kolaborasi yang efektif. Ringkasan dari studi-studi sebelumnya ini disusun dalam Tabel 1.

Tabel 1. Perbandingan Penelitian Sebelumnya

Peneliti/Tahun	Judul/Topik Penelitian	Metode/Teknologi	Kelebihan	Kekurangan/Celah
Musoffa, dkk. (2022)	Sistem Informasi Manajemen Aset Berbasis web di	Pengumpulan data Kualitatif, Metode Pengembangan Perangkat lunak (Spiral)	Mampu mendigitalisasi aset yang sebelumnya manual	Belum dijelaskan secara mendalam mengenai

	Universitas Teknologi Sumbawa		(Excel) menjadi lebih efektif, efisien, dan terpusat.	skalabilitas sistem jika jumlah aset melonjak drastis.
F.C Wiryadi (2023)	Perbedaan Umpan Balik Langsung dan Tertunda pada Keterampilan Resusitasi	Quasi Eksperimental (Pre & Post-test Control Study)	Memberikan bukti ilmiah bahwa umpan balik langsung meningkatkan keterampilan lebih tinggi (33,3%) dibanding tertunda.	Fokus pada bidang kesehatan (kebidanan), sehingga implementasi teknis umpan balik di sistem IT perlu penyesuaian.
T.A.Riza , dkk (2026)	Digitalisasi Manajemen Aset Desa Citeureup	Pengembangan Sistem Informasi Berbasis Website	Meningkatkan transparansi dan akurasi data aset desa yang sebelumnya dikelola secara manual.	Penggunaan Microsoft Excel sebagai pembanding dianggap masih memiliki keterbatasan aksesibilitas bagi masyarakat luas.
S.Syamsul, dkk (2025)	Pengembangan Sistem Repositori Jurnal Berbasis Web	Metode Agile, Framework YII, System Usability Scale (SUS)	Hasil pengujian SUS mencapai 83,3 (Kategori <i>Acceptable</i>), memudahkan pelacakan karya ilmiah mahasiswa	Masih berfokus pada pengelolaan draf jurnal, belum mencakup kolaborasi review konten digital yang lebih kompleks.
A.J.Zaidan, dkk (2025)	Analisis Performa Rendering Framework Astro, Next, Nuxt, dan SvelteKit	Extreme Programming (XP), Google Lighthouse, JMeter	Memberikan data komprehensif bahwa Nuxt JS unggul dalam stabilitas sistem (<i>response time</i> dan <i>throughput</i>).	Astro JS memang unggul di <i>Web Vitals</i> , namun kurang stabil pada pengujian beban tinggi dibanding Nuxt JS.

M.F. Ahmod, dkk (2023)	JavaScript Runtime Performance Analysis: Node and Bun	Performance Analysis (Memory, Response Time, Throughput)	Membuktikan Bun secara signifikan lebih cepat dalam eksekusi dan respons waktu dibanding Node.js.	Analisis masih bersifat satu sisi; aspek keamanan, kompatibilitas, dan reliabilitas Bun belum diuji mendalam.
Mutadin & Yulianti	Perancangan Sistem Informasi Pembuatan Akta Kelahiran Berbasis Web	Metode Waterfall	Mempercepat pelayanan publik dan pengelolaan data informasi di tingkat kecamatan secara terstruktur.	Metode Waterfall memiliki sifat kaku, sehingga sulit untuk mengakomodasi perubahan kebutuhan di tengah pengembangan.

2.2. Landasan Teori

Bagian ini akan memperkenalkan dasar-dasar pengembangan sistem repositori konten digital yang berfokus pada teori-teori yang secara langsung berkaitan dengan efisiensi kolaborasi dan teknologi yang diterapkan. Definisi dan konsep utama yang sesuai dengan topik proyek sebagai berikut :

1. Repositori

Repositori adalah sistem yang digunakan untuk mengelola, menyimpan serta menyediakan akses terhadap aset digital secara terstruktur sehingga mempermudah dalam proses pencarian, penyimpanan, dan pemanfaatan kembali informasi. Dalam implementasinya, karakteristik repositori tergantung pada tujuan dan konteks penggunaannya. Pada repositori institusi, repositori umumnya digunakan untuk menyimpan dokumen atau karya ilmiah yang telah dipublikasikan sehingga bersifat final. Sebaliknya, pada lingkungan pengembangan perangkat lunak dan aset digital, repositori berfungsi sebagai media kolaborasi antar pihak pengembang atau pihak terkait lainnya.

Oleh karena itu, repositori pada penelitian ini tidak diposisikan sebagai repositori final, melainkan sebagai repositori kolaboratif atau tidak bersifat final agar dapat mendukung dalam pengembangan sebuah aset digital. Aset yang dikelola adalah konten *WebGL* yang dikembangkan menggunakan *Unity*, sehingga setiap perubahan dapat diunggah, ditinjau, dan diberikan umpan balik (*feedback*) oleh *SME* sebelum menghasilkan versi final.

2. Konsep Umpan Balik (*Feedback*)

Untuk menjembatani komunikasi, sistem ini berisi platform ulasan. Wiryadi (2023) mengatakan: Umpan balik langsung yang diberikan di platform terintegrasi mereka jauh lebih berguna daripada umpan balik yang tertunda di PT Labtech Penta International [2].

3. Konten Digital

Konten digital adalah segala bentuk informasi yang dibuat, disimpan, diproses, serta didistribusikan berbasis digitalisasi sehingga dapat mudah diakses melalui perangkat elektronik. Konten digital tidak terbatas berbasis teks, tapi juga berupa gambar, video, audio, animasi, serta model 3d, aplikasi interaktif, hingga simulasi berbasis web.

Pada penelitian ini, konten digital yang dikelola berupa aplikasi *WebGL* yang dikembangkan menggunakan *Unity*. *WebGL* adalah proses hasil build aplikasi *Unity* sehingga dapat dijalankan melalui *web browser* tanpa perlu melakukan instalasi tertentu. *WebGL* yang dihasilkan berupa sebuah konten digital berbasis simulasi dan model 3d yang digunakan untuk menyampaikan informasi tertentu.

4. *Framework Nuxt.js* dan *Node.js*

Untuk menjamin stabilitas dan performa aplikasi, digunakan kombinasi teknologi berikut :

a. *Nuxt.js*

Arsitektur berbasis *Vue* dengan teknik *Rendering* sisi server (SSR). Menurut Zaidan dan Sutiyano (2025), *Nuxt.js* dipilih karena stabilitas waktu respons yang lebih baik dan

mengurangi kesalahan dibandingkan dengan kerangka kerja lainnya [5].

b. Node.js

Node.js berperan sebagai *runtime* lingkungan server. Meskipun terdapat teknologi lainnya seperti *bun*, *deno*, dan yang lainnya. *Node.js* tetap digunakan karena kematangannya dalam menjaga keseimbangan antara efisiensi memori dan waktu respons pada sistem skala luas, serta komunitas yang solid.

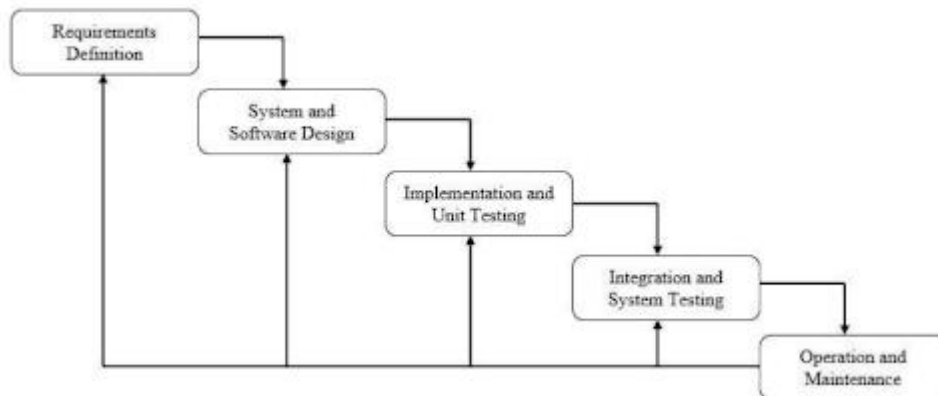
Uraian singkat mengenai studi kasus/objek penelitian sebagai berikut :

a. Profil Objek Penelitian : PT Labtech Penta International

PT Labtech Penta International adalah perusahaan yang bergerak dibidang alat peraga pendidikan teknik dan pengembangan konten digital. fokus utama penelitian ini adalah mengoptimalkan alur manajemen aset yang berjumlah lebih dari 1000 konten. Karna PT Labtech Sebelumnya menyimpan aset digital di server lokal yang membuat pihak lain kesulitan dalam kolaborasi.

2.3. Metode Pengembangan Produk

Metode pengembangan perangkat lunak dalam penelitian ini adalah metode *waterfall*. *Waterfall* adalah salah satu metode yang ada di dunia, metode ini hanya dapat dicapai jika seseorang menyelesaikan setiap tahap, tanpa gagal melewati yang sebelumnya. Metodologi ini dipilih untuk memberikan pengaturan yang transparan, terdokumentasi, dan lengkap di setiap tahap bagi pengembang repositori sistem PT Labtech Penta International. Tahapannya adalah sebagai berikut:



Gambar 1. *Waterfall*

1. *Requirements Definition*

Menurut Mutadin dan Yulianti (2023) menjelaskan bahwa langkah ini merupakan fase pengembangan sistem dengan memperoleh data yang menjadi kebutuhan sistem sehingga spesifikasi dan batasan perangkat lunak dapat dijelaskan. Pengembang sistem menggunakan kerangka observasi, di mana masalah PT Labtech Penta International yang dialami oleh para *programmer* dan pihak lain yang terlibat, serta data aset digital dikumpulkan.

2. *System and Software Design*

Ini adalah tahap di mana ditentukan arsitektur sistem apa yang akan digunakan dalam pengembangan sistem, misalnya UML (*Unified Modeling Language*) seperti *Use Case Diagram*, perancangan skema basis data *MongoDB*, dan kerangka desain antarmuka yang akan digunakan.

3. *Implementation and Unit Testing*

Tahap Implementasi melibatkan penerjemahan desain ke bahasa pemrograman, yang mana pengkodean akan dilakukan di sisi klien dengan *Nuxt.js* dan di sisi *backend* dengan *Node.js* untuk menangani logika *RESTful API*.

4. *Integration and System Testing*

Pada repositori konten digital, pengembang sistem akan melakukan pengujian untuk mengetahui fitur apa yang tersedia dan bahwa sistem dapat berfungsi dengan baik.

5. *Operation and Maintenance*

Selain itu, pada tahap terakhir (tahap Operasi dan Pemeliharaan), sistem sudah berjalan sesuai harapan dan pemeliharaan dilakukan jika diperlukan.

BAB III ANALISIS DAN PERANCANGAN

Bab ini menyajikan proses analisis kebutuhan dan perancangan sistem yang menyediakan solusi untuk masalah manajemen aset digital di PT Labtech Penta International.

3.1. Analisis Kebutuhan

Bagian ini menguraikan kondisi alur kerja yang sedang berjalan saat ini serta gambaran umum sistem yang diusulkan sebagai solusi teknis.

3.1.1 Proses Bisnis Berjalan

Konten digital di PT Labtech Penta International sangat besar, mencapai lebih dari 1000 unit konten web statis. Namun, ada beberapa kendala dalam proses manajemen asetnya:

a Deskripsi naratif

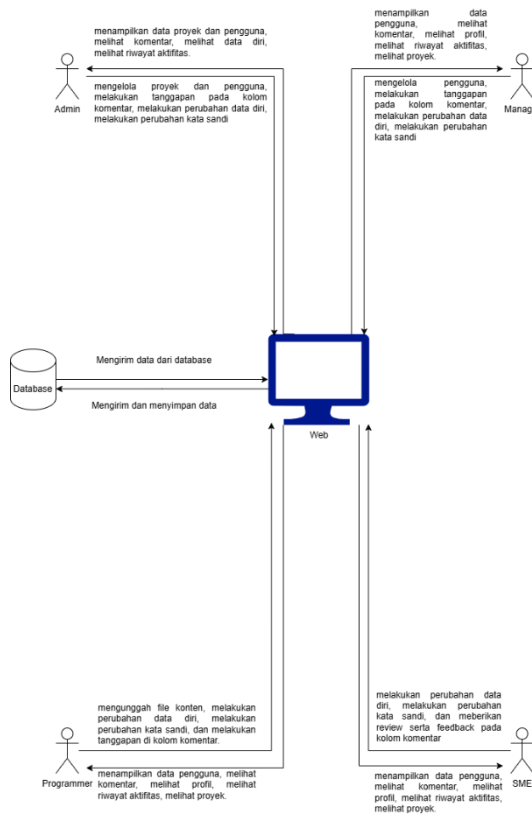
Saat ini, setiap *programmer* menyimpan aset pengembangan secara lokal di perangkat masing-masing. Ketika konten memerlukan tinjauan, *programmer* harus mengirim file melalui WhatsApp atau Email kepada Subject Matter Expert (SME).

b Permasalahan

Karena mekanisme ini, data menjadi terdesentralisasi dan sulit untuk dilacak atau dilihat oleh orang lain. Hal ini dapat menyebabkan umpan balik tertunda, menyebabkan perbaikan aset sering kehilangan momentum.

3.1.2 Gambaran Umum Sistem

Sistem yang diusulkan adalah Repositori Konten Digital Berbasis Web yang menghubungkan fitur penyimpanan dengan platform tinjauan kolaboratif. Di bawah struktur ini, para *programmer* dapat mengunggah aset ke situs penyimpanan yang sama dan para SME dapat memberikan rekomendasi langsung terkait kemajuan mereka tentang aset tersebut dan komentar ini dapat disampaikan ke situs web melalui antarmuka situs web berbasis *Nuxt.js*. Untuk basis data *MongoDB*, setiap set beserta riwayat komentarnya disimpan dengan cara yang terstruktur.



Gambar 2. Gambaran Umum Sistem

3.2. Perancangan

Perancangan ini dilakukan berdasarkan identifikasi masalah untuk memberikan spesifikasi teknis sebelum tahap implementasi.

3.2.3 Kebutuhan Fungsional

Berikut adalah daftar fungsi utama yang harus disediakan oleh sistem, sebagai berikut :

FR-01: Pengguna (*Admin, Manager, Programmer, SME*) dapat melakukan login untuk masuk ke dalam sistem.

FR-02: Pengguna (*Admin, Manager*) dapat melakukan pengelolaan data users (Tambah, Lihat, Ubah, Hapus) melalui fitur *User Management*.

FR-03: *Admin* dapat melakukan pengelolaan data Category (Tambah, Lihat, Ubah, Hapus) melalui fitur *Category Management*.

FR-04: *Admin* dapat melakukan pengelolaan data *Module* (Tambah, Lihat, Ubah, Hapus) melalui fitur *Module management*.

FR-05: *Admin* dapat melakukan *upload landing page* untuk menentukan tampilan pratinjau pada setiap modul.

FR-06: *Admin* dapat melakukan *sign task* untuk menugaskan pengerjaan modul pada programmer.

FR-07: Pengguna (*Admin, Manager, Programmer, SME*) dapat *logout* dari sistem

FR-08: Pengguna (*Admin, Manager, Programmer, SME*) dapat *update profile*

FR-09: Pengguna (*Admin, Manager, Programmer, SME*) dapat melakukan *comment*.

FR-10: Pengguna (*Admin, Manager, Programmer, SME*) dapat melihat *project* melalui fitur *View Projects*.

FR-11: Pengguna (*Admin, Manager, Programmer, SME*) dapat melihat *log activity* melalui fitur *View Log Activity*.

FR-12: Pengguna (*Admin, Manager, Programmer, SME*) dapat melakukan perubahan password melalui fitur *Change Password*.

FR-13: *Programmer* dapat melakukan *upload content* berdasarkan *sign task* yang diberikan oleh *admin*.

3.2.4 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah spesifikasi teknis yang tidak secara langsung merujuk pada fungsi sistem tetapi tetap penting untuk menjaga kinerja dan keberlanjutan sistem. Berikut adalah persyaratan non-fungsional:

NFR-01: Sistem harus mampu membatasi akses menu dan fitur sesuai dengan peran pengguna yang diperoleh dari proses *login*. Hal ini untuk memastikan bahwa data aset digital hanya dapat diakses dan dikelola oleh pihak yang berwenang.

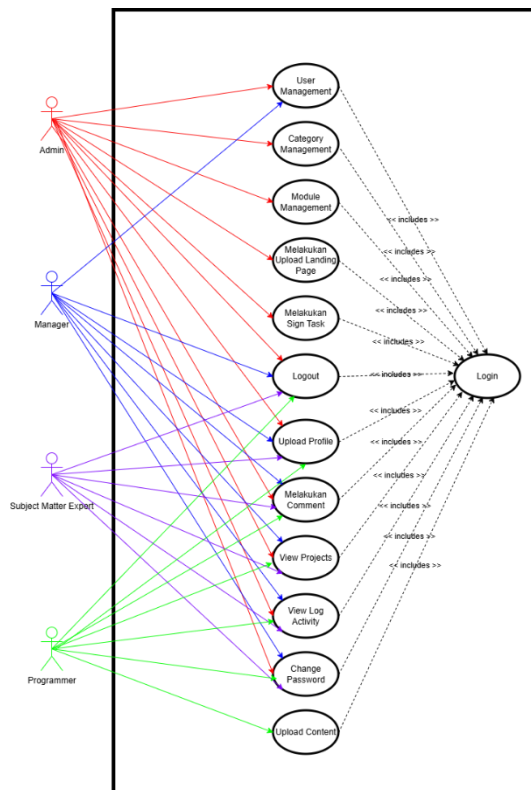
NFR-02: Persyaratan perangkat keras minimum untuk menjalankan sistem ini adalah perangkat keras seperti laptop atau komputer yang berada dalam jaringan yang sama dengan perusahaan. Tidak ada batasan spesifik yang harus dimiliki perangkat keras ini untuk dapat menjalankan sistem ini.

NFR-03: Untuk menjalankan sistem ini, perangkat pengguna setidaknya harus memiliki peramban seperti *Chrome, Firefox, Microsoft Edge*, atau peramban lainnya. Perangkat lunak yang digunakan dalam proses

pengembangan sistem meliputi *Code Editor (Visual Studio Code)*, Peramban, Alat basis data (*MongoDB*), *Tailwind CSS*, *Nuxt.js*, *Node.js*.

3.2.5 Use Case Diagram

Berdasarkan identifikasi kebutuhan fungsional yang telah dipaparkan sebelumnya, interaksi antara aktor dan sistem dipresentasikan melalui *Use Case Diagram* yang ditunjukkan pada Gambar 3 berikut ini.



Gambar 3. *Use Case Diagram*

3.2.6 Skenario Use Case

Skenario *use case* adalah skenario yang menggambarkan perilaku aktor terhadap sistem serta respons yang diberikan oleh sistem kepada aktor.

1. Skenario *Use Case Diagram - Login*

Skenario *use case diagram login* adalah skenario yang menjelaskan tahap yang dilakukan aktor terhadap sistem sebelum masuk ke dalam *website* dan respons yang akan diberikan oleh sistem akan dijelaskan pada Tabel 2.

Tabel 2. Skenario *Use Case Diagram - Login*

Use Case	:	<i>Login</i>
Deskripsi	:	<i>Use Case</i> ini mengenai verifikasi akun sebelum masuk ke dalam sistem.
Aktor	:	<i>Admin, Manager, Subject Matter Expert (SME), and Programmer.</i>
Kondisi Awal	:	Sistem menampilkan <i>Form Login</i>
Kondisi Berhasil	:	Sistem akan meneruskan (<i>Redirect</i>) ke halaman <i>Dashboard</i> berdasarkan peran setiap pengguna.
Kondisi Error	:	Sistem akan memberikan pesan <i>error</i> (<i>Invalid Email or Password</i>)
Skenario Normal		
Aktor melakukan <i>Input Email</i> dan <i>password</i> kedalam <i>form</i> .		
		Sistem akan melakukan validasi terhadap kredensial dengan yang ada di <i>database</i> .
		Sistem akan menampilkan halaman <i>dashboard</i> berdasarkan setiap peran masing-masing aktor.
Skenario <i>Error</i>		
Aktor melakukan <i>Input Email</i> dan <i>password</i> yang tidak sesuai dengan kredensial pengguna atau		

melakukan <i>typo</i> saat pengisian <i>form</i> .	
	Sistem Gagal memverifikasi akun karena tidak sesuai atau tidak ada di <i>database</i> .
	Sistem akan menampilkan pesan <i>error (Invalid Email or password)</i> .
	Aktor tetap dalam halaman <i>login</i> .

2. Skenario *Use Case Diagram – User Management*

Skenario *use case diagram CRUD Users* adalah skenario yang menjelaskan tahap yang dilakukan aktor terhadap sistem mengenai pengguna (*users*) seperti menambah, melihat, mengubah, dan menghapus, respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 3.

Tabel 3. Skenario *Use Case Diagram – User Management*

Use Case	:	<i>User Management</i>
Deskripsi	:	<i>Use Case</i> ini mengenai pengelolaan pengguna (<i>users</i>).
Aktor	:	<i>Admin dan Manager</i> .
Kondisi Awal	:	Aktor sudah <i>login</i> , dan berada di halaman <i>management users</i> .
Kondisi Berhasil	:	Sistem akan menampilkan pesan sukses dan menampilkan data pengguna baru yang akan muncul di tabel daftar <i>user</i> .
Kondisi Error	:	Dalam kondisi ini, sistem akan menampilkan 2 kondisi, sebagai berikut :

		1. Jikalau Aktor melakukan duplikat email saat <i>create user</i>, sistem akan menampilkan email sudah diregistrasi. 2. Jikalau Aktor saat mengisi <i>form confirm password</i> dan tidak sama dengan <i>form password</i>, maka akan menampilkan pesan <i>password tidak sama</i>.
Skenario Normal		
Aktor menekan tombol Tambah (<i>Create</i>) untuk menampilkan formulir <i>input</i> data pengguna baru.		
		Sistem akan menampilkan modal <i>form input</i> data.
Aktor melakukan pengisian data dengan benar dan format yang sesuai.		
Aktor akan menekan tombol <i>save</i> .		
		Sistem melakukan validasi data pada sisi <i>client</i> dan <i>server</i> . Pada sisi <i>frontend</i> , sistem memastikan seluruh input telah sesuai dengan format yang ditentukan (seperti tipe data <i>string</i>). Selanjutnya, pada sisi <i>server</i> , sistem melakukan pemeriksaan akhir untuk memastikan validitas data

	dan melakukan pengecekan <i>redundancy</i> agar tidak terjadi duplikasi data pada basis data.
	Sistem akan menutup modal secara otomatis.
	Sistem akan menampilkan <i>toast notification</i> sukses.
Aktor memilih data pengguna yang ingin diubah dengan menekan tombol Ubah (<i>Edit</i>)	
	Sistem akan menampilkan modal <i>form input</i> dengan kondisi <i>input</i> email tidak dapat diubah atau di kunci.
Aktor akan melakukan perubahan data seperti <i>username</i> , peran pengguna (<i>role</i>), dan <i>password</i> dengan benar sesuai standar.	
	Sistem akan melakukan validasi data yang diisi oleh aktor dengan data yang ada di <i>database</i> , jikalau ada di <i>database</i> maka data akan diperbarui.
	Sistem akan menampilkan <i>toast notification success</i> , dan menampilkan perubahan data pada pengguna yang telah dipilih sebelumnya.

Aktor memilih salah satu baris data pada tabel pengguna dan menekan tombol Hapus.	
	Sistem akan menampilkan <i>pop-up</i> (modal) untuk memastikan bahwa aktor yakin untuk menghapus data tersebut.
Aktor memilih ya	
	Sistem melakukan validasi dengan membandingkan <i>input</i> terhadap data yang tersimpan di dalam basis data. Jika data ditemukan, sistem akan mengeksekusi perintah pembaruan (<i>update</i>) untuk menggantikan data lama dengan data terbaru, kemudian melakukan sinkronisasi status agar perubahan tersebut langsung tercermin pada sisi <i>client</i> (antarmuka pengguna).
	Sistem menampilkan <i>notification toast success delete</i> , data pengguna yang sebelumnya dipilih, tidak akan ditampilkan kembali di daftar pengguna (<i>user</i>).
Aktor memilih tidak (<i>close</i>)	
	Sistem akan menutup modal <i>remove users</i> .

Skenario <i>Error A</i>	
Aktor menekan tombol Tambah (<i>Create</i>) untuk menampilkan formulir <i>input</i> data pengguna baru.	
	Sistem akan menampilkan modal <i>form input</i> data.
Aktor akan melakukan pengisian data <i>password</i> dan <i>confirm password</i> tidak sama	
	Sistem akan mendeteksi dan menampilkan <i>error</i> atau notifikasi bahwa <i>password</i> tidak sama.
Skenario <i>Error B</i>	
Aktor menekan tombol Tambah (<i>Create</i>) untuk menampilkan formulir <i>input</i> data pengguna baru.	
	Sistem akan menampilkan modal <i>form input</i> data.
Aktor akan melakukan pengisian data email yang sama dengan pengguna lainnya.	
Aktor menekan tombol <i>save</i> .	
	Sistem akan melakukan validasi data.
	Sistem mendeteksi bahwa salah satu data yang diisi oleh aktor (email) sama dengan yang ada di

	<i>database</i> . Oleh sebab itu sistem akan menampilkan pesan <i>error</i> bahwa ada duplikat data.
--	--

3. Skenario *Use Case Diagram – Category Management*

Skenario *use case diagram Category* adalah skenario yang menjelaskan tahap yang dilakukan aktor terhadap sistem mengenai data kategori seperti menambah, melihat, mengubah dan menghapus, respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 4.

Tabel 4. Skenario *Use Case Diagram – Category Management*

Use Case	:	<i>CRUD Category</i>
Deskripsi	:	<i>Use Case</i> ini mengenai pengelolaan kategori.
Aktor	:	Admin
Kondisi Awal	:	Aktor sudah <i>login</i> dan sedang berada di halaman kategori.
Kondisi Berhasil	:	Sistem akan menampilkan <i>toast notification success</i> dan menampilkan data kategori baru di daftar kategori.
Kondisi <i>Error</i>	:	Dalam kondisi ini, sistem akan menampilkan pesan <i>error</i> ketika aktor melakukan duplikat data saat pengisian kategori.
Skenario Normal		
Aktor menekan tombol tambah atau <i>create</i> untuk menambahkan kategori baru.		
		Sistem menampilkan modal formulir input kategori.

Aktor melakukan pengisian data nama kategori dengan benar.	
	Sistem menerima inputan sementara pada sisi <i>client</i> .
Aktor menekan tombol <i>save</i> .	
	1. Sistem melakukan validasi data pada sisi <i>client</i> dan <i>server</i>. 2. Sistem memastikan nama kategori belum terdaftar di basis data (<i>check uniqueness</i>).
	Sistem akan menampilkan <i>toast notification success</i> dan menampilkan data kategori baru di daftar kategori.
Aktor menekan tombol Ubah (<i>Edit</i>) pada salah satu kategori.	
	Sistem menampilkan modal input yang telah terisi data lama.
Aktor melakukan perubahan data dan menekan tombol <i>Update</i> .	
	Sistem memvalidasi data dan melakukan sinkronisasi pembaruan pada basis data.
	Sistem menutup modal, menampilkan <i>toast notification success</i> , dan memperbarui tampilan daftar kategori.

Aktor menekan tombol <i>Delete</i> pada salah satu kategori.	
	Sistem menampilkan modal konfirmasi penghapusan untuk memastikan tindakan aktor.
Aktor memilih opsi "Ya" pada modal konfirmasi.	
	1. Sistem memvalidasi data dan melakukan penghapusan pada basis data. 2. Sistem melakukan <i>Cascade Delete</i> terhadap seluruh data Modul dan Konten yang berelasi dengan kategori tersebut. 3. Sistem juga melakukan penghapusan pada folder kategori yang di dalamnya ada beberapa folder modul.
	Sistem menampilkan <i>toast notification success</i> dan memperbarui tampilan data kategori serta aset terkait.
Aktor memilih tidak atau <i>close</i>	
	Sistem menutup modal konfirmasi tanpa melakukan perubahan apapun pada basis data.

Skenario <i>Error</i>	
Aktor melakukan pengisian data kategori yang sudah ada sebelumnya.	
	Sistem akan menerima inputan di sisi <i>client</i>
Aktor menekan tombol <i>save</i>	
	Sistem melakukan validasi ke basis data dan menemukan bahwa data sudah tersedia (<i>Duplicate Entry</i>).
	Sistem tetap menampilkan modal input dan memberikan pesan error " <i>Duplicated data, please review your input</i> " kepada aktor.

4. Skenario Gabungan *Use Case Module Management, Sign Task* dan *Upload Landing Page*

Skenario *use case diagram* ini menjelaskan tahapan yang dilakukan oleh aktor untuk mengelola data *module*, yang mencakup proses pengunggahan *Landing Page* serta penugasan kepada *Programmer* (*sign task*) dalam satu rangkaian aktivitas. Detail interaksi antara aktor dan sistem akan dijelaskan pada Tabel 5.

Tabel 5. Skenario *Use Case – Module Management, Sign Task and Upload Landing page*

Use Case	:	<i>CRUD Module, Sign Task dan Upload Landing Page</i>
----------	---	---

Deskripsi	:	Skenario ini untuk pengelolaan modul serta melakukan <i>sign task</i> pada <i>programmer</i> dan melakukan <i>upload landing page</i> .
Aktor	:	Admin
Kondisi Awal	:	Aktor sudah <i>login</i> dan sedang di halaman modul.
Kondisi Berhasil	:	Sistem akan menampilkan <i>toast notification success</i> menampilkan perubahan pada data modul.
Kondisi <i>Error</i>	:	Sistem akan menampilkan pesan <i>error</i> seperti pesan duplikasi data.
Skenario Normal		
Aktor menekan tombol Tambah (<i>Add</i>) pada daftar modul.		
		Sistem menampilkan modal formulir penambahan modul baru.
Aktor memilih kategori, memasukkan nama modul, menunjuk pengguna (<i>Assign To</i>), dan mengunggah file <i>.zip landing page</i> .		
Aktor menekan tombol <i>Save</i> .		
		1. Sistem memvalidasi kelengkapan data dan format <i>file</i> yang diunggah (<i>.zip</i>).

	2. Sistem memastikan nama modul bersifat unik (<i>check uniqueness</i>).
	Sistem menyimpan data ke basis data, memberikan notifikasi sukses, dan memperbarui status penugasan pada pengguna terkait.
Aktor menekan tombol Ubah (<i>Edit</i>) pada modul yang dipilih.	
	Sistem menampilkan modal formulir dengan data modul yaitu nama modul.
Aktor mengubah informasi nama modul.	
	Sistem menerima perubahan inputan pada sisi <i>client</i> .
Aktor menekan tombol <i>Update/Save</i> .	
	1. Sistem memvalidasi nama modul yang baru. 2. Sistem melakukan pengecekan ke basis data untuk memastikan nama modul tidak duplikat dengan modul lain.
	Sistem memperbarui nama modul di basis data, menutup modal, dan menampilkan <i>toast notification success</i> .

Aktor memilih data modul dan menekan tombol Hapus	
	Sistem menampilkan modal konfirmasi penghapusan data untuk mencegah kesalahan aksi.
Aktor memilih opsi "Ya".	
	1. Sistem memvalidasi kebenaran data di basis data. 2. Sistem menghapus data modul dan secara otomatis menghapus seluruh data Konten yang berelasi dengan modul tersebut (<i>Cascade Delete</i>). 3. Sistem akan menghapus folder modul yang berisi beberapa folder konten.
	Sistem menampilkan <i>toast notification success</i> , menghapus baris data dari tabel, dan memastikan integritas data tetap terjaga.
Skenario <i>Error</i>	
Aktor memasukkan nama modul yang sudah terdaftar sebelumnya di dalam sistem	
	Sistem menerima inputan nama modul tersebut.

Aktor menekan tombol <i>Save</i> .	
	Sistem melakukan validasi ke basis data dan mendeteksi bahwa nama modul tersebut sudah digunakan (<i>Duplicate Entry</i>).
	Sistem menampilkan pesan <i>error</i> " <i>Duplicated data, Please Review your input</i> ", untuk mencegah penyimpanan, dan tetap pada modal formulir agar aktor dapat memperbaiki data.

5. Skenario *Use Case Diagram - Logout*

Skenario *use case* ini menjelaskan tahap yang dilakukan oleh aktor untuk mengakhiri sesi aktif pada aplikasi dan keluar dari sistem guna menjaga keamanan akun. Respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 6.

Tabel 6. Skenario *Use Case Diagram - Logout*

Use Case	:	<i>Logout</i>
Deskripsi	:	Proses untuk menghancurkan sesi (<i>session/token</i>) pengguna agar akun tidak dapat diakses tanpa <i>login</i> kembali.
Aktor	:	<i>Admin, Manager, programmer, dan SME.</i>
Kondisi Awal	:	Aktor dalam keadaan <i>Logged In</i> (sesi aktif) dan berada di dalam sistem.
Kondisi Berhasil	:	Sesi aktor dihapus, dan aktor diarahkan kembali ke halaman <i>Login</i> .

Kondisi <i>Error</i>	:	Sesi gagal dihapus karena kendala jaringan atau kegagalan pada server.
Skenario Normal		
Aktor menekan tombol profil di sudut kanan atas		
		Sistem akan menampilkan <i>dropdown</i> dan menampilkan <i>options</i> seperti <i>settings</i> dan <i>logout</i> .
Aktor menekan tombol <i>logout</i> .		
		1. Sistem akan menghapus sesi aktif yang ada pada <i>storage</i>. 2. Sistem akan memutuskan koneksi sesi aktor ke sistem.
		Sistem akan mengarahkan aktor ke halaman <i>login</i> .

6. Skenario Diagram *Use Case* - *Update Profile*

Skenario *use case* ini menjelaskan tahap yang dilakukan oleh aktor untuk mengunduh atau memperbarui profil pada akun masing-masing sebagai identitas pengguna. Respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 7.

Tabel 7. Skenario *Use Case Diagram* - *Update Profile*

Use Case	:	<i>Upload Profile</i>
Deskripsi	:	Skenario ini untuk melakukan
Aktor	:	<i>Admin, Manager, Programmer, SME</i>
Kondisi Awal	:	Aktor sudah <i>login</i> dan sedang berada di halaman profil.

Kondisi Berhasil	:	Profil aktor mengalami perubahan.
Skenario Normal		
Aktor menekan tombol <i>update profile</i>		
		Sistem akan menampilkan modal <i>form</i> .
Aktor melakukan perubahan data		
		Sistem akan memvalidasi data dan mengubah data di sistem <i>database</i> .
Aktor menekan foto profil		
		Sistem akan menampilkan <i>form</i> untuk perubahan foto profil.
Aktor memasukkan gambar ke formulir (<i>form</i>).		
		Sistem akan mengubah foto profil aktor.

7. Skenario *Use Case Diagram - Comment*

Skenario *use case* ini menjelaskan tahap yang dilakukan oleh aktor untuk memberikan umpan balik atau catatan dalam bentuk komentar pada modul atau konten yang tersedia. Respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 8.

Tabel 8. *Skenario Use Case Diagram - Comment*

Use Case	:	Skenario <i>Comment</i>
Deskripsi	:	Skenario ini tentang bagaimana cara kerja sistem untuk fitur <i>comment</i> .
Aktor	:	<i>Admin, manager, programmer, and SME</i> .

Kondisi Awal	:	Aktor sudah <i>login</i> dan sudah berada di halaman <i>module</i> tertentu.
Kondisi Berhasil	:	Pesan atau gambar terkirim ke semua aktor.
Skenario Normal		
Aktor mengirim <i>text</i> atau gambar melalui <i>form</i> yang telah disediakan di antarmuka		
		Sistem akan memvalidasinya, apakah sesuai dengan <i>state</i> yang telah ditentukan atau tidak melalui <i>socket</i> .
		Sistem mengirim data ke semua aktor lainnya.
Setiap aktor menerima pesan baru.		

8. Skenario *Use Case Diagram - View Projects*

Skenario *use case* ini menjelaskan tahap yang dilakukan oleh aktor untuk melihat daftar proyek atau modul yang sedang berjalan atau ditugaskan. Respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 9.

Tabel 9. Skenario *Use Case - View Projects*

Use Case	:	<i>View Projects</i>
Deskripsi	:	Skenario ini untuk melihat proyek apa saja yang ada.
Aktor	:	<i>Admin, Manager, Programmer, and SME.</i>
Kondisi Awal	:	Aktor sudah berada di halaman <i>dashboard</i> .

Kondisi Berhasil	:	Dapat melihat atau <i>preview</i> modul yang terdapat di setiap proyek (<i>project</i>).
Skenario Normal		
Aktor melihat daftar proyek atau kategori.		
Aktor memilih salah satu dari daftar proyek atau kategori.		
		Sistem akan melakukan validasi data modul berdasarkan id proyek di sistem.
		Sistem akan mengirim data ke <i>client</i> untuk diolah dan ditampilkan ke aktor.
		Sistem di <i>client</i> akan mengelola data dan melakukan <i>redirect</i> aktor ke <i>dynamic page</i> . yang mana itu akan menampilkan daftar modul yang dimiliki oleh proyek yang telah dipilih.
Aktor melihat daftar modul dan memilih salah satu modul untuk melihat atau <i>preview</i> .		
		Sistem akan <i>redirect</i> aktor ke halaman yang akan melakukan <i>iframe</i> ke modul yang telah dipilih oleh aktor.

9. Skenario *Use Case Diagram - View Log Activity*

Skenario *use case* ini menjelaskan tahap yang dilakukan oleh aktor untuk melihat riwayat aktivitas atau log yang terjadi di dalam sistem guna

keperluan audit atau *monitoring*. Respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 10.

Tabel 10. Skenario *Use Case Diagram – View Log Activity*

Use Case	:	<i>View Log activity.</i>
Deskripsi	:	Skenario ini untuk melihat aktivitas yang terjadi di sistem.
Aktor	:	<i>Admin, Manager, Programmer, and SME.</i>
Kondisi Awal	:	Aktor sudah login dan sedang berada di halaman <i>Log Activity</i> .
Kondisi Berhasil	:	Aktor melihat <i>real-time log activity</i>
Skenario Normal		
Aktor (Admin) sedang melakukan <i>create category</i> atau yang lainnya.		
		Sistem akan menyimpan data di <i>database</i> sebagai data baru jikalau memang tidak ada di <i>database</i> .
		Sistem akan melakukan pengisian data log menggunakan <i>socket</i> agar data terkirim ke semua aktor secara <i>real-time</i>
Semua aktor menerima log baru yang sama.		

10. Skenario *Use Case Diagram - Change Password*

Skenario *use case* ini menjelaskan tahap yang dilakukan oleh aktor untuk melakukan perubahan kata sandi lama menjadi kata sandi baru demi

meningkatkan keamanan akun. Respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 11.

Tabel 11. Skenario *Use Case Diagram - Change Password*

Use Case	:	<i>Change Password</i>
Deskripsi	:	Skenario untuk melakukan perubahan data <i>password</i> .
Aktor	:	<i>Admin, Manager, Programmer, dan SME.</i>
Kondisi Awal	:	Aktor sudah <i>login</i> dan berada di <i>settings profile</i> .
Kondisi Berhasil	:	Aktor mendapatkan <i>password</i> baru.
Kondisi Error	:	Aktor menerima notifikasi pesan <i>error</i> kalau <i>password</i> tidak sesuai dengan format yang berlaku atau <i>old password</i> sama dengan <i>new password</i> .
Skenario Normal		
Aktor menekan tombol <i>security settings</i> .		
		Sistem akan menampilkan modal <i>form</i> seperti <i>old password</i> , <i>new password</i> dan <i>confirm new password</i> .
Aktor mengisi data dengan benar		
		sistem akan melakukan pencarian user berdasarkan id.
		Sistem akan membandingkan <i>old password</i> dengan <i>password user</i> di <i>database</i> .

	Jikalau data sama, maka data seperti <i>password</i> akan mengalami perubahan.
	Sistem akan membiarkan aktor tetap pada halaman tersebut.
Skenario Error	
Aktor melakukan <i>input form</i> dengan <i>old password</i> dan <i>new password</i> yang sama.	
	Sistem akan melakukan validasi <i>user</i> berdasarkan id.
	Sistem akan melakukan pengecekan apakah <i>old password</i> sama dengan yang ada di sistem.
	Jikalau sama, maka sistem akan melakukan pengecekan apakah <i>old password</i> sama dengan <i>new password</i> .
	Jikalau sama, maka sistem akan memberikan notifikasi bahwa <i>new password</i> tidak boleh sama dengan <i>old password</i> .
Aktor melakukan input form.	
	Sistem akan melakukan validasi <i>user</i> berdasarkan id.
	Sistem akan melakukan pengecekan apakah <i>old password</i> sama dengan yang ada pada <i>database</i> .

	Sistem akan memberikan notifikasi bahwa <i>old password</i> tidak benar atau tidak sesuai dengan yang ada di <i>database</i> atau sistem.
--	---

11. Skenario *Use Case Diagram - Upload Content*

Skenario *use case* ini menjelaskan tahap yang dilakukan oleh aktor untuk mengunggah aset digital hasil pengerjaan berdasarkan instruksi penugasan yang telah diberikan sebelumnya. Respons sistem terhadap aktivitas ini akan dijelaskan pada Tabel 12.

Tabel 12. Skenario *Use Case Diagram - Upload Content*

Use Case	:	<i>Upload Content</i>
Deskripsi	:	Skenario untuk <i>upload content</i> berdasarkan tugas yang diberikan oleh admin (aktor).
Aktor	:	<i>Programmer</i> .
Kondisi Awal	:	Aktor sudah <i>login</i> dan sedang berada di halaman <i>upload content</i> .
Kondisi Berhasil	:	Aktor akan melihat notifikasi berhasil dan melihat daftar konten sudah berubah.
Kondisi Error	:	Aktor akan mendapatkan pemberitahuan <i>error</i> .
Skenario Normal		
Aktor melakukan <i>upload content</i> dengan menggunakan <i>file zip</i> .		
		Sistem akan mengirim data ke <i>database</i> .

	Sistem akan mengecek apakah ini <i>file zip</i> atau tidak
	Sistem akan melakukan <i>extract file</i> .
	Sistem akan menyimpan data di <i>database</i> .
	Sistem akan memberikan atau mengirim data ke <i>client</i> untuk ditampilkan ke aktor.
	Sistem akan mengubah data di <i>client</i> berdasarkan data yang baru.
	Sistem akan memberikan pemberitahuan bahwa berhasil melakukan <i>upload</i> .
Aktor akan menerima tampilan baru (dengan daftar yang sudah di <i>update</i>) di daftar <i>upload content</i> .	
Skenario <i>Error</i>	
	Sistem akan mengubah data di <i>client</i> berdasarkan data yang baru.

3.2.7 Sequence diagram

Bagian ini menjelaskan alur kerja dalam sistem pengumpulan konten digital melalui Diagram Urutan. Ini diilustrasikan di sini, dengan langkah awal, autentikasi, dan keterlibatan antara aktor dalam menangani modul dan konten. Menggunakan diagram ini untuk memberikan struktur tentang bagaimana data dan aktivitas mengalir di antara aktor, dapat mengikuti urutan yang jelas sesuai dengan persyaratan fungsional yang disajikan. Rincian tahap interaksi dalam sistem disajikan sebagai berikut:

A. *Sequence Diagram Authentication, Profile, Change Password and Monitoring*

Pada Gambar 4, diagram ini mempresentasikan interaksi dasar yang dapat dilakukan oleh seluruh aktor. Berikut adalah penjelasan singkat proses yang terdapat di Gambar 4 :

a. *Proses Login*

Proses *Login* Proses autentikasi dimulai ketika Aktor memasukkan kredensial dalam bentuk email dan kata sandi melalui antarmuka login. Sistem menangkap data ini dan melakukan verifikasi kode *hash* dengan yang disimpan dalam basis data. Setelah memvalidasi data melalui Logika Autentikasi, sistem akan mengizinkan peran untuk mengarahkan pengguna kembali ke halaman *dashboard* dengan setiap aktor sesuai hak akses.

b. *Proses Update profile & Change Password (Keamanan)*

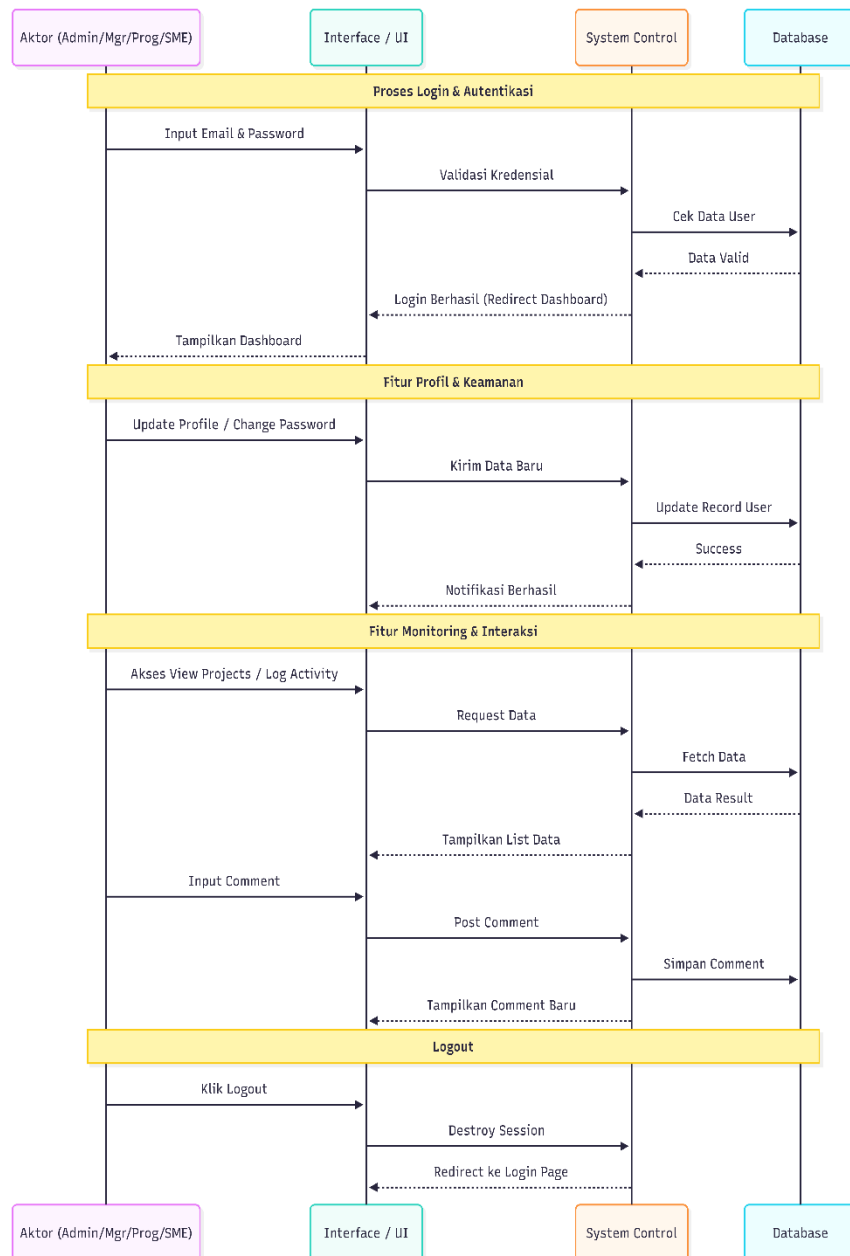
Aktor dapat melakukan pembaruan profil dan kata sandi. Sistem akan mengirim data baru ke *database*.

c. *Monitoring Process (View Projects / View Log Activity)*

Aktor dapat akses untuk melihat project atau kategori yang sudah ada, melihat module apa saja yang ada di project atau kategori tersebut dan melakukan *preview* untuk melihat aset digital tersebut. Aktor juga dapat melihat *log activity*, dan juga *comment*.

d. *Proses Logout*

Proses yang dimana *cookies* pengguna dihapus yang kemudian diarahkan ke halaman *login*.



Gambar 4. *Sequence Diagram Authentication, Profile, Change Password and Monitoring*

B. *Sequence Diagram Category Management*

Pada Gambar 5, diagram ini menjelaskan atau mempresentasikan alur sistem *category management*, berikut adalah penjelasan singkat mengenai alur sistem yang terdapat pada diagram ini, sebagai berikut :

a. *Category Management - create*

Pada tahap ini, Aktor (admin) akan membuat kategori baru seperti sebagai contoh *Automotive*, kategori baru tersebut akan tersimpan di *database* dan juga secara bersamaan akan membuat folder baru berdasarkan nama kategori tersebut.

b. *Category Management – Update*

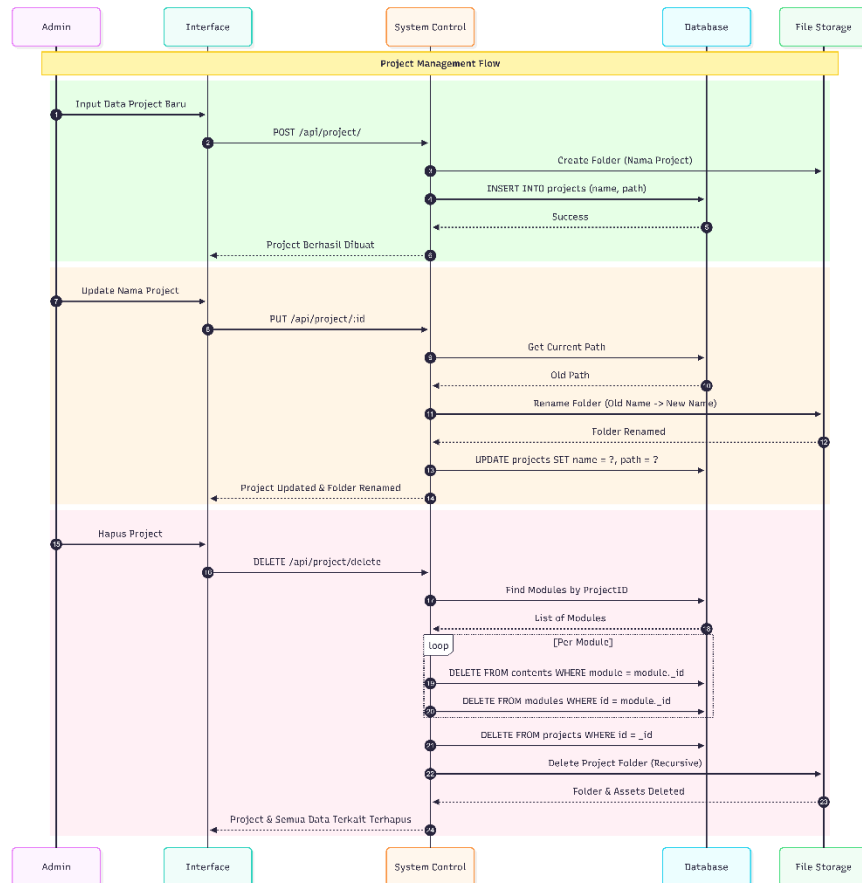
Pada tahap ini, Aktor (Admin) dapat mengubah kategori yang sudah ada di *database*, perubahan yang terjadi pada kategori seperti nama akan mempengaruhi nama folder yang telah kita buat sebelumnya.

c. *Category Management – Delete*

Pada tahap ini, Aktor (Admin) dapat melakukan penghapusan data yang sudah ada sebelumnya, data yang dihapus akan menghapus data yang terkait dengannya berdasarkan dengan *primary key* yang terdapat di *database*.

d. *Module Management, Sign Task and Upload Landing Page.*

Pada tahap ini, aktor harus mengirimkan modul yang sesuai dengan kategori. Pada titik ini, aktor juga akan menandatangani tugas atau penugasan kepada aktor pemrogram, dan akan mengunggah halaman landing. Di sisi server, kami akan mengekstrak file zip dan menyimpannya di penyimpanan file, serta menambah atau memodifikasi data. Pada langkah penghapusan, akan menghapus data modul termasuk data konten; juga akan menghapus folder modul dan file konten.



Gambar 5. *Sequence Diagram Category Management*

C. Sequence Diagram *Module Management*

Pada Gambar 6, diagram ini menjelaskan atau mempresentasikan alur sistem *module management*, berikut adalah penjelasan singkat mengenai alur sistem yang terdapat pada diagram ini, sebagai berikut :

a. *Module Management – Create*

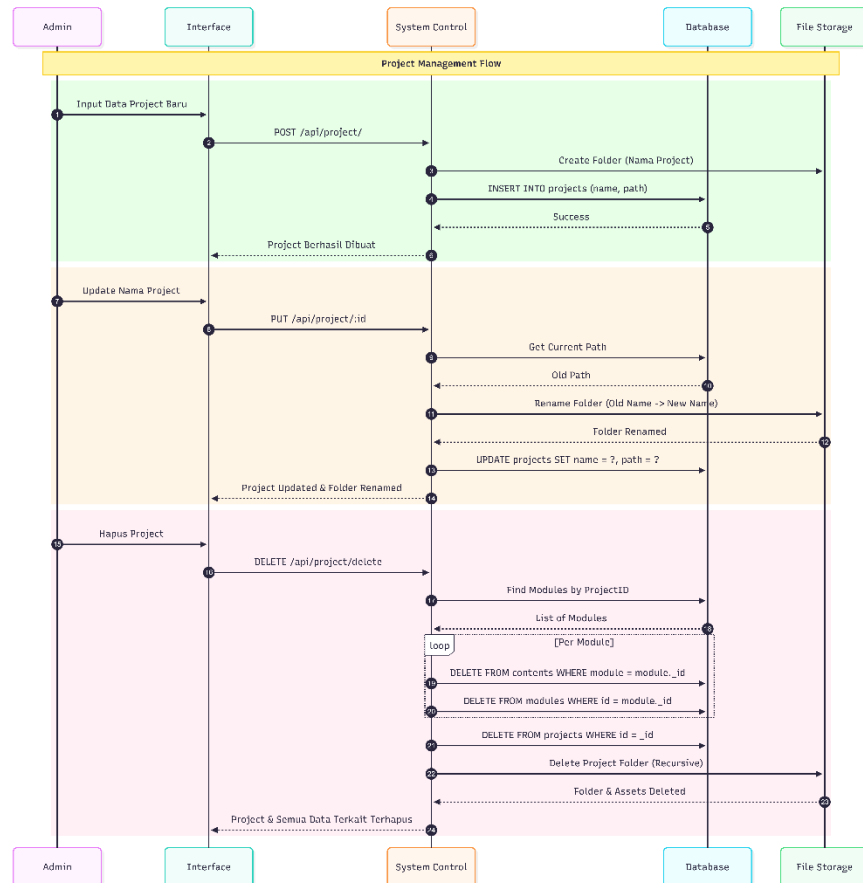
Pada tahap ini, Aktor (Admin) akan membuat data *module* baru yang akan memiliki *relation* atau hubungan terhadap data kategori yang ada. Data modul tersebut bersifat *unique* yang tidak dapat di duplikat dan juga secara bersamaan folder baru akan di buat berdasarakan nama module, folder tersebut akan menjadi sub folder dari folder kategori yang telah kita hubungkan ke data modul. Pada tahap ini juga akan terjadinya proses *assign task* ke aktor *programmer* serta proses *upload landing page*.

b. *Module Management – Update*

Pada tahap ini, Aktor (Admin) akan melakukan perubahan data, perubahan yang ada akan mempengaruhi nama folder yang telah kita buat sebelumnya.

c. *Module Management - Delete*

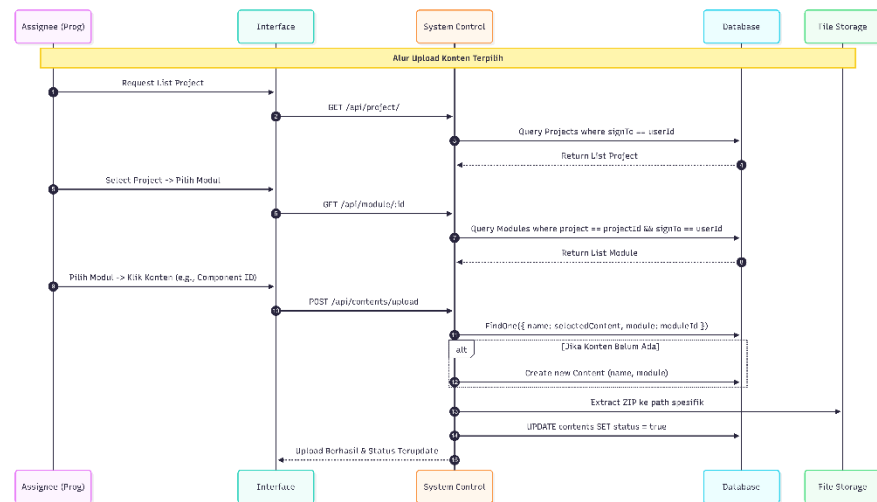
Pada tahap ini, Aktor (admin) akan melakukan penghapusan data yang sudah ada. Data yang akan dihapus akan mempengaruhi data yang terkait seperti data konten (*content*) serta menghapus folder modul serta folder content yang telah ada sebagai sub folder dari folder modul.



Gambar 6. Sequence Diagram Module Management

D. Sequence Diagram *Upload Content*

Pada Gambar 7, diagram ini menjelaskan atau mempresentasikan alur sistem *upload content*. tahap ini Aktor (*Programmer*) yang telah ditugaskan di tahap sebelumnya (*module management*) akan melakukan proses *upload content* berdasarkan kategori dan modul. Data yang diunggah akan diproses di backend, proses *extract file* akan terjadi di sini. *File* yang telah di *extract* akan diletakkan berdasarkan kategori dan modul yang telah ditugaskan.



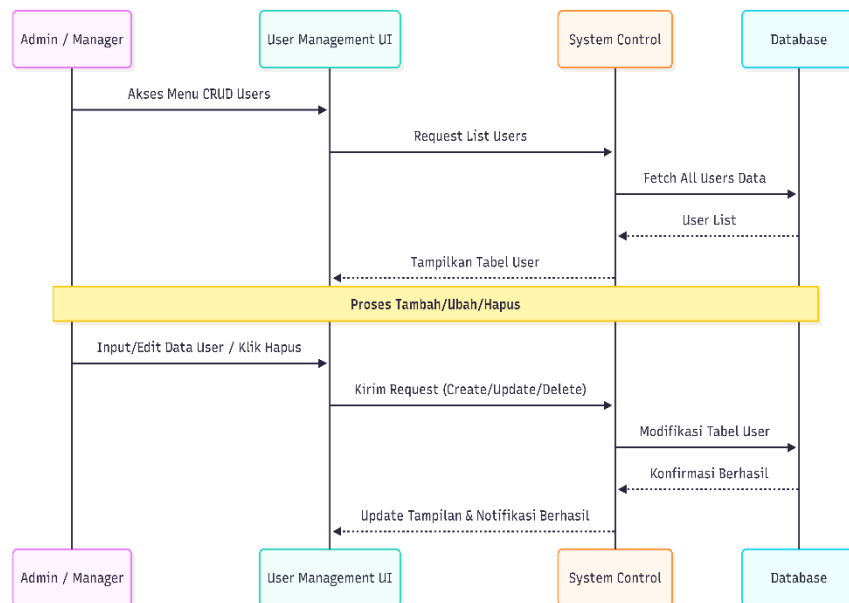
Gambar 7. Sequence Diagram Upload Content

E. Sequence Diagram Users Management

Pada Gambar X, pada diagram ini menjelaskan tentang bagaimana aktor (*admin* dan *manager*) dalam mengelola pengguna. Berikut adalah penjelasan singkat dari diagram ini :

a. Proses *CRUD Users*

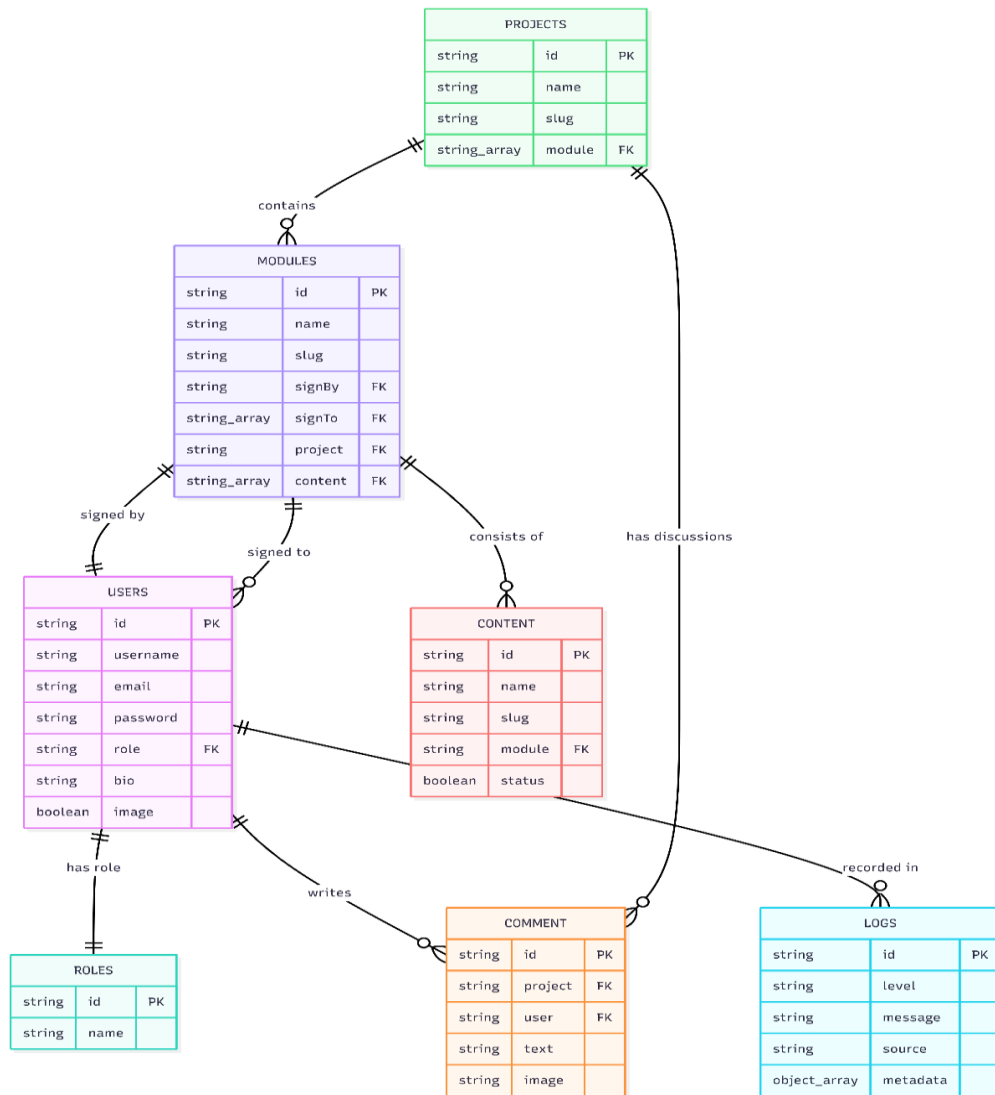
Pada proses ini, para aktor seperti *admin* dan *manager* dapat melakukan perubahan data seperti Tambah, Edit, Melihat dan Menghapus data pengguna.



Gambar 8. *Sequence Diagram Users Management*

3.2.8 ER Diagram

Desain basis data dalam sistem repositori konten digital ini disajikan menggunakan Diagram ER. Diagram ini berfungsi sebagai cetak biru yang memvisualisasikan struktur penyimpanan data, hubungan antara entitas, dan atribut yang melekat pada setiap objek dalam sistem. Desain Diagram ER akan ditampilkan pada Gambar 5.



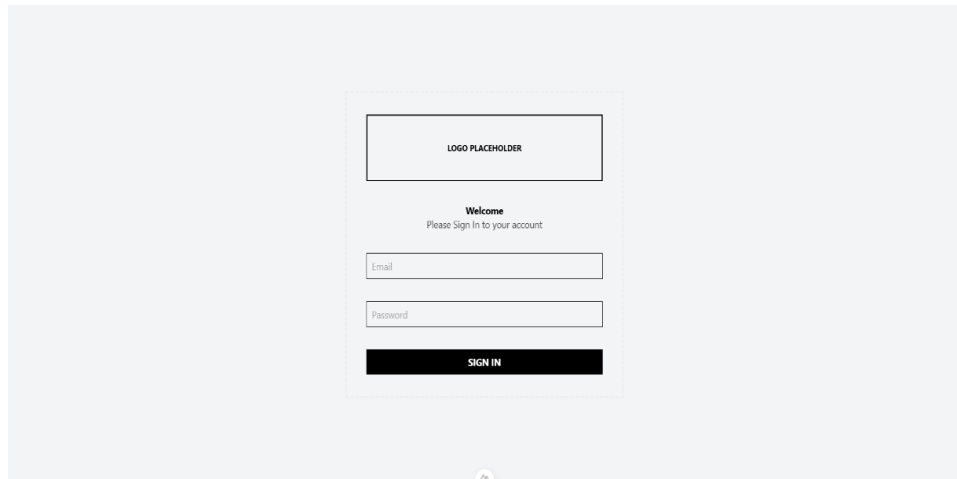
Gambar 9. ER Diagram

3.2.9 Perancangan Antarmuka (*Wireframe*)

Di bagian ini, desainer akan memperkenalkan desain antarmuka atau wireframe sebagai kerangka visual sebelum pengkodean. Dalam desain ini, tujuannya bukan untuk menciptakan desain yang bagus atau indah, melainkan untuk fokus pada tata letak elemen seperti *div*, *section*, atau lainnya. Desain ini akan mencakup semua fungsionalitas saya sebelumnya, yaitu login, manajemen pengguna, manajemen kategori, manajemen modul, manajemen konten, pembaruan profil, ganti kata sandi, melihat proyek, melihat aktivitas log, dan lainnya. Berikut adalah desain antarmuka (*wireframe*):

A. Antarmuka (*Wireframe*) Login

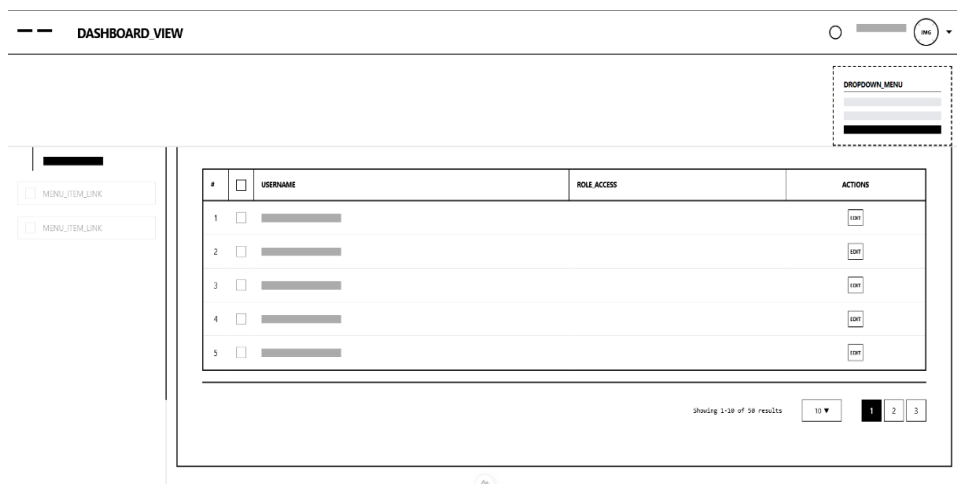
Bagian ini akan menampilkan *wireframe* sebuah halaman *login*



Gambar 10. *Wireframe Login*

B. Antarmuka (*Wireframe*) User management

Bagian ini akan menampilkan *wireframe* tentang proses pengelolaan pengguna atau *users*, seperti proses menambah, melihat, mengubah, dan menghapus.



Gambar 11. *Wireframe User Management - List*

DASHBOARD

MENU LABEL

SUB LABEL ACTIVE

SUB LABEL INACTIVE

DIRECT LINK

USERS MANAGEMENT

	#	USERNAME
☐	1	admin_user
☐	2	dev_user

CREATE USER

USERNAME

EMAIL

PASSWORD

CONFIRM PASSWORD

ROLE

Select...

Gambar 12. Wireframe User Management - Create

DASHBOARD

MENU LABEL

SUB LABEL ACTIVE

SUB LABEL INACTIVE

DIRECT LINK

USERS MANAGEMENT

	#	USERNAME
☐	1	admin_user
☐	2	dev_user

REMOVE USERS

!

This action will delete the data permanently.
CONTINUE?

CLOSE YES, REMOVE

Gambar 13. Wireframe User Management - Delete

DASHBOARD

MENU LABEL

SUB LABEL ACTIVE

SUB LABEL INACTIVE

DIRECT LINK

USERS MANAGEMENT

	#	USERNAME
☐	1	admin_user
☐	2	dev_user

EDIT USERS

USERNAME

EMAIL (LOCKED)

ROLE

NEW PASSWORD

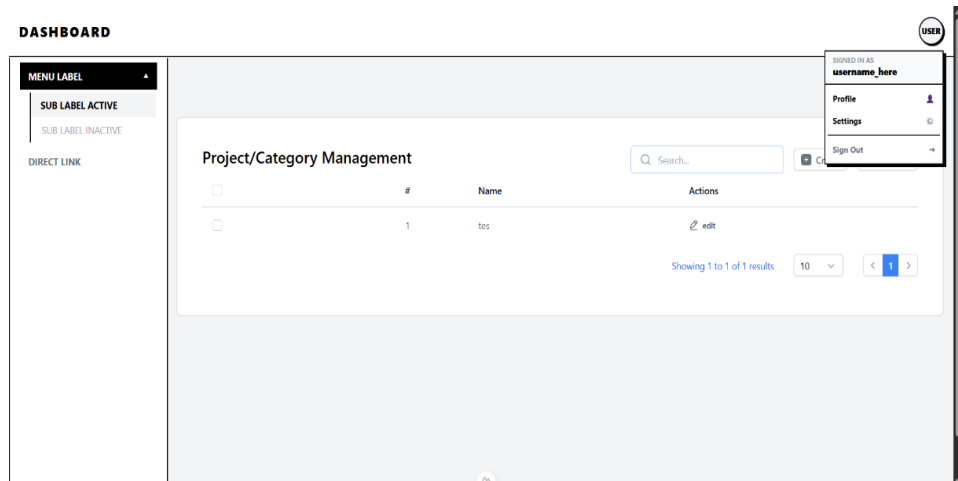
CONFIRM NEW PASSWORD

CLOSE SAVE

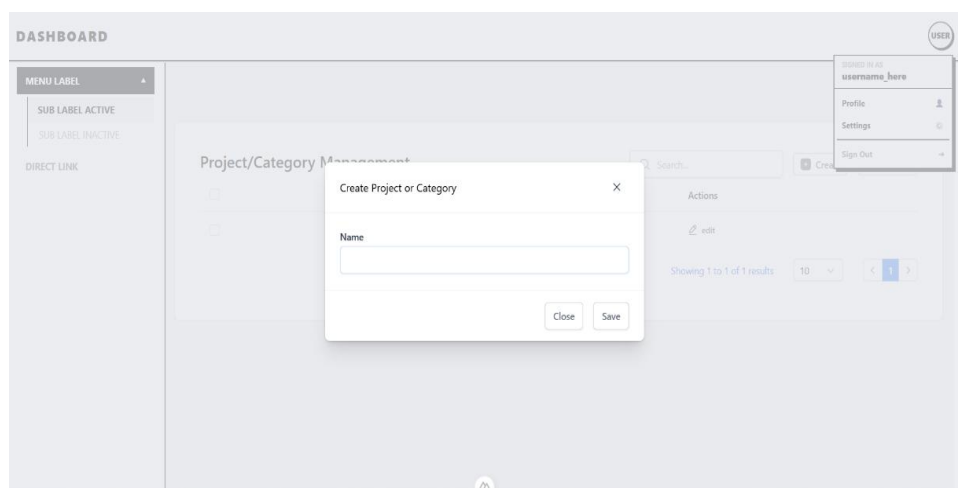
Gambar 14. Wireframe User Management - Update

C. Antarmuka (*Wireframe*) Category Management

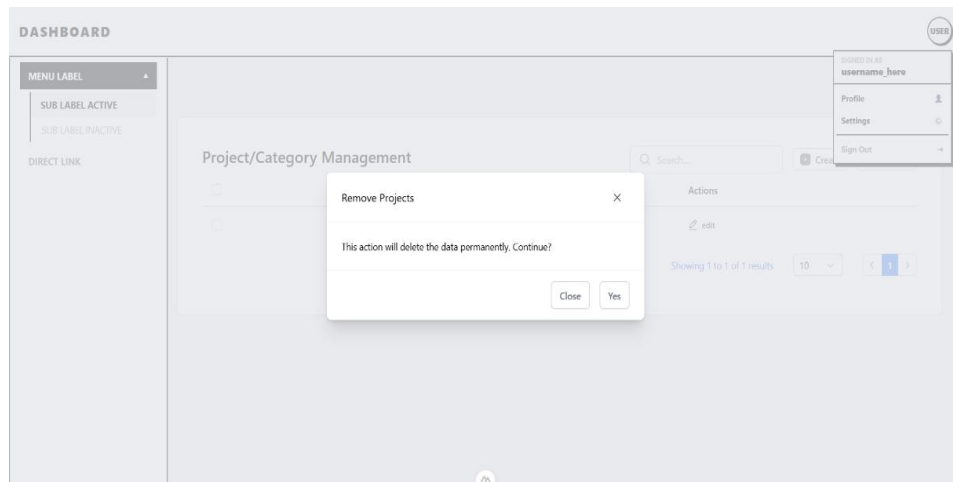
Bagian ini akan menampilkan wireframe tentang proses pengelolaan kategori seperti proses menambah, melihat, mengubah, dan menghapus.



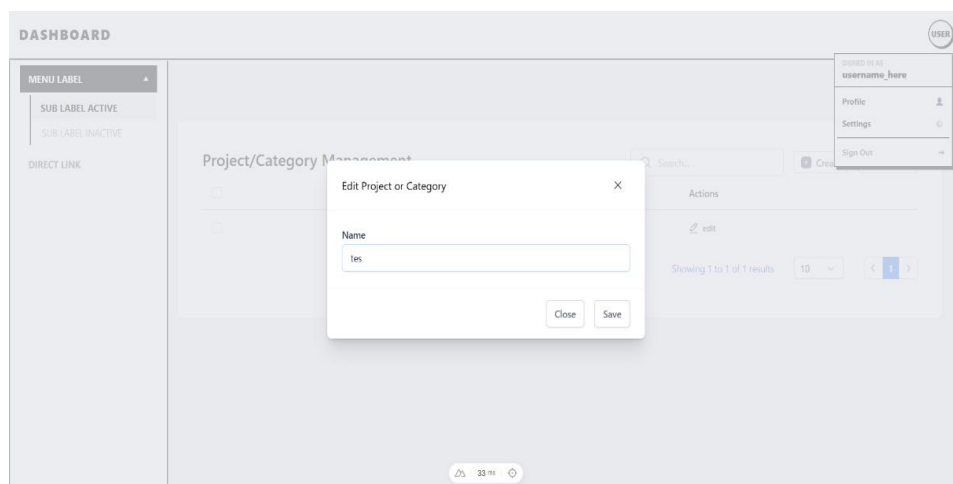
Gambar 15. *Wireframe Category Management - List*



Gambar 16. *Wireframe Category Management – Create*



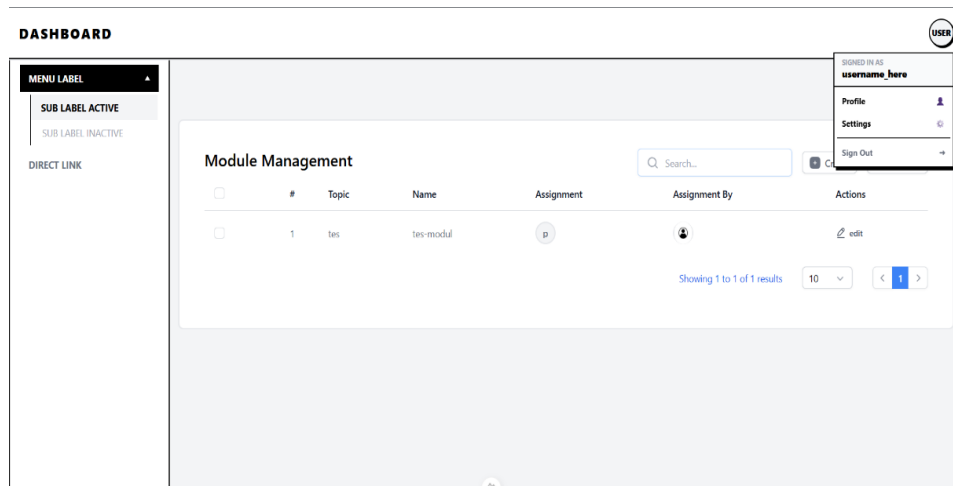
Gambar 17. *Wireframe Category Management - Delete*



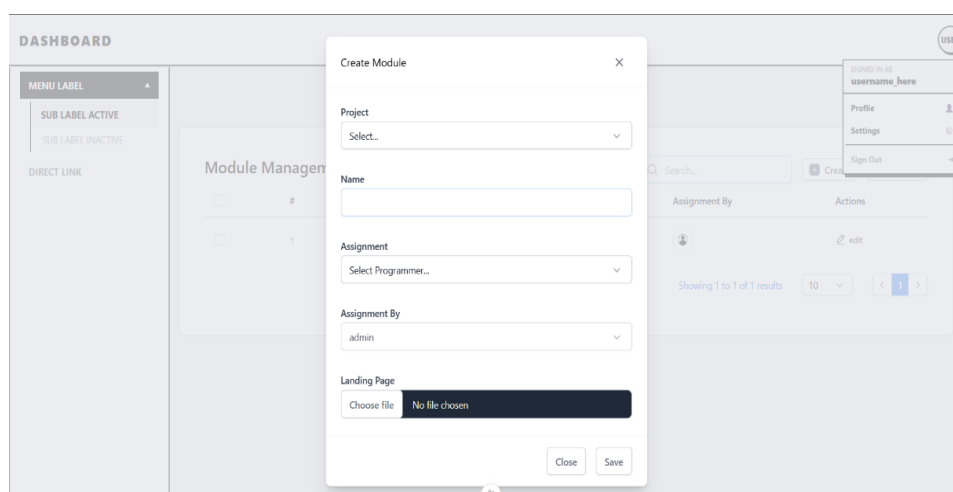
Gambar 18. *Wireframe Category Management - Update*

D. Antarmuka (*Wireframe*) Module Management & Sign Task & Upload Landing page

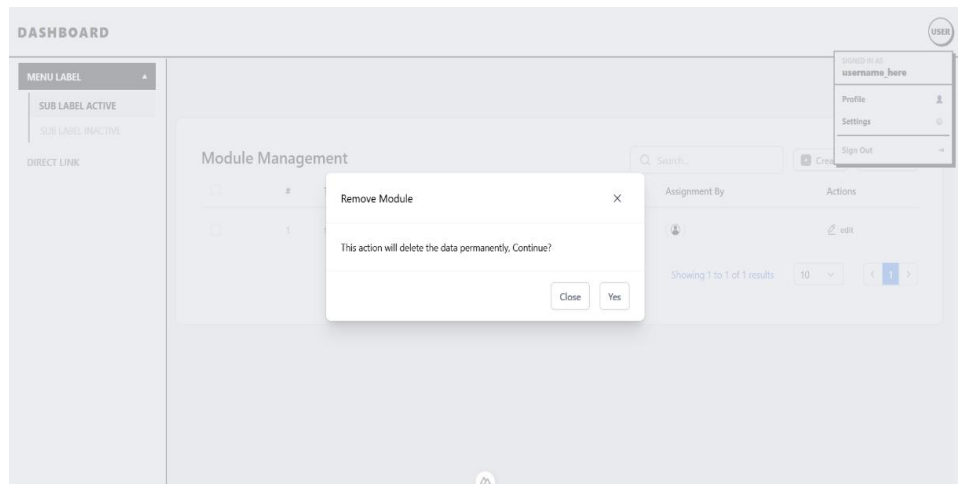
Bagian ini akan menampilkan *wireframe* tentang proses pengelolaan pengguna atau *users*, seperti proses menambah, melihat, mengubah, dan menghapus serta sekaligus proses *sign task* dan proses *upload landing page*.



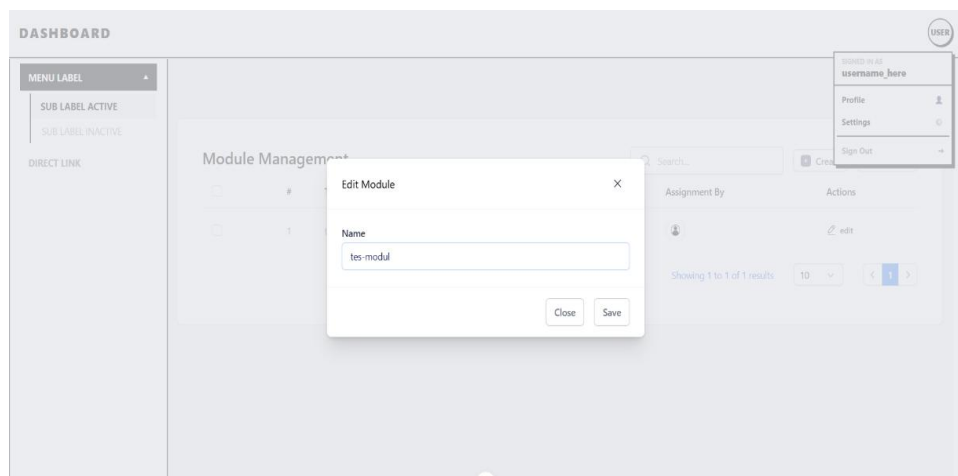
Gambar 19. Wireframe Module Management - List



Gambar 20. Wireframe Module Management - Create



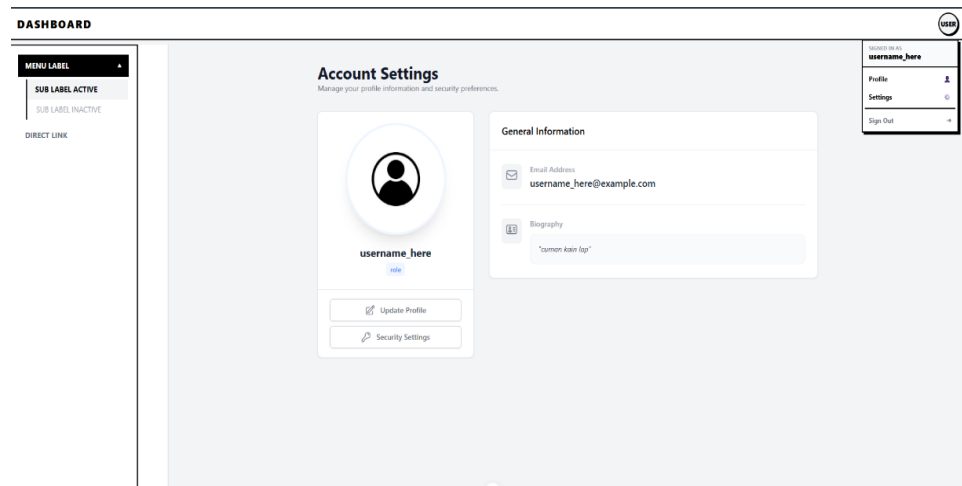
Gambar 21. Wireframe Module Management - Remove



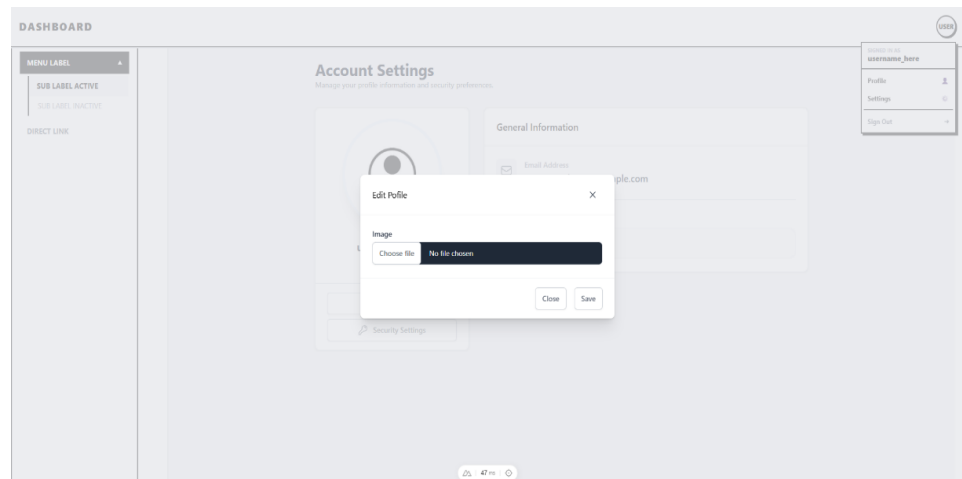
Gambar 22. Wireframe Module Management - Update

E. Antarmuka (Wireframe) Update Profile

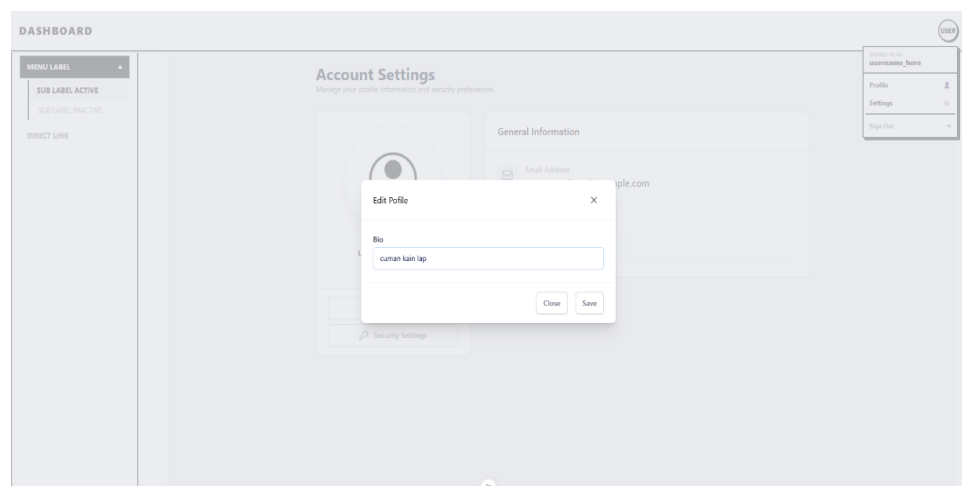
Bagian ini akan menampilkan *wireframe* tentang proses update profil



Gambar 23. Wireframe Profile - View



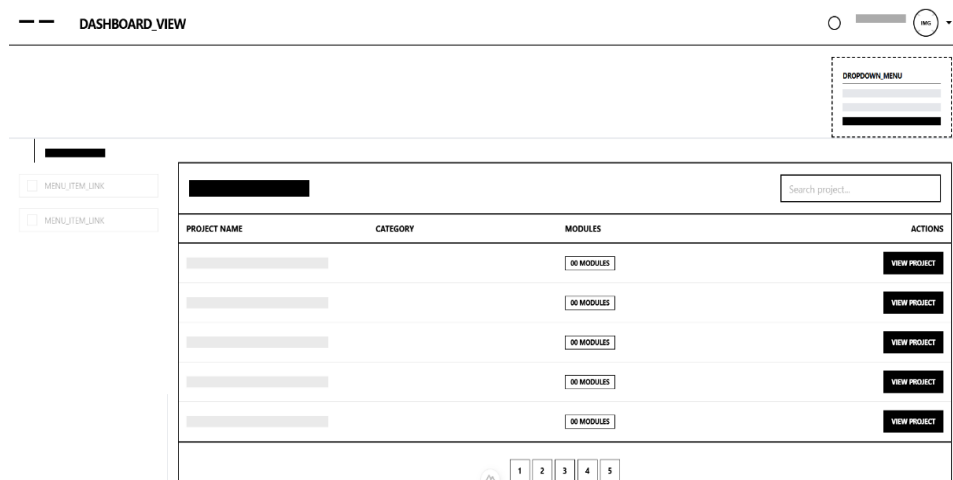
Gambar 24. Wireframe profile - Edit Photo



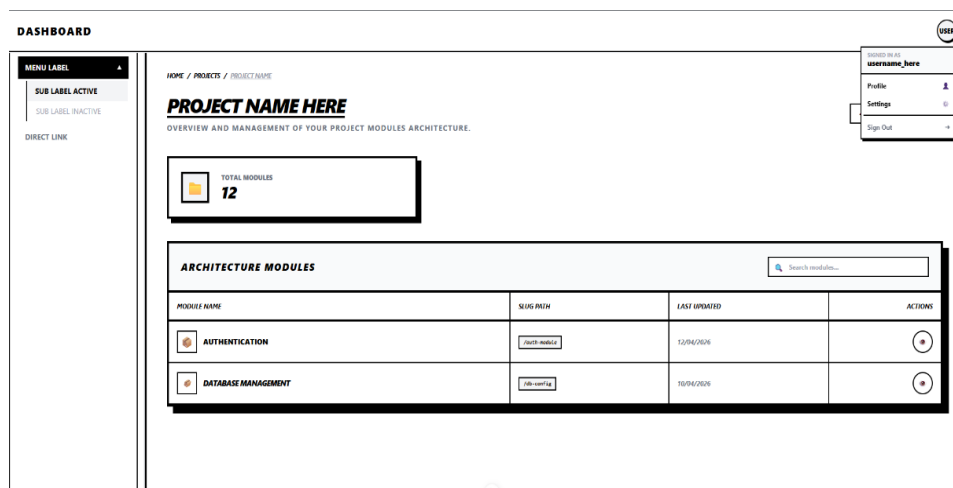
Gambar 25. Wireframe Profile - Edit Bio

F. Antarmuka (Wireframe) View projects & Comment

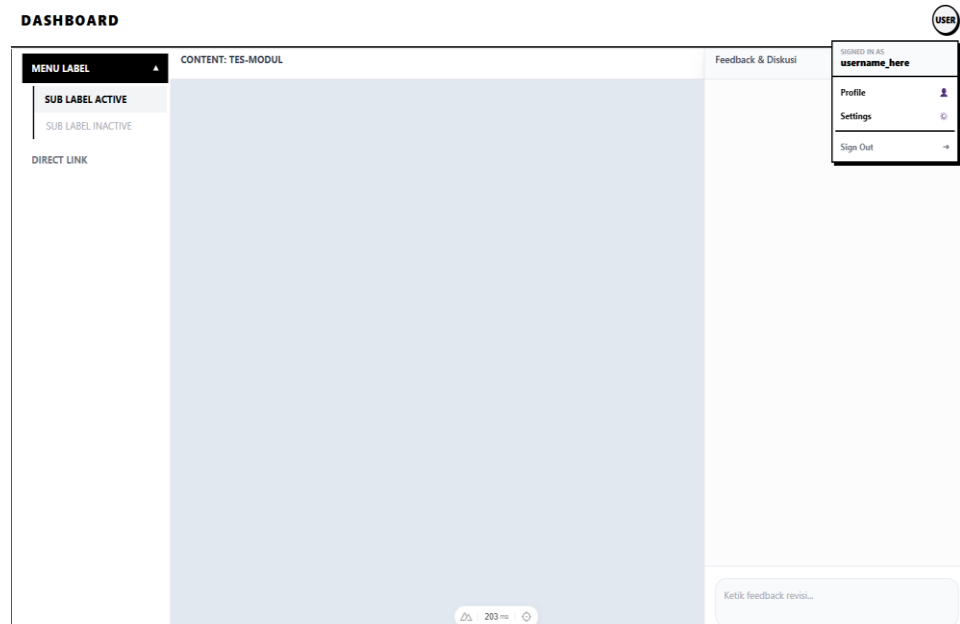
Bagian ini akan menampilkan *wireframe* tentang tahapan dari *view projects* yang menggunakan *iframe* dan sekaligus proses *comment*. Proses tersebut dari Gambar X ke Gambar Y dan kemudian ke Gambar Z. Gambar X adalah halaman *dashboard*, dan ketika aktor memilih *view projects* dia akan diarahkan ke Gambar Y, halaman ini akan menampilkan daftar modul yang terdapat di proyek yang telah dipilih. Di halaman tersebut ada *action preview* untuk menampilkan bagaimana aset digital modul tersebut yang akan diarahkan ke Gambar Z sekaligus di halaman tersebut para aktor dapat melakukan *comment*.



Gambar 26. Wireframe Dashboard



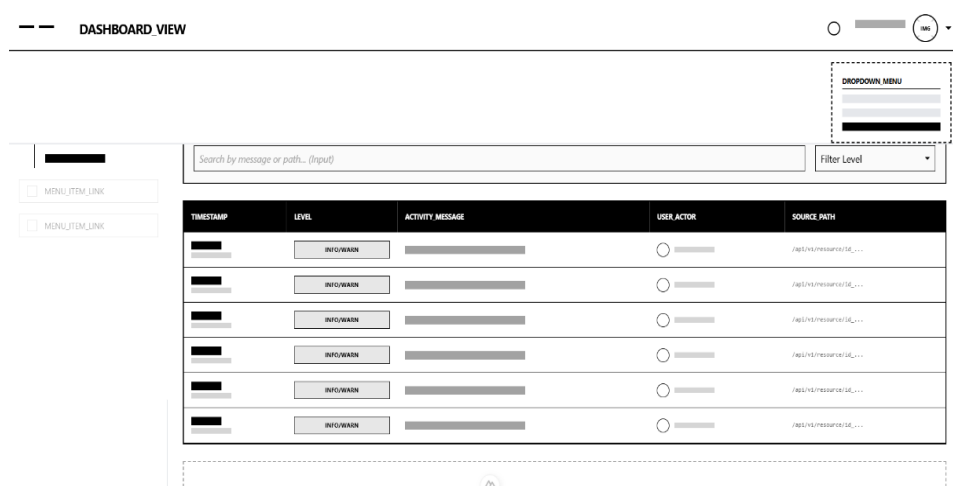
Gambar 27. Wireframe View Projects - Modules List



Gambar 28. Wireframe View Projects – Preview & Comment

G. Antarmuka (Wireframe) View Log Activity

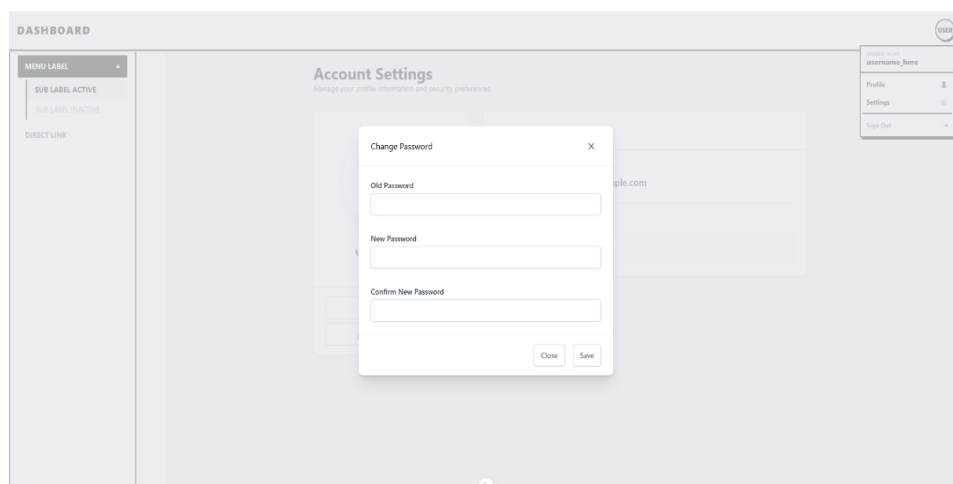
Bagian ini akan menampilkan *wireframe* mengenai log activity.



Gambar 29. Wireframe Log Activity

H. Antarmuka (Wireframe) Change Password

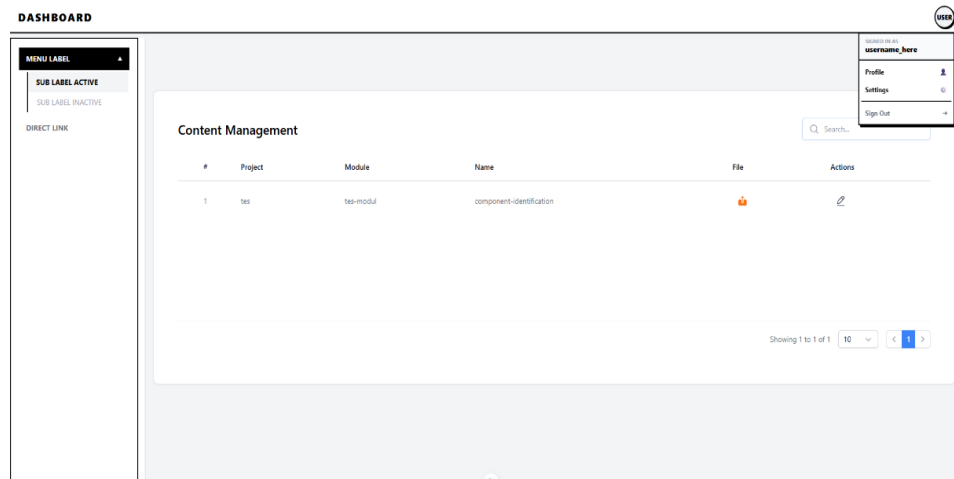
Bagian ini akan menampilkan *wireframe* bagaimana proses *change password*.



Gambar 30. Wireframe Change Password

I. Antarmuka (Wireframe) Upload Content

Bagian ini akan menampilkan *wireframe* bagaimana aktor melakukan *upload content*.



Gambar 31. Wireframe Upload Content

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini menyajikan hasil implementasi dari Repositori Konten Digital Berbasis Web yang telah dibahas pada bab sebelumnya. Proses implementasi ini mencakup transformasi desain perangkat lunak ke dalam kode program sehingga sistem dapat dijalankan pada infrastruktur peladen yang ditentukan. Selain memaparkan hasil antarmuka sistem, bab ini juga memuat hasil pengujian *User Acceptance Testing* (UAT) untuk memastikan bahwa setiap fitur yang dikembangkan telah memenuhi kebutuhan fungsional pengguna dan siap digunakan dalam lingkungan operasional PT Labtech.

4.1. Hasil Implementasi

Bagian ini menjelaskan hasil akhir dari pengembangan sistem Repositori Konten Digital Berbasis Web yang telah berhasil dibangun dan diimplementasikan pada infrastruktur PT Labtech.

4.1.1 Deskripsi Proses Implementasi

Implementasi sistem dilakukan dengan mengintegrasikan *framework Nuxt JS* pada sisi *frontend* dan *Node JS* pada sisi *backend*. Untuk penyimpanan data, digunakan basis data *MongoDB* yang dikonfigurasi pada *server* lokal PT Labtech. Proses penyebaran (*deployment*) dilakukan dengan langkah-langkah sebagai berikut :

1. Konfigurasi Sistem : Penulis menyiapkan lingkungan kerja di *server* dengan menginstall *Node JS* sebagai *runtime* utama. Untuk menjaga agar aplikasi tetap menyala 24 jam dan otomatis melakukan *restart* jika ada kendala, penulis memasang PM2 sebagai pengelola proses.
2. Konfigurasi Database : Penulis melakukan konfigurasi *environment variables* untuk menghubungkan sistem dengan basis data. Di tahap ini, keamanan menjadi prioritas dengan mengatur *connection string* yang terenkripsi, sehingga data modul dan konten tersimpan dengan aman di *server* Labtech.

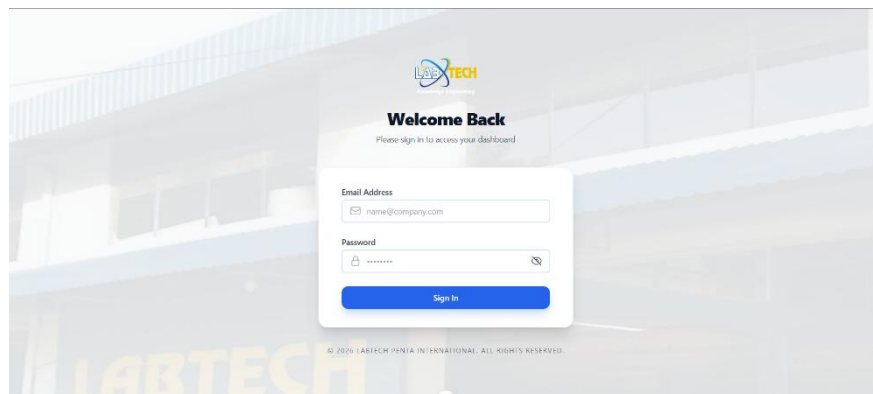
3. Build : Sebelum dijalankan, penulis melakukan proses *build* pada *framework Nuxt.js*. Tahapan ini bertujuan untuk mengompresi kode program agar lebih ringan dan cepat saat diakses oleh *staf internal* perusahaan melalui jaringan lokal.

4.1.2 Tampilan Antarmuka Sistem

Berikut adalah hasil dari antarmuka sistem yang telah di implementasikan:

A. Halaman *Login*

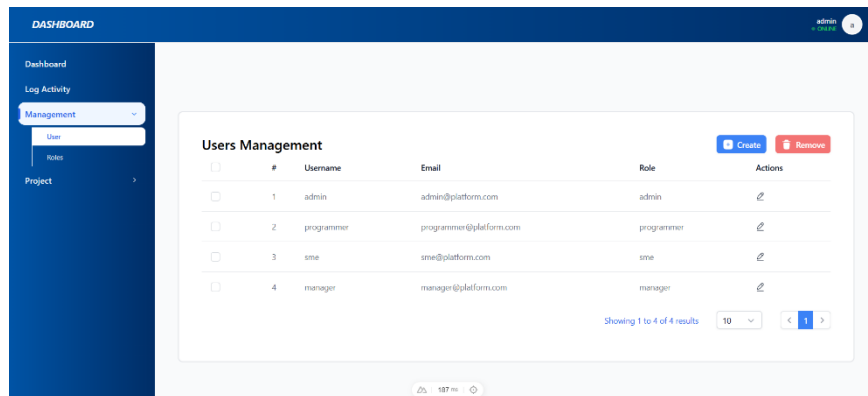
Halaman ini merupakan pintu masuk sebelum masuk ke dalam sistem, yang mana pengguna harus *login* menggunakan kredensial sah (email dan kata sandi) untuk masuk ke dalam sistem, Penulis juga melakukan validasi terhadap kredensial tersebut dan dapat mengenali peran setiap pengguna.



Gambar 32. Antarmuka - *Login*

B. Halaman Pengelolaan Pengguna (*CRUD Users*)

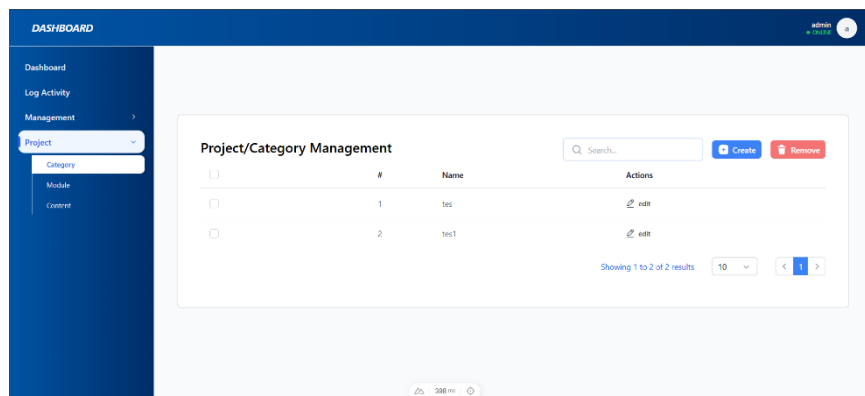
Rangkaian antarmuka pada Gambar 31 merupakan rangkaian yang bertujuan untuk melakukan pengelolaan data pengguna seperti menambah, melihat, menghapus atau mengubah.



Gambar 33. Antarmuka - *CRUD Users*

C. Halaman Pengelolaan kategori (*CRUD Category*)

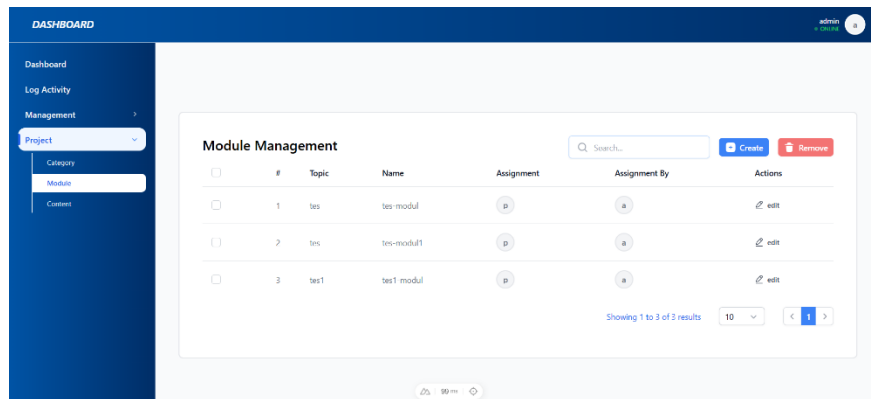
Rangkaian antarmuka pada Gambar 32 merupakan rangkaian yang bertujuan untuk melakukan pengelolaan data kategori seperti menambah, melihat, menghapus atau mengubah.



Gambar 34. Antarmuka - *CRUD Category*

D. Halaman Pengelolaan Modul (*CRUD Module*), *Sign Task*, dan *Upload Landing Page*

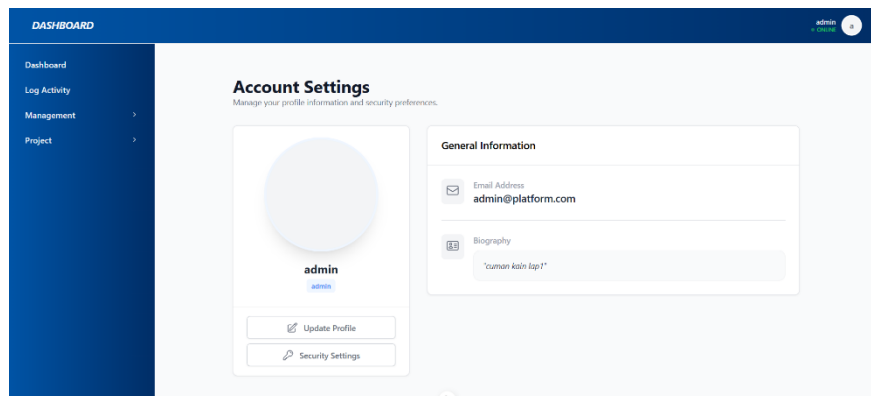
Rangkaian antarmuka pada Gambar 33 merupakan rangkaian yang bertujuan untuk melakukan pengelolaan data modul seperti menambah, melihat, menghapus atau mengubah. Akan tetapi, di rangkaian ini sudah termasuk dalam skenario *sign task dan upload landing page* saat melakukan *create module*.



Gambar 35. Antarmuka - *CRUD Module*

E. Halaman *Update Profile*

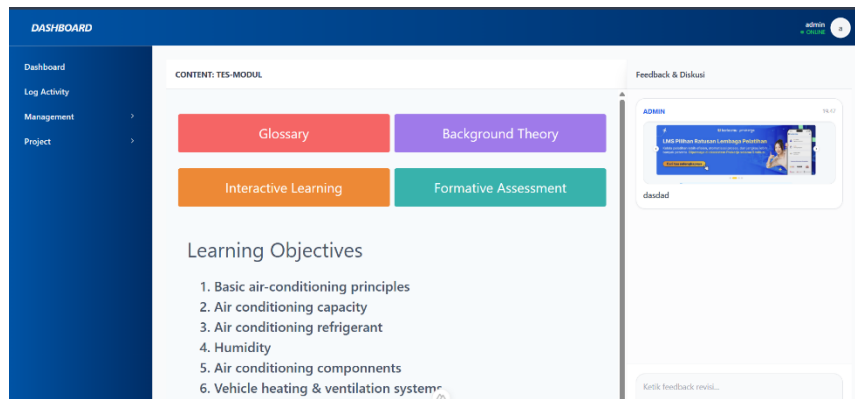
Rangkaian antarmuka pada Gambar 34 merupakan rangkaian yang bertujuan untuk melakukan perubahan pada profil pengguna seperti pembaharuan bio atau gambar.



Gambar 36. Antarmuka - *Update Profile*

F. Halaman *View Projects & comment*

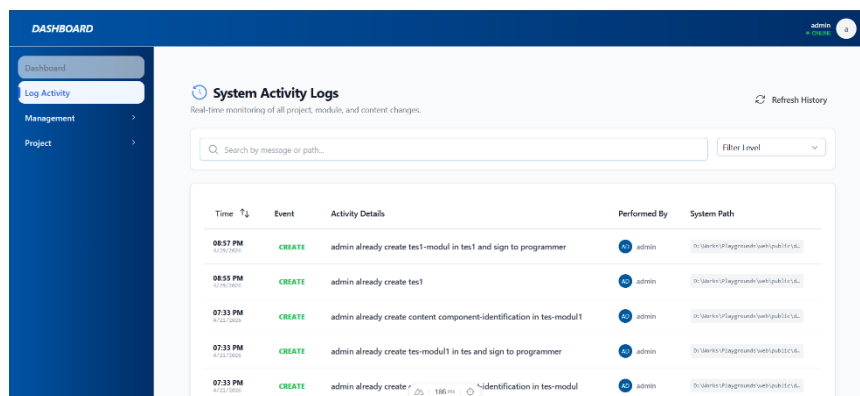
Rangkaian antarmuka pada Gambar 35 merupakan rangkaian yang berfungsi sebagai media untuk melihat atau memantau perkembangan modul serta terdapat fitur komentar. Rancangan fitur komentar ini bertujuan agar memungkinkan aktor dapat melakukan atau memberikan umpan balik terhadap modul tersebut.



Gambar 37. Antarmuka - View Projects & Comment

G. Halaman *View Log Activity*

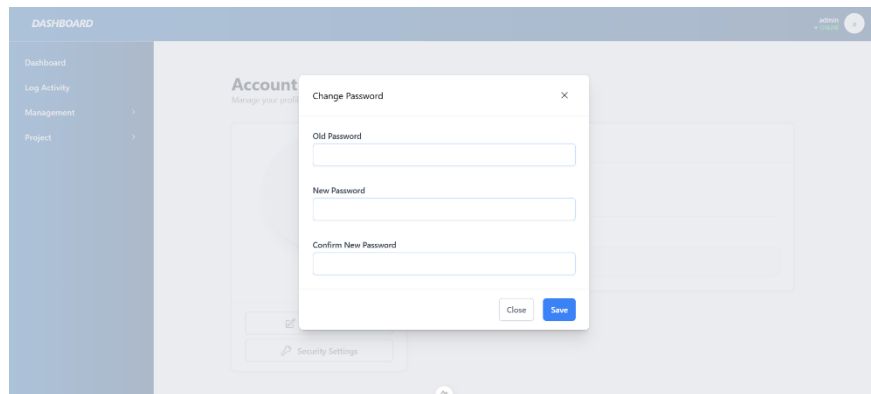
Aktivitas sistem diperlihatkan pada Gambar 36, halaman atau antarmuka ini bertujuan untuk menampilkan jejak sebuah aktivitas tertentu yang terjadi di sistem.



Gambar 38. Antarmuka - Log Activity

H. Halaman *Change Password*

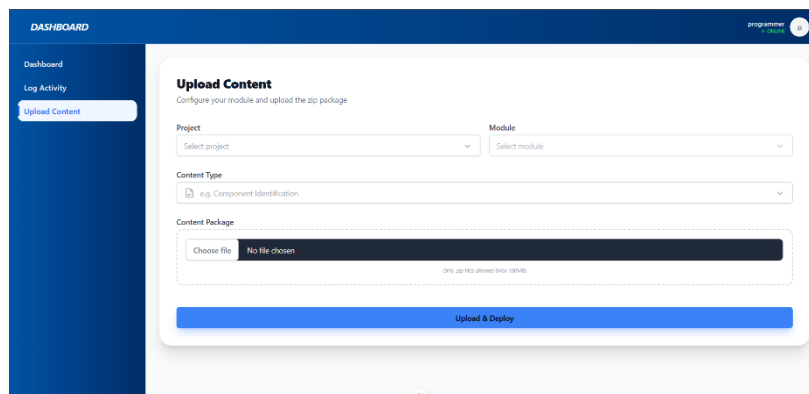
Pada Gambar 37, halaman ini bertujuan untuk demi keamanan, yang mana setiap aktor dapat melakukan perubahan pada kredensial keamanan mereka.



Gambar 39. Antarmuka - Change Password

I. Halaman Upload Content

Rangkaian antarmuka pada Gambar 38, rangkaian ini bertujuan untuk aktor (*programmer*) dapat melakukan *upload content* yang telah ditugaskan oleh aktor (*admin*) kepada aktor (*programmer*).



Gambar 40. Antarmuka - Upload Content

4.2. Pengujian

Metode pengujian sistem ini menggunakan metode pengujian blackbox, metode ini yang berfokus pada pengujian fungsionalitas seperti input dan output aplikasi. Berikut adalah hasil pengujian dalam pengembangan Rancang Bangun Repositori Konten Digital Berbasis Web sebagai *Platform Review* di PT Labtech Penta International.

Tabel 13. Pengujian Sistem

Kode	<i>Input</i>	<i>Output</i>	Keterangan	Penguji
FR-01	<i>Admin/Manager/Programmer dan SME input Email dan Password.</i>	<i>Admin/Manager/Programmer dan SME input dapat melakukan login</i>	Sukses	Yossie Dharma Putra (Leader SWD) dan Ferrry Sugianto (Manager Virtual Learning Department)
FR-02	<i>Admin/Manager mengisi form tambah/lihat/ubah/hapus data pengguna.</i>	<i>Admin/Manager dapat mengelola data pengguna seperti tambah/lihat/ubah/hapus</i>	Sukses	Yossie Dharma Putra (Leader SWD) dan Ferrry Sugianto (Manager Virtual Learning Department)
FR-03	<i>Admin melakukan tambah/lihat/ubah/hapus data kategori.</i>	<i>Admin dapat mengelola data kategori seperti tambah/lihat/ubah/hapus.</i>	Sukses	Yossie Dharma Putra (Leader SWD) dan Ferrry Sugianto (Manager Virtual Learning Department)
FR-04, FR-05, FR-06	<i>Admin melakukan pengelolaan data modul serta juga mengunggah file landing page serta penugasan.</i>	<i>Admin dapat mengelola data modul, mengunggah file landing</i>	Sukses	Yossie Dharma Putra (Leader SWD) dan Ferrry Sugianto (Manager

		<i>page</i> serta penugasan.		<i>Virtual Learning Department)</i>
FR-07	<i>Admin/Manager/Programmer/SME dapat melakukan logout dari sistem.</i>	<i>Admin/Manager/Programmer/SME dapat melakukan logout dari sistem</i>	Sukses	Yossie Dharma Putra (<i>Leader SWD</i>) dan Ferrry Sugianto (<i>Manager Virtual Learning Department</i>)
FR-08	<i>Admin/Manager/Programmer/SMEi dapat melakukan Update profile</i>	<i>Admin/Manager/Programmer/SME dapat melakukan update profile seperti mengunggah foto dan ubah bio.</i>	Sukses	Yossie Dharma Putra (<i>Leader SWD</i>) dan Ferrry Sugianto (<i>Manager Virtual Learning Department</i>)
FR-09	<i>Admin/Manager/Programmer/SMEi dapat melakukan Comment.</i>	<i>Admin/Manager/Programmer/SME dapat melakukan comment terhadap modul/konten.</i>	Sukses	Yossie Dharma Putra (<i>Leader SWD</i>) dan Ferrry Sugianto (<i>Manager Virtual Learning Department</i>)
FR-10	<i>Admin/Manager/Programmer/SMEi dapat melakukan View Projects.</i>	<i>Admin/Manager/Programmer/SME dapat melihat</i>	Sukses	Yossie Dharma Putra (<i>Leader SWD</i>) dan Ferrry Sugianto (<i>Manager</i>

		proyek yang ada di sistem.		<i>Virtual Learning Department)</i>
FR-11	<i>Admin/Manager/Programmer/SMEi dapat melakukan View Log Activity.</i>	<i>Admin/Manager/Programmer/SMEi dapat melihat aktivitas yang telah terjadi di sistem.</i>	Sukses	Yossie Dharma Putra (<i>Leader SWD</i>) dan Ferrry Sugianto (<i>Manager Virtual Learning Department</i>)
FR-12	<i>Admin/Manager/Programmer/SMEi dapat melakukan Change Password.</i>	<i>Admin/Manager/Programmer/SMEi dapat melakukan perubahan password.</i>	Sukses	Yossie Dharma Putra (<i>Leader SWD</i>) dan Ferrry Sugianto (<i>Manager Virtual Learning Department</i>)
FR-13	<i>Programmer dapat mengunggah konten yang ditugaskan.</i>	<i>Programmer bisa mengunggah konten yang telah ditugaskan kepadanya.</i>	Sukses	Yossie Dharma Putra (<i>Leader SWD</i>) dan Ferrry Sugianto (<i>Manager Virtual Learning Department</i>)

BAB V KESIMPULAN

5.1 Kesimpulan

Setelah melewati berbagai proses pengerjaan, saya dapat menyimpulkan beberapa hal penting sebagai berikut :

1. Aplikasi repositori ini telah berhasil membangun sebuah sistem yang berfungsi sebagai tempat penyimpanan aset digital (repositori) dan *platform review*.
2. Aplikasi repositori ini telah dibatasi dan diimplementasikan berdasarkan peran masing-masing pengguna.
3. Aplikasi repositori ini telah memenuhi seluruh kebutuhan fungsional yang telah ditetapkan dan sesuai dengan prosedur di PT Labtech.

5.2 Saran

Untuk pengembangan lebih lanjut agar sistem ini memberikan dampak yang lebih besar bagi operasional perusahaan, disarankan :

1. Menambahkan fitur *approval* pada sistem untuk meninjau aset digital.
2. Mengingat batasan saat ini yang hanya di lingkup internal, dimasa mendatang sistem dapat berjalan di lingkup eksternal.

DAFTAR PUSTAKA

- M.Z. Musoffa, E.S.Susanto, and Y.Mulyanto. (2022). Sistem Informasi Manajemen Aset Berbasis Web di Universitas Teknologi Sumbawa. Jurnal Informatika Teknologi dan Sains (JINTEKS), 4(01), 42.
- F.C.Wiryadi. (2023).Perbedaan Umpan Balik Langsung dan Umpan Balik Tertunda Dosen pada Keterampilan Resusitasi Bayi Baru Lahir. Jurnal Kesehatan Aeromedika (JKA), 9(01),12-15.
- T.A.Riza, M.Fitri Yani. (2026). Digitalisasi Manajemen Aset Desa Citeureup : Implementasi Aplikasi Untuk Pengelolaan Aset yang Transparan dan Efisien. Community Service Journal (NJCOM), 2(01),27.
- S.Syamsul, F.Yusuf, Rahmad, Adhy Rizaldy, A.M Abidin. (2025).Pengembangan Sistem Repositori Jurnal Menggunakan Metode Agile dan Framework Yii berbasis Web. Information System and Professional (INSYPRO), 10(02), 1.
- A.J.Zaidan, D.F.Suyatno .(2025).Rendering Performance Analysis of Astro JS, Next JS, Nuxt JS and SvelteKit Frameworks using Google Lighthouse, PageSpeed Insight, and Jmeter.Journal of Emerging Information System and Business Intelligence (JEISBI), 6(01), 13.
- M.F.Ahmod. (2023). Javascript Runtime Performance Analysis : Node and Bun. Tampere University, Tampere, Finland.
- Mutadin and Yulianti. (2023). Perancangan Sistem Informasi Pembuatan Akta Kelahiran Anak Berbasis Web dengan Metode Waterfall di Kecamatan Ciseeng Kabupaten Bogor. Journal of Research and Publication Innovation, 1(01), 31.

DAFTAR LAMPIRAN

Lampiran A. Link Produk

<https://github.com/wildy-simanjuntak/platform-review/>

Lampiran B. Link List Of Digital Content

<https://drive.google.com/file/d/1BTUkYJZZSzyrbqLRo6kNOs0WayAVjMt/view?usp=sharing>