

**RANCANG BANGUN SISTEM INFORMASI
ABSENSI DAN PELAPORAN KINERJA
KARYAWAN BERBASIS WEBSITE**

TUGAS AKHIR

Disusun Oleh:

Hugo Hamonangan

3312201066

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



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
2025**

KATA PENGANTAR

Segala puji dan rasa syukur penulis haturkan ke hadirat Allah Subhanahu wa Ta'ala atas rahmat, ilmu, kesehatan, dan kesempatan yang diberikan-Nya, sehingga penyusunan Tugas Akhir dengan judul “Rancang Bangun Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Website” dapat diselesaikan dengan baik.

Tugas Akhir ini disusun sebagai salah satu syarat kelulusan Program Diploma III Teknik Informatika. Selama pengerjaannya, penulis memperoleh dukungan, bimbingan, dan bantuan dari berbagai pihak. Untuk itu, dengan penuh kerendahan hati, penulis mengucapkan terima kasih sedalam-dalamnya kepada:

1. Allah Subhanahu wa Ta'ala atas rahmat, ilmu, kesehatan, dan kesempatan yang diberikan-Nya.
2. Kedua orang tua tercinta, yang senantiasa memberikan doa, serta motivasi tanpa henti selama masa studi dan penyusunan Tugas Akhir ini .
3. Bapak Swono Sibagariang, S.Kom., M.Kom., selaku dosen pembimbing, atas bimbingan, masukan, dan arahan yang diberikan selama penyusunan Tugas Akhir ini.
4. PT. Mitra Integritas Indonesia, atas kesempatan, fasilitas, serta dukungan yang diberikan selama pelaksanaan magang dan pengumpulan data untuk Tugas Akhir ini.
5. Teman-teman seperjuangan atas dukungan dan semangat yang senantiasa diberikan selama masa studi hingga penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih banyak kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk perbaikan karya ini di masa mendatang. Semoga laporan Tugas Akhir ini dapat memberikan manfaat serta menambah wawasan bagi para pembaca.

Batam, 17 Oktober 2025

Hugo Hamonangan

DAFTAR ISI

| | |
|---|-----------|
| KATA PENGANTAR..... | 2 |
| DAFTAR ISI..... | 3 |
| DAFTAR TABEL..... | 5 |
| DAFTAR GAMBAR..... | 7 |
| DAFTAR LAMPIRAN..... | 8 |
| ABSTRAK..... | 9 |
| 1.1. Latar Belakang..... | 10 |
| 1.2. Rumusan Masalah..... | 11 |
| 1.3. Tujuan..... | 11 |
| 1.4. Batasan Masalah..... | 11 |
| 1.5. Manfaat..... | 12 |
| A. Manfaat Bagi Perusahaan..... | 12 |
| B. Manfaat Bagi Mahasiswa..... | 12 |
| BAB II TINJAUAN PUSTAKA..... | 13 |
| 2.1. Penelitian Terkait..... | 13 |
| 2.2. Landasan Teori..... | 16 |
| 2.2.1. Tinjauan Teknologi (Bahasa Pemrograman, Framework dan <i>Library</i>)..... | 16 |
| 2.2.2. <i>Typescript</i> | 17 |
| 2.2.3. TailwindCSS..... | 17 |
| 2.2.4. Angular..... | 17 |
| 2.2.5. PrimeNG..... | 18 |
| 2.2.6. Nest JS..... | 18 |
| 2.2.7. Sequelize..... | 19 |
| 2.2.8. MySQL..... | 19 |
| 2.2.9. MySQL..... | 19 |
| 2.2.10. Visual Studio Code..... | 20 |
| 2.2.11. Figma..... | 20 |
| 2.2.12. Amazon Web Services..... | 20 |
| 2.3. Metode Pengembangan Produk..... | 21 |
| 2.3.1. Analisis Kebutuhan (<i>Requirement Analysis</i>)..... | 24 |
| 2.3.2. Desain Sistem (System Design)..... | 24 |
| 2.3.3. Implementasi (Implementation)..... | 24 |

| | |
|---|-----------|
| 2.3.4. Pengujian (Testing)..... | 25 |
| 2.3.5. Deployment..... | 25 |
| 2.3.6. Pemeliharaan (Maintanance)..... | 25 |
| BAB III ANALISIS DAN PERANCANGAN..... | 27 |
| 3.1. Analisis Kebutuhan..... | 27 |
| 3.1.1. Proses Bisnis Berjalan..... | 27 |
| 3.1.2. Gambaran Umum Sistem..... | 27 |
| 3.2. Perancangan..... | 28 |
| 3.2.1. Kebutuhan Fungsional..... | 28 |
| 3.2.2. Kebutuhan Non-Fungsional..... | 30 |
| 3.2.2. Diagram <i>Use Case</i> | 31 |
| 3.2.3. Skenario <i>Use Case</i> | 32 |
| 3.2.4. <i>Activity Diagram</i> | 41 |
| 3.3. Mekanisme Alur Kerja Pengguna..... | 54 |
| 3.3.1. Absensi..... | 54 |
| 3.3.2. Penugasan..... | 55 |
| BAB IV IMPLEMENTASI DAN PENGUJIAN | 56 |
| 4.1. Hasil Implementasi..... | 56 |
| 4.2. Pengujian <i>User Acceptance Testing</i> (UAT)..... | 57 |
| BAB V KESIMPULAN..... | 61 |
| 5.1 Kesimpulan..... | 61 |
| 5.2 Saran..... | 61 |
| DAFTAR PUSTAKA..... | 62 |
| DAFTAR LAMPIRAN..... | 64 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 1. Perbandingan Penelitian Sebelumnya..... | 14 |
| Tabel 2. Kebutuhan Fungsional..... | 28 |
| Tabel 3. Kebutuhan Non-Fungsional..... | 30 |
| Tabel 4. Skenario <i>Use Case</i> Manajer <i>Login</i> | 32 |
| Tabel 5. Skenario <i>Use Case</i> Manajer Proyek <i>Login</i> | 33 |
| Tabel 6. Skenario <i>Use Case</i> Karyawan dan Peserta Magang <i>Login</i> | 33 |
| Tabel 7. Skenario <i>Use Case</i> Manajer Penambahan Akun Pengguna..... | 34 |
| Tabel 8. Skenario <i>Use Case</i> Manajer Penghapusan Akun Pengguna..... | 35 |
| Tabel 9. Skenario <i>Use Case</i> Manajer Pengelompokan Divisi Pengguna..... | 35 |
| Tabel 10. Skenario <i>Use Case</i> Manajer Mengunduh Dokumen..... | 36 |
| Tabel 11. Skenario <i>Use Case</i> Manajer Melihat Data Riwayat Kehadiran..... | 36 |
| Tabel 12. Skenario <i>Use Case</i> Manajer Proyek Memberikan Tugas | 37 |
| Tabel 13. Skenario <i>Use Case</i> Manajer Proyek Memantau Tugas | 38 |
| Tabel 14. Skenario <i>Use Case</i> Karyawan dan Peserta Magang Presensi..... | 38 |
| Tabel 15. Skenario <i>Use Case</i> Manajer Proyek Komentar..... | 39 |
| Tabel 16. Skenario <i>Use Case</i> Karyawan dan Peserta Magang Komentar..... | 39 |
| Tabel 17. Skenario <i>Use Case</i> Manajer Proyek Menghapus Tugas | 39 |
| Tabel 18. Skenario <i>Use Case</i> Karyawan dan Peserta Magang Mengerjakan Tugas..... | 40 |
| Tabel 19. Skenario <i>Use Case</i> Manajer Proyek Melihat Notifikasi | 40 |
| Tabel 20. Hasil <i>Blackbox Testing</i> Manajer..... | 57 |
| Tabel 21. Hasil <i>Blackbox Testing</i> Manajer Proyek..... | 59 |
| Tabel 22. Hasil <i>Blackbox Testing</i> Karyawan dan Peserta Magang | 59 |

| | |
|--|-----------|
| Tabel 23. Tabel lampiran requirement..... | 64 |
| Tabel 24. Tabel lampiran pertanyaan..... | 64 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1. Metode Waterfall..... | 22 |
| Gambar 2. Penjelasan Alur Waterfall..... | 23 |
| Gambar 3. Gambaran Umum Sistem..... | 28 |
| Gambar 4. Gambar Diagram Use Case | 32 |
| Gambar 5. Gambar Activity Diagram Manager..... | 42 |
| Gambar 6. Gambar Activity Diagram Project Manager | 43 |
| Gambar 7. Gambar Activity Diagram Karyawan & Peserta Magang..... | 44 |
| Gambar 8. Gambar Diagram Use Case | 46 |
| Gambar 9. Gambar Diagram Use Case | 47 |
| Gambar 10. Notification Layout..... | 48 |
| Gambar 11. Confirmation Modal..... | 48 |
| Gambar 12. Message Toast..... | 49 |
| Gambar 13. Dashboard..... | 49 |
| Gambar 14. User Page..... | 50 |
| Gambar 15. Add User Modal..... | 50 |
| Gambar 16. User Detail Modal..... | 51 |
| Gambar 17. Division Page..... | 51 |
| Gambar 18. Attendance Page..... | 52 |
| Gambar 19. Assignment Page..... | 52 |
| Gambar 20. Add Assignment Modal..... | 53 |
| Gambar 21. Add Assignment Todo Modal..... | 53 |
| Gambar 22. Assignment Detail Modal..... | 54 |
| Gambar 24. Lampiran halaman Dashboard Manajer | 66 |
| Gambar 25. Lampiran halaman Dashboard Manajer Proyek..... | 67 |
| Gambar 27. Lampiran halaman Add assignment..... | 68 |
| Gambar 28. Lampiran halaman Add Project Groups..... | 68 |
| Gambar 29. Lampiran halaman Attendance..... | 68 |
| Gambar 30. Lampiran halaman Division | 69 |
| Gambar 31. Lampiran halaman User | 69 |
| Gambar 32. Lampiran halaman User Detail..... | 69 |

DAFTAR LAMPIRAN

| | |
|--|-----------|
| Lampiran A. Dokumen Proses Pengumpulan Requirement..... | 64 |
| Lampiran B. Link Produk..... | 65 |
| Lampiran C. Dokumen Pengujian UAT..... | 65 |

ABSTRAK

PT. Mitra Integritas Indonesia merupakan perusahaan pengembang perangkat lunak yang berlokasi di Batam, Kepulauan Riau, Indonesia. Berdiri sejak tahun 2017, perusahaan mulai membuka program magang pada tahun 2021. Seiring bertambahnya jumlah karyawan dan peserta magang, manajemen perlu melakukan evaluasi terhadap kehadiran serta laporan kinerja setiap akhir bulan. Saat ini, proses absensi dilakukan melalui Slack dengan cara mengunggah foto diri di tempat kerja ke saluran khusus, sedangkan laporan kerja disusun secara manual menggunakan Microsoft Word dan dikirim melalui Slack. Proses tersebut memerlukan waktu lama, berpotensi menimbulkan kesalahan, dan menyulitkan proses rekapitulasi data.

Untuk mengatasi permasalahan tersebut, peneliti merancang dan mengembangkan Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Website. Penelitian ini menggunakan metode wawancara untuk mengumpulkan kebutuhan sistem dari pihak manajemen dan pengguna, serta menerapkan model pengembangan sistem *waterfall* yang meliputi tahap analisis, perancangan, implementasi, pengujian, dan pemeliharaan. Sistem yang dibangun mengintegrasikan fitur absensi dan pelaporan dalam satu platform digital, sehingga memungkinkan akses data secara *real-time* serta mempermudah proses pengelolaan dan pemantauan oleh pihak manajemen. Aplikasi dikembangkan menggunakan Angular, Tailwind CSS, PrimeNG, NestJS, Sequelize, MySQL, Visual Studio Code, dan layanan Amazon Web Services S3 (AWS S3).

Kata Kunci: sistem absensi, pelaporan kinerja, aplikasi berbasis web, manajemen magang, metodologi Waterfall, angular, tailwindcss, nestjs, sequelize, mysql, vscode, aws

BAB I PENDAHULUAN

1.1. Latar Belakang

PT. Mitra Integritas Indonesia adalah perusahaan pengembang perangkat lunak yang berlokasi di Batam, Kepulauan Riau. Perusahaan ini telah beroperasi sejak tahun 2017 dan mulai membuka program magang pada tahun 2021. Seiring bertambahnya jumlah karyawan dan peserta magang, manajemen menghadapi tantangan dalam memantau kehadiran serta mengevaluasi kinerja.

Saat ini, proses pencatatan kehadiran dilakukan dengan cara mengunggah foto diri ke saluran khusus di *slack*. Sementara itu, laporan kerja dibuat secara manual menggunakan *Microsoft Word* dilengkapi tangkapan layar aktivitas lalu dikirim melalui *slack* ke saluran yang sama. Metode ini membutuhkan waktu lama untuk merekap data, rentan terjadi kesalahan atau kelalaian, dan format laporan sering tidak seragam. Akibatnya, manajemen kesulitan mendapatkan gambaran utuh tentang kinerja individu secara *real-time*.

Di sisi lain, peserta magang sering bingung tentang apa yang harus dilaporkan dan bagaimana cara melaporkannya, karena selama ini tidak terdapat panduan tertulis maupun alur baku segala instruksi mengenai pelaporan disampaikan secara lisan atau melalui pesan informal di *slack*. Tanpa platform terpusat, tidak ada rekam jejak digital yang jelas atas kontribusi setiap orang, sehingga sulit untuk memverifikasi aktivitas kerja atau membandingkan kinerja antar periode. Kondisi ini menunjukkan bahwa pendekatan berbasis komunikasi verbal dan dokumen terpisah tidak lagi memadai untuk mendukung kebutuhan manajemen kinerja yang transparan dan terdokumentasi.

Untuk mengatasi masalah tersebut, peneliti merancang dan mengembangkan Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Website. Sistem ini mengintegrasikan proses absensi dan pelaporan kinerja dalam satu platform digital terpusat, yang menyediakan antarmuka terstruktur bagi pengguna serta akses data berbasis periode waktu bagi manajemen. Dengan memanfaatkan teknologi seperti Angular, NestJS,

TypeScript, MySQL, dan penyimpanan cloud AWS S3, sistem ini tidak hanya menggantikan cara manual, tetapi juga memberikan akses data *real-time* bagi manajemen serta panduan jelas bagi pengguna dalam menjalankan tugas dan pelaporannya.

1.2. Rumusan Masalah

Berdasarkan uraian yang terdapat pada latar belakang, maka dapat diidentifikasi beberapa permasalahan yang dihadapi oleh PT. Mitra Integritas Indonesia, yaitu:

1. Bagaimana aplikasi *Rancang Bangun Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Website* dapat mengintegrasikan pencatatan kehadiran serta pelaporan kerja yang saat ini masih dilakukan melalui *Slack* dan secara manual menggunakan *Microsoft Word* ?
2. Bagaimana aplikasi *Rancang Rancang Bangun Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Website* memastikan data kehadiran dan laporan kerja terdokumentasi secara sistematis dan dapat diakses kembali dengan mudah berdasarkan periode waktu tertentu ?

1.3. Tujuan

1. Membuat aplikasi berbasis web yang mengintegrasikan mekanisme pencatatan kehadiran dan pelaporan kinerja dalam satu antarmuka, menggantikan penggunaan *Slack* dan *Microsoft Word* yang terpisah dan manual.
2. Merancang dan menerapkan struktur basis data serta antarmuka pengguna yang memungkinkan dokumentasi otomatis data kehadiran dan laporan kerja, serta pengambilan data berdasarkan filter periode waktu tertentu.

1.4. Batasan Masalah

Dalam pembangunan aplikasi ini, terdapat batasan-batasan masalah seperti berikut:

1. Aplikasi ini hanya diperuntukkan dan hanya bisa diakses oleh PT. Mitra Integritas Indonesia.
2. Aplikasi ini berfokus pada manajemen absensi, tugas dan pelaporan kerja yang berkaitan dengan kegiatan internal perusahaan.
3. Aplikasi ini tidak dapat diakses diluar jam kerja kecuali kepada *role* yang diberikan akses.

1.5. Manfaat

Dalam melakukan penelitian ini, maka ada beberapa manfaat yang diperoleh sebagai berikut:

A. Manfaat Bagi Perusahaan

1. Membantu melakukan pelaporan kerja yang lebih cepat.
2. Menghilangkan kesalahan yang mungkin terjadi pada saat menggunakan proses manual.
3. Menjadikan dokumen dan data yang sebelumnya diproses secara terpisah terintegrasi dalam satu sistem.
4. Memudahkan penelusuran terkait data dan dokumen yang dibutuhkan

B. Manfaat Bagi Mahasiswa

1. Mampu membangun dan menyelesaikan aplikasi sesuai yang dibutuhkan perusahaan.
2. Mendapat pengetahuan dan pembelajaran mengenai praktik pembuatan aplikasi dari awal hingga akhir dengan studi kasus yang nyata.

BAB II TINJAUAN PUSTAKA

2.1. Penelitian Terkait

1. Penelitian oleh Gabriela Maria Wowor (2021) mengembangkan aplikasi pelaporan kinerja berbasis web untuk Dinas Pendidikan dan Kebudayaan Kota Tomohon guna mengatasi keterbatasan sistem lama yang hanya dapat digunakan oleh ASN. Sistem baru ini memungkinkan seluruh pegawai (termasuk tenaga kontrak) untuk melaporkan kegiatan harian yang kemudian direkap secara otomatis per bulan, serta divalidasi oleh atasan terkait. Meskipun sistem ini berhasil menyatukan seluruh pegawai dalam satu platform, penelitian ini tidak menyertakan fitur absensi atau verifikasi kehadiran, sehingga belum mampu memastikan bahwa laporan kinerja dibuat pada waktu dan tempat yang sesuai dengan ketentuan kerja.
2. Penelitian yang dilakukan oleh Muhammad Ervanza Devo Valerian & Dahlia (2024) menganalisis dampak penerapan sistem absensi berbasis website terhadap kinerja dan efisiensi waktu pegawai di DPRD Provinsi DKI Jakarta. Penelitian ini menggunakan pendekatan kuantitatif dengan analisis statistik (regresi linear sederhana, uji-t, R^2) untuk membuktikan bahwa sistem absensi digital memiliki pengaruh signifikan terhadap peningkatan kinerja, meskipun masih memiliki celah keamanan seperti potensi kecurangan absensi jika tidak dibatasi aksesnya.
3. Penelitian oleh Rika Rahmawati (2018) mengembangkan sistem pelaporan kinerja pegawai berbasis web di Kementerian Agama Kabupaten Luwu Utara untuk menggantikan sistem manual berbasis kertas. Sistem ini memungkinkan pegawai menginput laporan harian secara digital dan atasan menyetujui laporan tersebut melalui antarmuka terpisah. Meskipun sistem terbukti berfungsi dengan baik berdasarkan pengujian black-box, penelitian ini tidak menyertakan fitur absensi atau validasi kehadiran, sehingga belum mampu memastikan bahwa laporan kinerja dibuat pada waktu dan tempat yang sesuai dengan ketentuan kerja.

4. Penelitian oleh Pramitha Dwi Larasati dan Nur Fajri Sa'ba (2021) merancang sistem pelaporan absensi berbasis web untuk PT. Solar Control Specialist guna mengatasi keterbatasan sistem fingerprint yang tidak praktis bagi tenaga lapangan seperti teknisi, kurir, dan tim klaim. Sistem yang diusulkan memungkinkan pegawai melaporkan kehadiran secara online dari lokasi kerja menggunakan perangkat mobile, serta mencakup pengajuan cuti, izin, dan laporan dinas dalam satu platform.

Tabel 1. Perbandingan Penelitian Sebelumnya

| Peneliti/Tahun | Judul/Topik Penelitian | Metode/Teknologi | Kelebihan | Kekurangan/Celah |
|--|--|--|---|--|
| Gabriela Maria Wowor (2021) | Aplikasi Pelaporan Kinerja berbasis Web (studi kasus: Dinas Pendidikan dan Kebudayaan Daerah Kota Tomohon) | Waterfall, berbasis web dengan PHP dan MYSQL | Mengintegrasikan seluruh pegawai (ASN dan tenaga kontrak) dalam satu sistem pelaporan, mengatasi keterbatasan sistem lama yang hanya untuk ASN. | Tidak memiliki notifikasi otomatis untuk pengingat pelaporan, antarmukanya belum responsif untuk perangkat mobile, dan tidak menyediakan audit log untuk melacak perubahan data. |
| Muhammad Ervanza Devo Valerian dan Dahlia (2024) | Analisa Penerapan Sistem Absensi Pegawai Berbasis Website Pada DPRD Provinsi DKI Jakarta | React Native, Firebase | Membuktikan secara statistik bahwa sistem absensi berbasis web meningkatkan kinerja sebesar (69% variasi dijelaskan). Sistem memudahkan | Hanya mengandalkan jaringan LAN internal tanpa solusi teknis bawaan sistem. kuantitatif). |

| | | | | |
|--|--|--|--|--|
| | | | pencatatan real-time dan manajemen data. | |
| Rika Rahmawati (2018) | Sistem Pelaporan Kinerja Pegawai Berbasis Web Pada Kementerian Agama Kabupaten Luwu Utara. | PHP Triad (Xampp, PHP, dan My SQL), <i>waterfall</i> | Menggantikan sistem pelaporan manual (kertas) yang boros dan rentan hilang dengan sistem digital yang terstruktur. | Sistem ini hanya berfokus pada pelaporan kinerja tanpa integrasi absensi atau verifikasi kehadiran, tidak memiliki mekanisme anti-kecurangan (seperti validasi waktu, lokasi, atau foto), tidak dilengkapi notifikasi otomatis, antarmukanya belum responsif untuk mobile, dan tidak menyediakan audit log untuk melacak perubahan data. |
| Pramitha Dwi Larasati dan Nur Fajri Sa'ba (2021) | Perancangan Sistem Pelaporan Absensi Berbasis Web pada PT. Solar Control Specialist | PHP Triad (Xampp, PHP, dan My SQL), <i>waterfall</i> | Mengatasi keterbatasan sistem absensi fingerprint yang tidak praktis bagi teknisi dan tenaga lapangan yang | Tidak menyebutkan notifikasi otomatis atau rekap laporan real-time, yang penting untuk manajemen kinerja harian. |

| | | | | |
|--|--|--|-------------------------|--|
| | | | bekerja di luar kantor. | |
|--|--|--|-------------------------|--|

Berdasarkan kajian terhadap penelitian terdahulu, topik sistem informasi absensi dan pelaporan kinerja karyawan telah dikembangkan dalam berbagai konteks organisasi. Namun, sebagian besar penelitian tersebut memperlakukan absensi dan pelaporan kinerja sebagai dua proses terpisah, atau hanya mengembangkan salah satunya. Penelitian Gabriela Maria Wowor (2021) dan Rika Rahmawati (2018) mengembangkan sistem pelaporan kinerja digital, tetapi tidak menyertakan mekanisme absensi atau validasi kehadiran. Sebaliknya, penelitian Muhammad Ervanza dkk. (2024) dan Pramitha Dwi Larasati dkk. (2021) fokus pada sistem absensi, namun tidak mengintegrasikannya dengan pelaporan kinerja harian secara teknis maupun fungsional.

Penelitian ini berbeda dari penelitian-penelitian sebelumnya melalui pengintegrasian fitur timelog sebagai bagian wajib dalam proses pelaporan tugas. Timelog berfungsi sebagai mekanisme pencatatan otomatis yang merekam waktu mulai dan selesai pengerjaan suatu tugas secara langsung dalam sistem. Data timelog dikaitkan secara eksplisit dengan setiap entri laporan kinerja dan tidak dapat dimanipulasi oleh pengguna setelah sesi pengerjaan berakhir. Dengan demikian, laporan kinerja dalam sistem ini tidak hanya berisi deskripsi aktivitas, tetapi juga mencakup rekaman temporal objektif yang dihasilkan selama proses pengerjaan tugas tersebut.

2.2. Landasan Teori

2.2.1. Tinjauan Teknologi (Bahasa Pemrograman, Framework dan *Library*)

Menurut Ahmad Taufik et. al (2022:1) Teknologi merupakan sebuah perkembangan perangkat keras (hardware) maupun perangkat lunak (software) yang didasari ilmu pengetahuan dengan seiring

perkembangan zaman dan didasari kebutuhan pengguna saat ini. Berdasarkan pengertian tersebut, maka teknologi harus selalu beradaptasi dengan perkembangan zaman dan memenuhi kebutuhan pengguna. Dalam ukuran memenuhi persyaratan teknologi terkini. Maka PT. Mitra Integritas Indonesia sudah menerapkan teknologi terkini untuk mendukung pengembangan aplikasi yang sesuai dengan kebutuhan saat ini. Oleh karena itu, peneliti akan mengikuti apa yang sudah diterapkan perusahaan tersebut. Teknologi yang digunakan untuk membuat aplikasi ini yaitu, *Typescript, Tailwind CSS, Angular, PrimeNg, Nest Js, Sequelize, MySQL, Visual Studio Code, Figma* dan *Amazon Web Service S3 (AWS S3)*.

2.2.2. Typescript

Menurut Boris Cherny (2019:5-6) TypeScript adalah superset dari JavaScript yang menambahkan fitur-fitur seperti *static typing* dan *type annotations*, yang membantu pengembang menulis kode yang lebih terstruktur, aman, dan mudah dikelola, terutama dalam aplikasi berskala besar. Typescript menghadirkan berbagai peningkatan performa dan fitur yang lebih canggih, seperti peningkatan dukungan terhadap *ECMAScript*, peningkatan inferensi tipe, dan perbaikan *bug* yang memastikan stabilitas serta pengalaman pengembangan yang lebih baik.

2.2.3. TailwindCSS

Tailwind CSS merupakan salah satu framework *Cascading Style Sheets (CSS)* yang digunakan untuk melakukan perancangan dan desain pada user interface (UI) dengan waktu yang relatif cepat. *Tailwind CSS* bersifat *utility first* yang artinya dapat menyesuaikan kebutuhan kita sebagai developer dalam mengkonfigurasi dan mengatur tampilan *UI* seperti ukuran *border, warna, font, shadows, breakpoints*, dan sebagainya.

2.2.4. Angular

Menurut Aleksis Kujala (2023:21-22) *Angular* adalah *framework open-source* yang dikembangkan oleh Google dengan bahasa pemrograman *TypeScript* untuk memudahkan pembuatan aplikasi web *single-page* yang interaktif dan responsif. Framework ini pertama kali diperkenalkan pada tahun 2010 dengan nama *AngularJS* (versi 1.x), yang langsung menarik perhatian komunitas pengembang web karena kemampuannya menyederhanakan proses pengembangan melalui fitur-fitur seperti data binding dua arah dan arsitektur *MVC (Model-View-Controller)*. Pada tahun 2016, *Angular* mengalami perubahan signifikan dengan peluncuran *Angular 2*, di mana framework ini dikembangkan ulang dari dasar dengan menggunakan *TypeScript* sebagai bahasa utama dan pendekatan arsitektur yang lebih modern.

2.2.5. PrimeNG

Menurut Sudheer Jonna dan Oleg Varaksin (2017:43) *PrimeNG* adalah sebuah *open-source library* yang kaya akan komponen antarmuka pengguna (UI) untuk *Angular 2+*. *PrimeNG* berasal dari *PrimeFaces* - suite komponen *JavaServer Faces (JSF)* yang paling populer. Saat ini, *PrimeNG* memiliki lebih dari 80 widget yang mudah digunakan. *Widget-widget* ini terbagi ke dalam beberapa kelompok, seperti komponen *input* dan *select*, *panel*, *overlay*, *menu*, *chart*, *message*, *multimedia*, *drag-and-drop*, dan berbagai komponen lainnya.

2.2.6. Nest JS

Menurut Pham, Anh Duc (2020: 12-13) *NestJS* adalah sebuah *framework* untuk membangun *backend* yang kuat dan *scalable* menggunakan *Node.js* dan *TypeScript* yang dikembangkan oleh Kamil Myśliwiec. Framework ini mendukung bahasa *JavaScript* dan *TypeScript*. Selain itu, *NestJS* menggabungkan komponen dari *FP (Pemrograman*

Fungsional), OOP (Pemrograman Berorientasi Objek), dan FRP (Pemrograman Reaktif Fungsional).

2.2.7. Sequelize

Sequelize versi 6 adalah *Object Relational Mapping Tools* yang mendukung berbagai database seperti Postgres, MySQL, MariaDB, SQLite, Microsoft SQL Server dsb. Fitur-fitur yang tersedia pada *sequelize* diantaranya adalah *transaction, relations, eager, lazy loading* dsb. Dengan *sequelize*, pengembangan dapat melakukan operasi database seperti CRUD tanpa harus menulis query SQL secara langsung. *Sequelize* pertama kali dikembangkan oleh Sascha Depol, seorang pengembang perangkat lunak. Tujuan diciptakan *sequelize* adalah agar memberikan kemudahan berinteraksi dengan database dalam aplikasi Node.js.

2.2.8. MySQL

Menurut Russell J.T Dyer (2015:19) *MySQL* adalah sistem manajemen basis data relasional (RDBMS) yang bersifat *open-source, multithreaded*, dan diciptakan oleh Michael Monty Widenius pada tahun 1995, digunakan untuk menyimpan dan mengelola data aplikasi. *MySQL* mendukung berbagai fitur canggih, termasuk replikasi data, transaksi ACID, dan skalabilitas tinggi, yang membuatnya menjadi pilihan utama dalam pengembangan aplikasi web dan enterprise.

2.2.9. MySQL

Menurut Russell J.T Dyer (2015:19) *MySQL* adalah sistem manajemen basis data relasional (RDBMS) yang bersifat *open-source, multithreaded*, dan diciptakan oleh Michael Monty Widenius pada tahun 1995, digunakan untuk menyimpan dan mengelola data aplikasi. *MySQL* mendukung berbagai fitur canggih, termasuk replikasi data, transaksi

ACID, dan skalabilitas tinggi, yang membuatnya menjadi pilihan utama dalam pengembangan aplikasi web dan enterprise.

2.2.10. Visual Studio Code

Menurut Alessandro Del Sole (2019:1-2) Visual Studio Code adalah sebuah editor kode yang dikembangkan oleh Microsoft. Dikenal sebagai alat pengembangan perangkat lunak yang ringan namun kaya fitur. Seiring waktu, *VSCode* mengalami perkembangan pesat dan menjadi salah satu editor kode yang paling banyak digunakan oleh pengembang perangkat lunak di seluruh dunia, berkat kemudahan penggunaannya dan kemampuannya untuk berintegrasi dengan alat-alat pengembangan modern.

2.2.11. Figma

Figma adalah *tools* pendukung *User Interface* (UI) dan *User Experience* (UX) yang berbasis website yang digunakan untuk mendesain sebuah aplikasi. *Figma* pertama kali didirikan pada tahun 2012 yang berbasis cloud. *Figma* dapat diakses dimana saja seperti *desktop* atau *laptop* yang terhubung dengan jaringan internet. Kelebihan *Figma* adalah memberikan kemungkinan bagi penggunanya untuk mendesain *interface* dari sebuah aplikasi secara kolaboratif dan *real-time*.

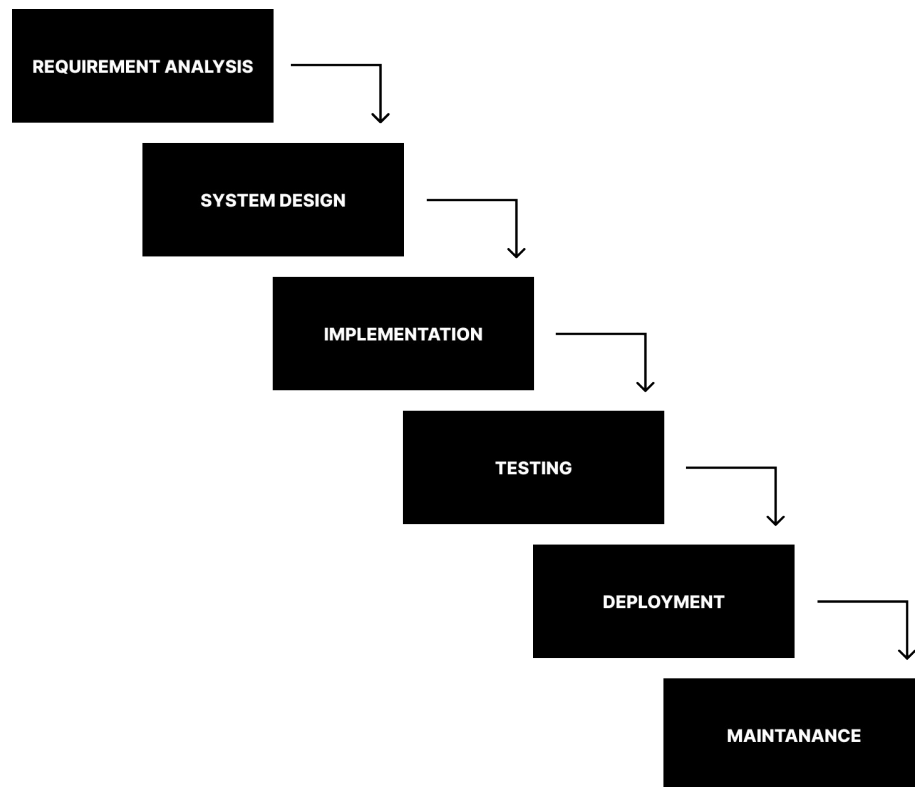
2.2.12. Amazon Web Services

Amazon Web Services (*AWS*) *S3* pertama kali diluncurkan pada tahun 2006 oleh Amazon.com sebagai bagian dari upaya mereka untuk menyediakan layanan cloud yang dapat diakses oleh pengembang dan perusahaan di seluruh dunia. Seiring berjalannya waktu, *S3* berkembang menjadi salah satu layanan penyimpanan cloud paling populer dan digunakan oleh banyak perusahaan besar dan kecil di berbagai industri.

2.3. Metode Pengembangan Produk

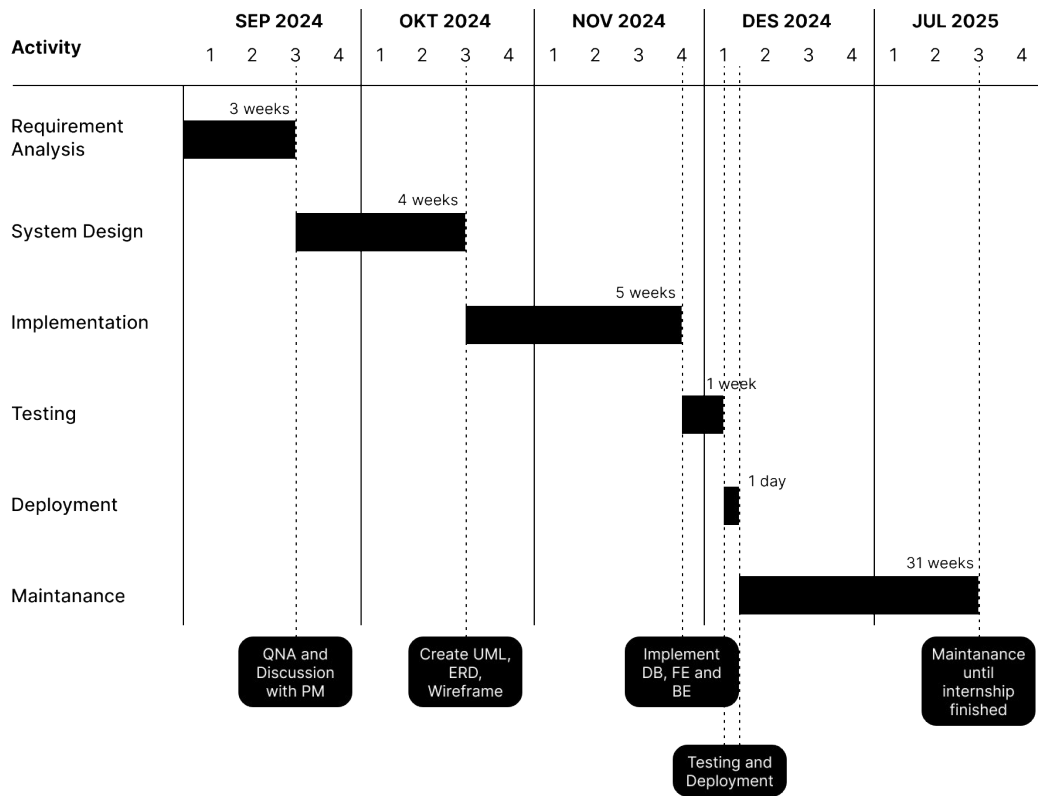
Metodologi pengembangan sistem menurut Joseph Valacich dan Joey F. George (2020:30), adalah proses pekerjaan dalam perusahaan perangkat lunak untuk membuat sistem informasi dengan melewati langkah-langkah fundamental. Langkah-langkah mendasar dalam proses pengembangan sistem yang meliputi perencanaan, analisis, perancangan, implementasi, dan pemeliharaan. Setiap langkah ini saling berhubungan dan membentuk sebuah siklus yang dikenal sebagai *System Development Life Cycle (SDLC)*. *SDLC* merupakan sebuah metodologi dalam pengembangan sistem yang terstruktur, dimana setiap tahapan pengembangan saling terkait untuk menciptakan alur kerja yang jelas. Setiap tahapan dalam siklus ini menggambarkan proses yang harus dilalui agar sistem yang dibangun dapat berfungsi dengan baik sesuai dengan kebutuhan pengguna. Selain itu, *SDLC* juga memberikan gambaran tentang bagaimana sistem dirancang, diimplementasikan, dan dikelola secara berkelanjutan agar tetap relevan dengan perkembangan teknologi dan kebutuhan bisnis. Berbagai model siklus pengembangan sistem yang banyak digunakan adalah *Waterfall*, *Agile*, dan *Spiral*.

Dalam penelitian ini, peneliti akan menggunakan metode *Waterfall*. Menurut Indra Rianto (2021:44), metode *Waterfall* merupakan model pengembangan sistem yang mengikuti pendekatan linier atau berurutan, di mana setiap fase dalam proses pengembangan harus diselesaikan secara tuntas sebelum melanjutkan ke fase berikutnya. Pendekatan ini tidak memperbolehkan adanya tumpang tindih antar fase, sehingga menghasilkan proses yang terstruktur dan sistematis. Giandari Maulani et al (2024:7) menjelaskan bahwa *Waterfall* menerapkan langkah-langkah berurutan yang dimulai dari *Requirement Analysis*, *System Design*, *Implementation*, *Testing*, *Deployment* dan *Maintenance*.



Gambar 1. Metode Waterfall

Berikutnya disajikan gambaran visual yang menggambarkan secara menyeluruh tahapan-tahapan pengembangan aplikasi, mulai dari proses awal perencanaan hingga pelaksanaan implementasi, pengujian, dan penyerahan sistem. Diagram ini bertujuan untuk memberikan gambaran kronologis dan struktural terhadap alur kerja yang dilakukan selama proses pembangunan sistem, sehingga memungkinkan pemahaman yang lebih jelas mengenai urutan aktivitas, keterkaitan antar tahap, serta tanggung jawab yang diemban pada setiap fase pengembangan.



Gambar 2. Penjelasan Alur Waterfall

Pengembangan sistem ini mengikuti alur kerja terstruktur yang terbagi dalam lima tahap utama. Tahap pertama adalah QNA and Discussion with PM, di mana dilakukan wawancara dan diskusi dengan Project Manager untuk memahami kebutuhan fungsional, batasan, serta ekspektasi sistem dari sisi bisnis. Tahap kedua, Create UML, ERD, Wireframe, melibatkan pembuatan dokumen desain sistem. Di tahap ini, dibuat diagram UML (Use Case, Activity, Sequence), model entitas-relasi (ERD) untuk struktur basis data, serta wireframe antarmuka pengguna sebagai representasi visual tampilan sistem sebelum implementasi. Tahap ketiga, Implement DB, FE and BE, merupakan proses pengembangan teknis. Basis data dirancang dan diimplementasikan menggunakan MySQL, sementara bagian front-end dikembangkan dengan Angular dan Tailwind CSS, dan back-end menggunakan NestJS. Integrasi antar lapisan dilakukan secara bertahap. Setelah tahap implementasi selesai, dilanjutkan dengan Testing and

Deployment. Pada tahap ini, sistem diuji secara fungsional dan non-fungsional, kemudian dideploy ke lingkungan produksi menggunakan layanan cloud AWS S3 untuk penyimpanan statis dan server backend. Tahap terakhir adalah Maintenance until internship finished, di mana sistem dipantau dan diperbaiki jika terdapat bug atau perubahan kebutuhan selama masa magang berlangsung, hingga akhir periode magang.

2.3.1. Analisis Kebutuhan (*Requirement Analysis*)

Analisis kebutuhan dalam metode Waterfall merupakan tahap awal yang sangat krusial dalam proses pengembangan sistem. Tahap ini bertujuan untuk memahami dan mendefinisikan secara mendalam kebutuhan serta permasalahan yang dihadapi oleh pengguna atau organisasi. Proses ini berfokus pada pengumpulan, identifikasi, dan dokumentasi kebutuhan sistem secara rinci untuk memastikan bahwa pengembang memiliki pemahaman yang jelas, terstruktur, dan lengkap mengenai tujuan serta fungsi yang harus dicapai oleh sistem yang akan dibangun. Dalam penelitian ini, peneliti mengumpulkan data melalui metode wawancara dan diskusi dengan manajer proyek untuk mendapatkan pemahaman yang mendalam mengenai kebutuhan sistem yang akan dibuat. Kemudian, peneliti menyusun functional requirements (FR), non-functional requirements (NFR), use case dan activity diagram.

2.3.2. Desain Sistem (*System Design*)

Desain sistem merupakan tahap yang dilakukan setelah analisis, tahap ini berfokus pada perencanaan yang terstruktur mengenai bagaimana sistem atau perangkat lunak akan dibangun berdasarkan spesifikasi yang telah ditetapkan pada analisis. Proses ini meliputi *Entity Relationship Diagram (ERD)*, pembuatan *User Interface (UI)* dan pertimbangan *User Experience (UX)*.

2.3.3. Implementasi (*Implementation*)

Implementasi merupakan tahap pengkodean dan penerapan sistem dengan berbagai *tools* dan bahasa pemrograman sesuai dengan kebutuhan tim dan perusahaan. dan sistem yang akan dibangun menggunakan beberapa teknologi terkini diantaranya, *Angular* sebagai framework *frontend* yang berbasis bahasa pemrograman *TypeScript* , *NestJS* yang berbasis bahasa pemrograman *TypeScript*, *Tailwind CSS* sebagai framework *styling* antarmuka, *MySQL* sebagai database, *AWS S3* sebagai tempat *cloud storage* untuk menyimpan berbagai file berupa dokumen dan gambar, *Figma* untuk merancang desain antarmuka dan yang terakhir *Visual Studio Code* sebagai kode editor.

2.3.4. Pengujian (Testing)

Tahap testing merupakan fase setelah tahap implementasi dalam metodologi Waterfall yang bertujuan untuk memverifikasi dan memvalidasi bahwa sistem yang telah diimplementasikan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan. Pada tahap ini, proses pengujian akan dilakukan dengan metode black-box testing. Fokus metode ini adalah pada validasi fungsionalitas sistem dari perspektif pengguna akhir.

2.3.5. Deployment

Deployment adalah proses publikasi perangkat lunak atau aplikasi ke lingkungan produksi atau server sehingga dapat diakses dan digunakan oleh pengguna akhir. Dalam penelitian ini, penyedia server yang akan digunakan yaitu *AWS* sebagai standarisasi yang sudah ditetapkan perusahaan.

2.3.6. Pemeliharaan (Maintenance)

Pemeliharaan adalah perbaikan *bug* atau kesalahan yang ditemukan oleh pengguna setelah sistem di terapkan ke lingkungan produksi. Selain

itu, tahap ini juga mencakup pembaharuan dan peningkatan fitur yang menyesuaikan terhadap perubahan kebutuhan pengguna atau teknologi.

BAB III ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan

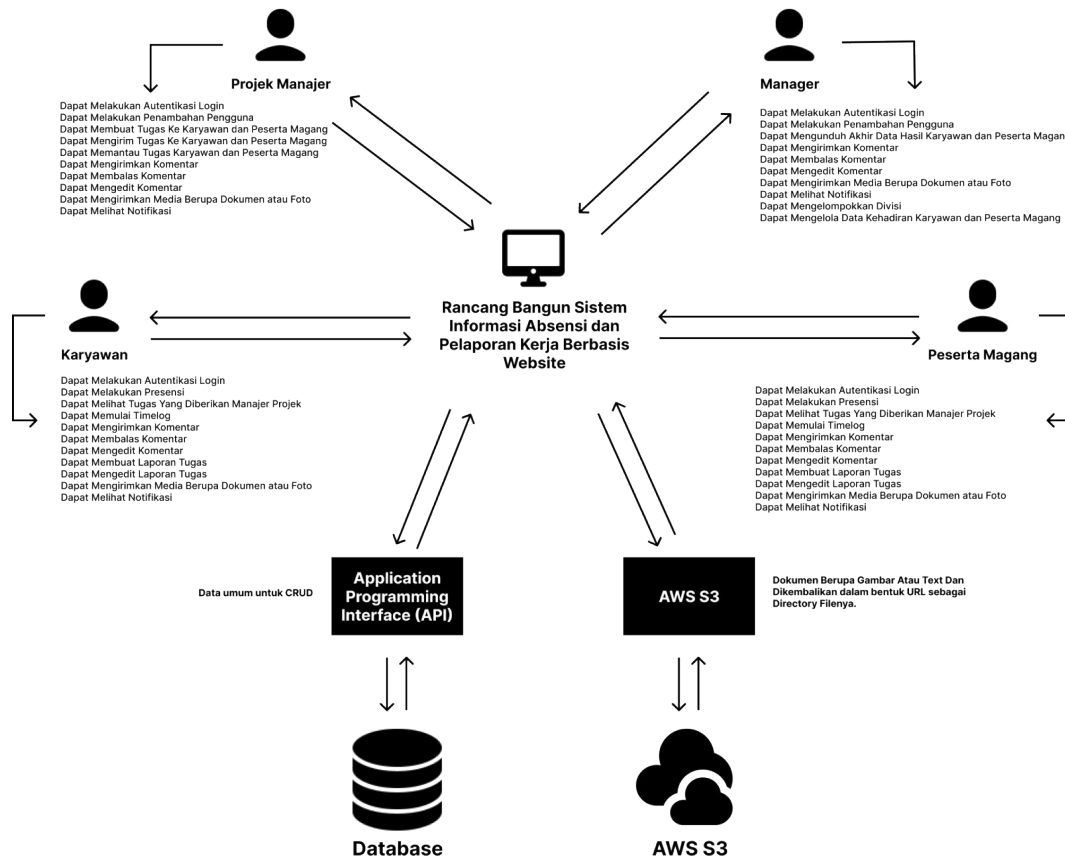
3.1.1. Proses Bisnis Berjalan

Proses absensi dan pelaporan kinerja di PT. Mitra Integritas Indonesia saat ini menggunakan Slack sebagai platform komunikasi utama. Setiap karyawan dan peserta magang diwajibkan mengunggah foto diri yang diambil di lokasi tempat kerja ke saluran *channel* Slack yang telah ditentukan sebagai bukti kehadiran harian. Secara paralel, laporan kerja harian disusun oleh masing-masing pengguna menggunakan aplikasi Microsoft Word. Laporan tersebut memuat deskripsi kegiatan yang telah dilakukan selama hari kerja, dilengkapi dengan tangkapan layar *screenshot* sebagai bukti aktivitas. Dokumen laporan kemudian dikirim melalui pesan langsung atau diposting ke saluran Slack yang sama tempat absensi dilakukan.

Seluruh data absensi dan laporan kinerja tetap berada dalam bentuk tidak terstruktur dan tersebar di dalam riwayat percakapan Slack. Setiap akhir bulan, manajemen melakukan rekapitulasi data secara manual dengan meninjau ulang unggahan foto dan dokumen Word satu per satu. Proses ini memerlukan waktu yang lama, tidak memiliki standar format yang konsisten, serta rentan terhadap kelalaian atau kehilangan data akibat keterbatasan pencarian dan arsip dalam platform komunikasi tersebut.

3.1.2. Gambaran Umum Sistem

Gambaran Umum Sistem adalah sebuah deskripsi singkat atau ilustrasi yang memberikan penjelasan menyeluruh tentang sistem. Berdasarkan pengertian tersebut, peneliti akan menjelaskan melalui Gambar 2 yang akan menggambarkan sistem yang akan diterapkan.



Gambar 3. Gambaran Umum Sistem

3.2. Perancangan

3.2.1. Kebutuhan Fungsional

Menurut S Tiun et al (2020:1-2) *Functional Requirement (FR)* dapat diartikan sebagai Kebutuhan perangkat lunak misalnya fitur eksplisit, atau fungsi dari produk. Berdasarkan pengertian tersebut, maka peneliti akan memaparkan FR dalam bentuk tabel sebagai berikut.

Tabel 2. Kebutuhan Fungsional

| No | Aktor | Deskripsi |
|----|-------|-----------|
|----|-------|-----------|

| | | |
|-------|----------------|--|
| FR-1 | Manajer | Manajer dapat melakukan autentikasi login ke dalam aplikasi |
| FR-2 | Manajer | Manajer dapat melihat riwayat kehadiran karyawan dan peserta magang |
| FR-3 | Manajer | Manajer dapat mengelola hak akses pengguna lain dalam aplikasi |
| FR-4 | Manajer | Manajer dapat mengunduh laporan kehadiran karyawan dan peserta magang |
| FR-5 | Manajer | Manajer dapat menambahkan akun baru ke dalam aplikasi |
| FR-6 | Manajer | Manajer dapat menghapus akun karyawan dan peserta magang yang tidak bekerja di perusahaan lagi |
| FR-7 | Manajer | Manajer dapat mengelompokkan divisi karyawan dan peserta magang |
| FR-8 | Manajer | Manajer dapat logout dari aplikasi |
| FR-9 | Proyek Manajer | Proyek Manajer dapat melihat data karyawan dan peserta magang yang berada di bawah tanggung jawabnya |
| FR-10 | Proyek Manajer | Proyek Manajer dapat memberikan tugas kepada karyawan dan peserta magang yang berada di bawah tanggung jawabnya |
| FR-11 | Proyek Manajer | Proyek Manajer dapat memberikan komentar tugas kepada karyawan dan peserta magang yang berada di bawah tanggung jawabnya |
| FR-12 | Proyek Manajer | Proyek Manajer dapat memantau progress tugas karyawan dan peserta magang yang berada di bawah tanggung jawabnya |
| FR-13 | Proyek Manajer | Proyek Manajer dapat menghapus tugas karyawan dan peserta magang yang berada di bawah tanggung jawabnya |
| FR-14 | Proyek Manajer | Proyek Manajer dapat logout dari aplikasi |
| FR-15 | Karyawan | Karyawan dapat melakukan autentikasi login ke dalam aplikasi |
| FR-16 | Karyawan | Karyawan dapat mengirimkan bukti kehadiran |
| FR-17 | Karyawan | Karyawan dapat melihat daftar tugas yang diberikan |

| | | |
|-------|----------------|--|
| | | oleh proyek manajer |
| FR-18 | Karyawan | Karyawan dapat memulai tugas dengan memulai timelog |
| FR-19 | Karyawan | Karyawan dapat mengirim bukti penyelesaian tugas |
| FR-20 | Karyawan | Karyawan dapat mengunduh dokumen yang diberikan proyek manajer |
| FR-21 | Karyawan | Karyawan dapat logout dari aplikasi |
| FR-22 | Peserta Magang | Peserta Magang dapat melakukan autentikasi login ke dalam aplikasi |
| FR-23 | Peserta Magang | Peserta Magang dapat mengirimkan bukti kehadiran |
| FR-24 | Peserta Magang | Peserta Magang dapat melihat daftar tugas yang diberikan oleh proyek manajer |
| FR-25 | Peserta Magang | Peserta Magang dapat memulai tugas dengan memulai timelog |
| FR-26 | Peserta Magang | Peserta Magang dapat mengirim bukti penyelesaian tugas |
| FR-27 | Peserta Magang | Peserta Magang dapat mengunduh dokumen yang diberikan proyek manajer |
| FR-28 | Peserta Magang | Peserta Magang dapat logout dari aplikasi |

3.2.2. Kebutuhan Non-Fungsional

Menurut S Tiun et al (2020:1-2) *Non-Functional Requirement* (NFR) adalah standar kualitas yang tidak secara langsung terlihat (implisit), namun sangat penting untuk sebuah produk. Berdasarkan pengertian tersebut, maka peneliti akan memaparkan NFR dalam bentuk tabel sebagai berikut.

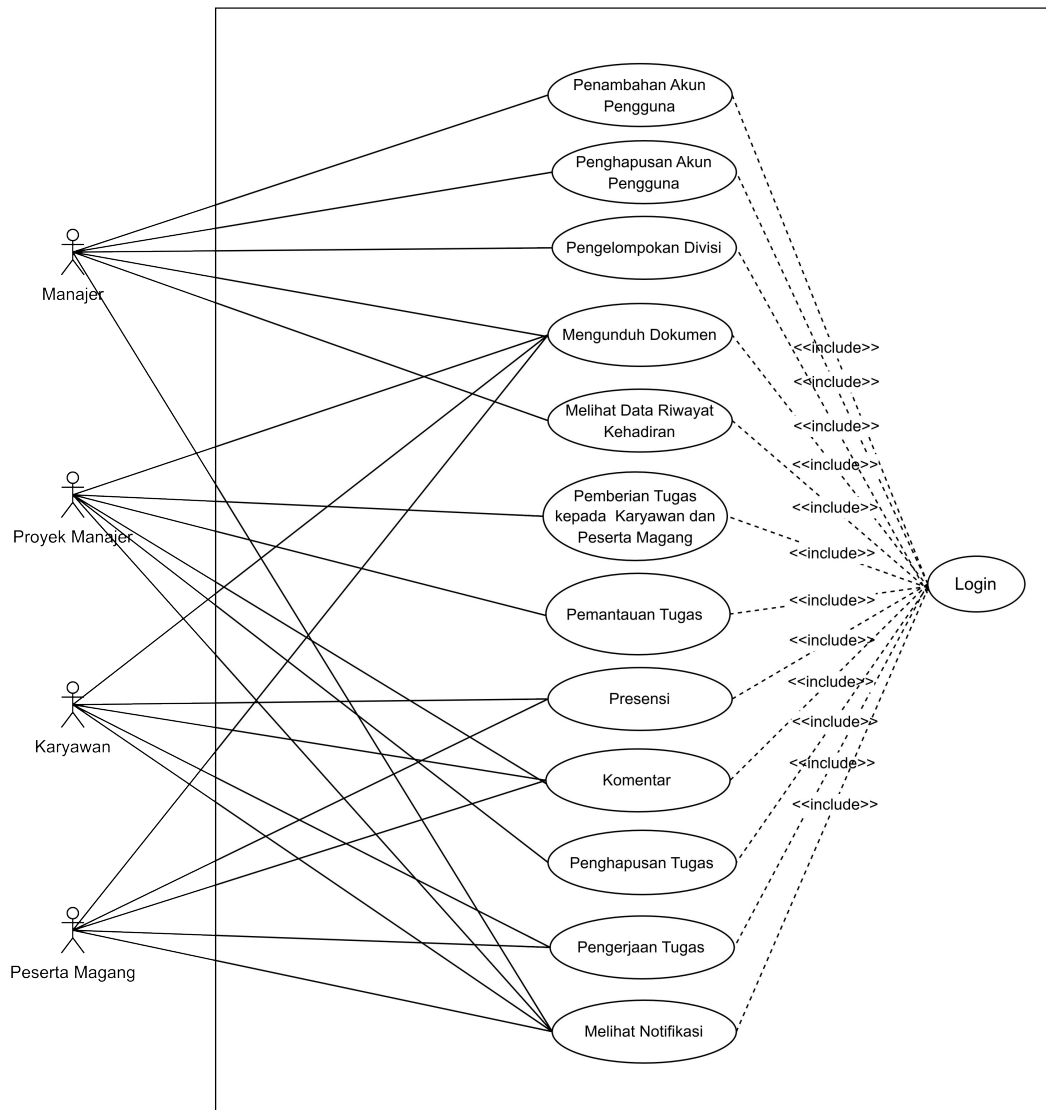
Tabel 3. Kebutuhan Non-Fungsional

| No | Kategori | Deskripsi |
|----|----------|-----------|
|----|----------|-----------|

| | | |
|--------|-------------------------|--|
| NFR-1 | Operational Requirement | Aplikasi berjalan ketika terhubung dengan internet |
| NFR-2 | Operational Requirement | Aplikasi dapat diakses hanya pada jam kerja |
| NFR-3 | Operational Requirement | Aplikasi dapat dijalankan melalui desktop dan mobile menggunakan web browser |
| NFR-4 | Operational Requirement | Aplikasi dapat beradaptasi / responsif terhadap layar pengguna |
| NFR-5 | Operational Requirement | Aplikasi memiliki User Interface yang modern |
| NFR-6 | Operational Requirement | Aplikasi memiliki User Experience yang mudah digunakan |
| NFR-7 | Operational Requirement | Aplikasi menggunakan soft delete / teknik menghapus yang tidak permanen |
| NFR-8 | Operational Requirement | Aplikasi memberikan respons yang ramah pengguna dalam kasus kegagalan, seperti pesan kesalahan yang jelas dan informatif |
| NFR-9 | Performance Requirement | Aplikasi memproses data dengan cepat, estimasi harus kurang dari 3 detik |
| NFR-10 | Performance Requirement | Aplikasi dapat menangani banyak pengguna dalam satu waktu tanpa mengalami penurunan kinerja |
| NFR-11 | Security Requirement | Aplikasi hanya dapat diakses oleh yang berwenang |
| NFR-12 | Security Requirement | Aplikasi menggunakan HTTPS dengan sertifikat SSL yang valid |
| NFR-13 | Security Requirement | Aplikasi menggunakan bearer token autentikasi dan otorisasi |

3.2.2. Diagram Use Case

Menurut Mulyani, S (2016:80) Pengertian *Use Case Diagram* adalah salah satu komponen dari *use case model* yang menggambarkan proses serta bagaimana sistem berinteraksi atau berhubungan dengan aktor. Berdasarkan pengertian tersebut maka peneliti menyusun *use case diagram* seperti pada gambar 3.



Gambar 4. Gambar Diagram Use Case

3.2.3. Skenario Use Case

Tabel 4. Skenario Use Case Manajer Login

| | |
|------------------------|--|
| Nama | Manajer Login |
| Tujuan | Masuk ke dalam sistem |
| Deskripsi | Langkah awal manajer untuk memasuki sistem |
| Aktor | Manajer |
| Skenario Normal | |
| Kondisi Awal | Form Login muncul di layar manajer |

| | |
|--|--|
| Aksi Aktor | Reaksi Sistem |
| Mengisi Form Login sesuai kredensial | Memeriksa input sesuai dengan validasi <i>frontend</i> dan <i>backend</i> |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| Mengisi Form Login tidak sesuai kredensial | Memeriksa input sesuai dengan validasi frontend, memunculkan pesan error dan <i>redirect</i> ke halaman login |
| Mengisi Form Login kembali dengan benar | Memeriksa input sesuai dengan validasi <i>frontend</i> dan <i>backend</i> , kemudian <i>redirect</i> ke <i>dashboard</i> |
| Kondisi Akhir | Manajer dapat mengakses dashboard dan fitur lainnya |

Tabel 5. Skenario *Use Case* Manajer Proyek Login

| | |
|--|--|
| Nama | Manajer Proyek Login |
| Tujuan | Masuk ke dalam sistem |
| Deskripsi | Langkah awal manajer proyek untuk memasuki sistem |
| Aktor | Manajer Proyek |
| Skenario Normal | |
| Kondisi Awal | Form Login muncul di layar manajer Proyek |
| Aksi Aktor | Reaksi Sistem |
| Mengisi Form Login sesuai kredensial | Memeriksa input sesuai dengan validasi <i>frontend</i> dan <i>backend</i> |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| Mengisi Form Login tidak sesuai kredensial | Memeriksa input sesuai dengan validasi frontend, memunculkan pesan error dan <i>redirect</i> ke halaman login |
| Mengisi Form Login kembali dengan benar | Memeriksa input sesuai dengan validasi <i>frontend</i> dan <i>backend</i> , kemudian <i>redirect</i> ke <i>dashboard</i> |
| | |
| Kondisi Akhir | Manajer Proyek dapat mengakses dashboard dan fitur lainnya |

Tabel 6. Skenario *Use Case* Karyawan dan Peserta Magang Login

| | |
|---------------|-----------------------------------|
| Nama | Karyawan dan Peserta Magang Login |
| Tujuan | Masuk ke dalam sistem |

| | |
|--|--|
| Deskripsi | Langkah awal Karyawan dan Peserta Magang untuk memasuki sistem |
| Aktor | Karyawan dan Peserta Magang |
| Skenario Normal | |
| Kondisi Awal | Form Login muncul di layar Karyawan dan Peserta Magang |
| Aksi Aktor | Reaksi Sistem |
| Mengisi Form Login sesuai kredensial | Memeriksa input sesuai dengan validasi <i>frontend</i> dan <i>backend</i> , memeriksa apakah sudah melakukan presensi |
| | Presensi belum dilakukan maka akan redirect ke halaman presensi |
| Mengisi Form Presensi | Memeriksa input sesuai validasi <i>frontend</i> dan <i>backend</i> |
| | <i>Redirect</i> ke <i>dashboard</i> dan mengakses fitur lainnya |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| Mengisi Form Login tidak sesuai kredensial | Memeriksa input sesuai dengan validasi <i>frontend</i> , memunculkan pesan error dan <i>redirect</i> ke halaman login |
| Mengisi Form Login kembali dengan benar | Memeriksa input sesuai dengan validasi <i>frontend</i> dan <i>backend</i> , kemudian <i>redirect</i> ke <i>dashboard</i> |
| | |
| Kondisi Akhir | Karyawan dan Peserta Magang dapat mengakses <i>dashboard</i> dan fitur lainnya |

Tabel 7. Skenario *Use Case* Manajer Penambahan Akun Pengguna

| | |
|------------------------|--|
| Nama | Manajer Penambahan Akun Pengguna |
| Tujuan | Menambahkan pengguna baru |
| Deskripsi | Menambahkan pengguna baru yang dapat mengakses sistem sesuai <i>rolnya</i> |
| Aktor | Manajer |
| Skenario Normal | |
| Kondisi Awal | Form Penambahan Akun Pengguna berupa <i>modal</i> telah dimunculkan di layar Manajer |
| Aksi Aktor | Reaksi Sistem |
| Mengisi Form | Memeriksa input sesuai dengan validasi <i>frontend</i> dan |

| | |
|--|---|
| Penambahan Akun Pengguna | <i>backend</i> |
| | Menambahkan Akun Pengguna dan menutup <i>modal</i> |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| Mengisi Form Penambahan Akun namun masih ada yang kosong | Memeriksa input sesuai dengan validasi frontend, memunculkan pesan error |
| Mengisi Form penambahan akun dengan benar | Memeriksa input sesuai dengan validasi <i>frontend</i> dan <i>backend</i> |
| | Menambahkan Akun Pengguna dan menutup <i>modal</i> |
| Kondisi Akhir | Pengguna dapat dilihat dalam daftar pengguna |

Tabel 8. Skenario *Use Case* Manajer Penghapusan Akun Pengguna

| | |
|-----------------------------------|--|
| Nama | Manajer Penghapusan Akun Pengguna |
| Tujuan | Menghapus pengguna |
| Deskripsi | Menghapus pengguna dari sistem |
| Aktor | Manajer |
| Skenario Normal | |
| Kondisi Awal | Halaman <i>detail</i> Akun Pengguna yang berisi tombol <i>delete</i> di layar Manajer |
| Aksi Aktor | Reaksi Sistem |
| Menekan tombol <i>delete user</i> | Memeriksa pengguna, menghapus pengguna dari sistem dan redirect ke <i>list</i> pengguna dan memunculkan pesan berhasil menghapus |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Pengguna terhapus dari daftar pengguna |

Tabel 9. Skenario *Use Case* Manajer Pengelompokan Divisi Pengguna

| | |
|---------------|--|
| Nama | Manajer Mengelompokkan Divisi Pengguna |
| Tujuan | Mengelompokkan divisi pengguna |

| | |
|--|--|
| Deskripsi | Manajer dapat mengelompokkan divisi pengguna sesuai dengan divisi masing-masing |
| Aktor | Manajer |
| Skenario Normal | |
| Kondisi Awal | Halaman <i>Division</i> |
| Aksi Aktor | Reaksi Sistem |
| Menyeret pengguna ke divisi yang dikehendaki | Menyimpan <i>state id</i> dan Memeriksa pengguna sebelumnya ke dalam backend dengan mengambil <i>id</i> kemudian menimpa pengguna dengan <i>state id</i> tersebut dengan <i>division</i> baru. Kemudian memunculkan pesan berhasil dipindahkan |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Pengguna berada di dalam divisi baru |

Tabel 10. Skenario *Use Case* Manajer Mengunduh Dokumen

| | |
|------------------------------|---|
| Nama | Manajer Mengunduh dokumen |
| Tujuan | Mengunduh dokumen yang terdapat pada sistem |
| Deskripsi | Manajer dapat mengunduh semua dokumen terkait yang terdapat pada sistem |
| Aktor | Manajer |
| Skenario Normal | |
| Kondisi Awal | Halaman <i>Detail</i> pengguna |
| Aksi Aktor | Reaksi Sistem |
| Menekan tombol unduh dokumen | Mengunduh dokumen sesuai <i>filter</i> yang telah ditetapkan |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Dokumen terunduh |

Tabel 11. Skenario *Use Case* Manajer Melihat Data Riwayat Kehadiran

| | |
|---------------|--|
| Nama | Manajer Melihat Data Riwayat Kehadiran |
| Tujuan | Manajer dapat melihat data riwayat kehadiran |

| | |
|---|---|
| | pengguna |
| Deskripsi | Manajer dapat melihat data riwayat kehadiran pengguna |
| Aktor | Manajer |
| Skenario Normal | |
| Kondisi Awal | Halaman <i>Attendance</i> pengguna |
| Aksi Aktor | Reaksi Sistem |
| Melihat data riwayat kehadiran pengguna | Menampilkan data riwayat kehadiran pengguna |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Memaparkan riwayat kehadiran pengguna |

Tabel 12. Skenario *Use Case* Manajer Proyek Memberikan Tugas

| | |
|--|--|
| Nama | Manajer Proyek Memberikan Tugas |
| Tujuan | Manajer Proyek dapat memberikan tugas |
| Deskripsi | Manajer Proyek memberikan tugas kepada pengguna yang dikehendaki |
| Aktor | Manajer Proyek |
| Skenario Normal | |
| Kondisi Awal | Halaman <i>Assignment</i> pengguna |
| Aksi Aktor | Reaksi Sistem |
| Menekan tombol <i>create assignemnt</i> | Menampilkan <i>modal form create assignment</i> |
| Mengisi <i>form create assignment</i> dan menekan tombol <i>submit</i> | Memasukkan tugas ke sistem, menutup <i>modal create assignment</i> dan memberikan notifikasi ke pengguna terkait |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Tugas dapat dilihat di <i>list</i> |

Tabel 13. Skenario *Use Case* Manajer Proyek Memantau Tugas

| | |
|------------------------------------|---|
| Nama | Manajer Proyek Memantau Tugas |
| Tujuan | Manajer Proyek dapat memantau tugas |
| Deskripsi | Manajer Proyek memantau tugas yang diberikan kepada pengguna yang dikehendaki |
| Aktor | Manajer Proyek |
| Skenario Normal | |
| Kondisi Awal | Halaman <i>dashboard</i> |
| Aksi Aktor | Reaksi Sistem |
| Melihat tugas yang sedang berjalan | Menampilkan tugas yang sedang berjalan yang sedang dilakukan pengguna |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Tugas yang sedang berjalan yang sedang dilakukan pengguna dapat dilihat |

Tabel 14. Skenario *Use Case* Karyawan dan Peserta Magang Presensi

| | |
|--|--|
| Nama | Karyawan dan Peserta Magang Presensi |
| Tujuan | Karyawan dan Peserta Magang melakukan presensi |
| Deskripsi | Karyawan dan Peserta Magang melakukan presensi setelah login |
| Aktor | Karyawan dan Peserta Magang |
| Skenario Normal | |
| Kondisi Awal | Halaman Presensi |
| Aksi Aktor | Reaksi Sistem |
| Melakukan Presensi dengan mengisi <i>form presensi</i> | Memeriksa input sesuai validasi <i>frontend</i> dan <i>backend</i> |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Menuju <i>Dashboard</i> |

Tabel 15. Skenario *Use Case* Manajer Proyek Komentar

| | |
|---------------------------------------|--|
| Nama | Manajer Proyek Komentar |
| Tujuan | Manajer Proyek mengomentari tugas |
| Deskripsi | Manajer Proyek mengomentari tugas yang telah diberikan kepada pengguna |
| Aktor | Manajer Proyek |
| Skenario Normal | |
| Kondisi Awal | <i>Modal</i> Komentar |
| Aksi Aktor | Reaksi Sistem |
| Melakukan Komentar dan menekan submit | Memasukkan komentar ke tugas terkait |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Komentar terdapat pada <i>list</i> |

Tabel 16. Skenario *Use Case* Karyawan dan Peserta Magang Komentar

| | |
|---------------------------------------|---|
| Nama | Karyawan dan Peserta Magang Komentar |
| Tujuan | Karyawan dan Peserta Magang mengomentari tugas |
| Deskripsi | Karyawan dan Peserta Magang mengomentari tugas yang telah diberikan dari manajer proyek |
| Aktor | Karyawan dan Peserta Magang |
| Skenario Normal | |
| Kondisi Awal | <i>Modal</i> Komentar |
| Aksi Aktor | Reaksi Sistem |
| Melakukan Komentar dan menekan submit | Memasukkan komentar ke tugas terkait |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Komentar terdapat pada <i>list</i> |

Tabel 17. Skenario *Use Case* Manajer Proyek Menghapus Tugas

| | |
|-------------|--------------------------------|
| Nama | Manajer Proyek Menghapus Tugas |
|-------------|--------------------------------|

| | |
|---|---|
| Tujuan | Manajer Proyek Menghapus Tugas |
| Deskripsi | Manajer Proyek dapat menghapus tugas yang dikehendaki |
| Aktor | Manajer Proyek |
| Skenario Normal | |
| Kondisi Awal | <i>Modal Assignment</i> |
| Aksi Aktor | Reaksi Sistem |
| Melakukan penghapusan tugas dengan menekan tombol hapus | Menghapus tugas dan menampilkan pesan berhasil |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Tugas dihapus dari <i>list</i> |

Tabel 18. Skenario *Use Case* Karyawan dan Peserta Magang Mengerjakan Tugas

| | |
|--|--|
| Nama | Karyawan dan Peserta Magang Mengerjakan Tugas |
| Tujuan | Karyawan dan Peserta Magang Mengerjakan Tugas |
| Deskripsi | Karyawan dan Peserta Magang Mengerjakan Tugas |
| Aktor | Karyawan dan Peserta Magang |
| Skenario Normal | |
| Kondisi Awal | <i>Modal Assignment</i> |
| Aksi Aktor | Reaksi Sistem |
| Melakukan pengerjaan tugas dengan menekan tombol start timelog | Pengerjaan dimulai dengan memulai timelog dan menampilkan timelog di sebelah kanan layar |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Pengerjaan tugas dengan menampilkan timelog di sebelah kanan layar |

Tabel 19. Skenario *Use Case* Manajer Proyek Melihat Notifikasi

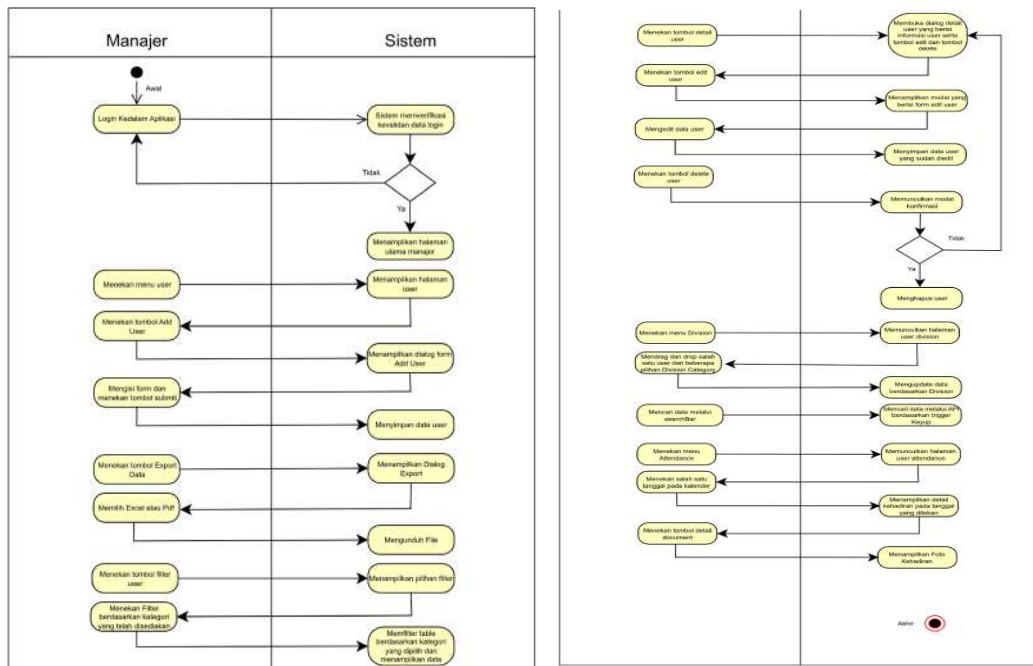
| | |
|-------------|-----------------------------------|
| Nama | Manajer Proyek Melihat Notifikasi |
|-------------|-----------------------------------|

| | |
|---|---|
| Tujuan | Manajer Proyek Melihat Notifikasi |
| Deskripsi | Manajer Proyek Melihat Notifikasi |
| Aktor | Manajer Proyek |
| Skenario Normal | |
| Kondisi Awal | <i>Modal Assignment</i> |
| Aksi Aktor | Reaksi Sistem |
| Menekan tombol lonceng di <i>header</i> | Memunculkan notifikasi |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| - | - |
| Kondisi Akhir | Muncul popover notifikasi di atas layar |

3.2.4. Activity Diagram

Pengertian Activity Diagram menurut Veronika Ivanova, Ani Boneva dan Plamen Vasilev (2024:120-121) adalah salah satu diagram dalam UML yang menggambarkan alur perilaku dinamis sistem, termasuk alur aktivitas secara berurutan, bercabang, paralel, dan simultan. Berdasarkan pengertian tersebut maka peneliti akan memaparkan beberapa gambar terkait sebagai berikut.

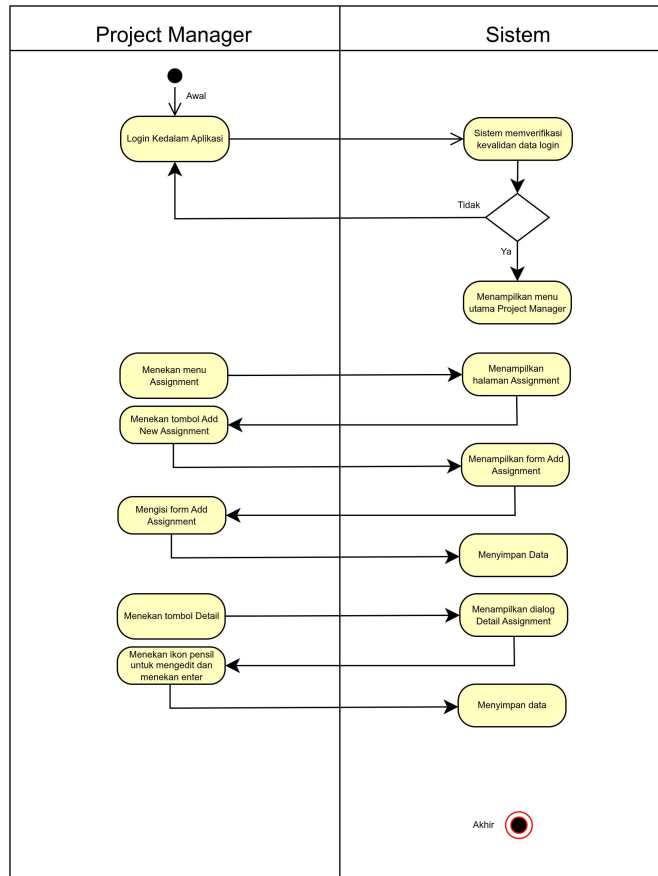
1. Activity Diagram Manager



Gambar 5. Gambar Activity Diagram Manager

Gambar di atas ini adalah *activity diagram* yang menunjukkan aktivitas *manager* dalam aplikasi Rancang Bangun Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Web. Dalam aplikasi ini, *manager* berperan sebagai user yang memiliki hak akses untuk mengatur *user*, *attendance* dan *division*. Proses dimulai ketika *manager* login ke dalam aplikasi dengan cara mengisi form login dengan data kredensial yang telah dibuat dan menekan tombol login, kemudian sistem akan memvalidasi data kredensial yang dikirimkan, jika valid maka sistem akan memunculkan halaman utama *manager* dan apabila tidak valid maka sistem tetap pada halaman login. Kemudian, *manager* dapat mengelola *user*, *attendance*, *division*.

2. Activity Diagram Project Manager

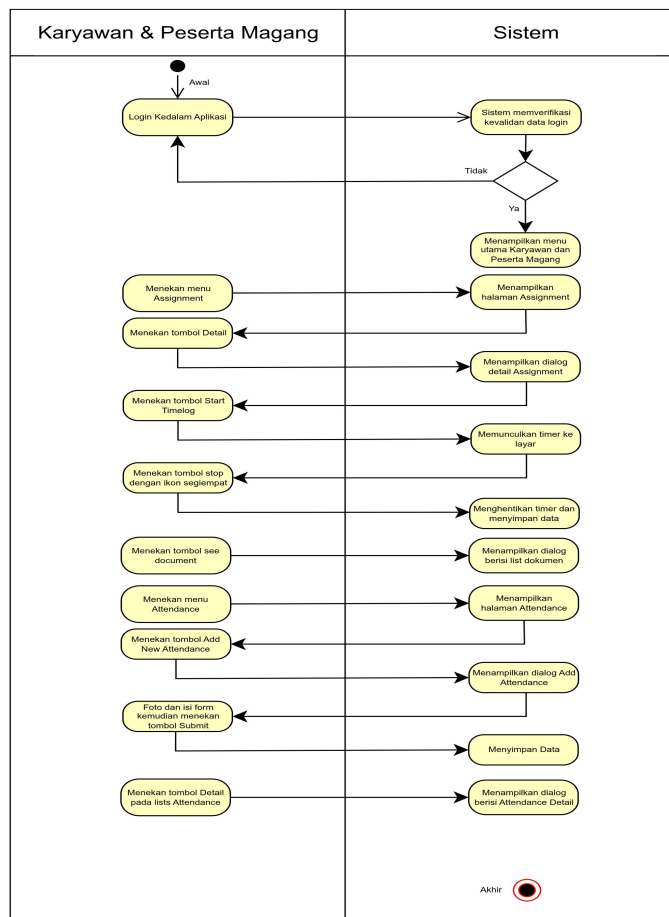


Gambar 6. Gambar Activity Diagram Project Manager

Gambar di atas ini adalah *activity diagram* yang menunjukkan aktivitas *project manager* dalam aplikasi Rancang Bangun Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Web. Dalam aplikasi ini, *project manager* berperan sebagai user yang memiliki hak akses untuk mengatur *user* dan *task*. Proses dimulai ketika *project manager* login ke dalam aplikasi dengan cara mengisi form login dengan data kredensial yang telah dibuat dan menekan tombol login, kemudian sistem akan memvalidasi data kredensial yang dikirimkan, jika valid maka sistem akan memunculkan halaman utama *project manager* dan apabila tidak valid maka sistem tetap pada halaman login. Terdapat beberapa perbedaan antara

manager dan *project manager*, seperti yang dipaparkan pada gambar 5 bahwa *manager* dapat mengatur *user*, *division* yang berfokus pada operasional perusahaan sedangkan *project manager* hanya dapat mengatur *user* dan *task* yang berfokus pada pelaksanaan dan pengawasan tugas.

3. Activity Diagram Karyawan & Peserta Magang



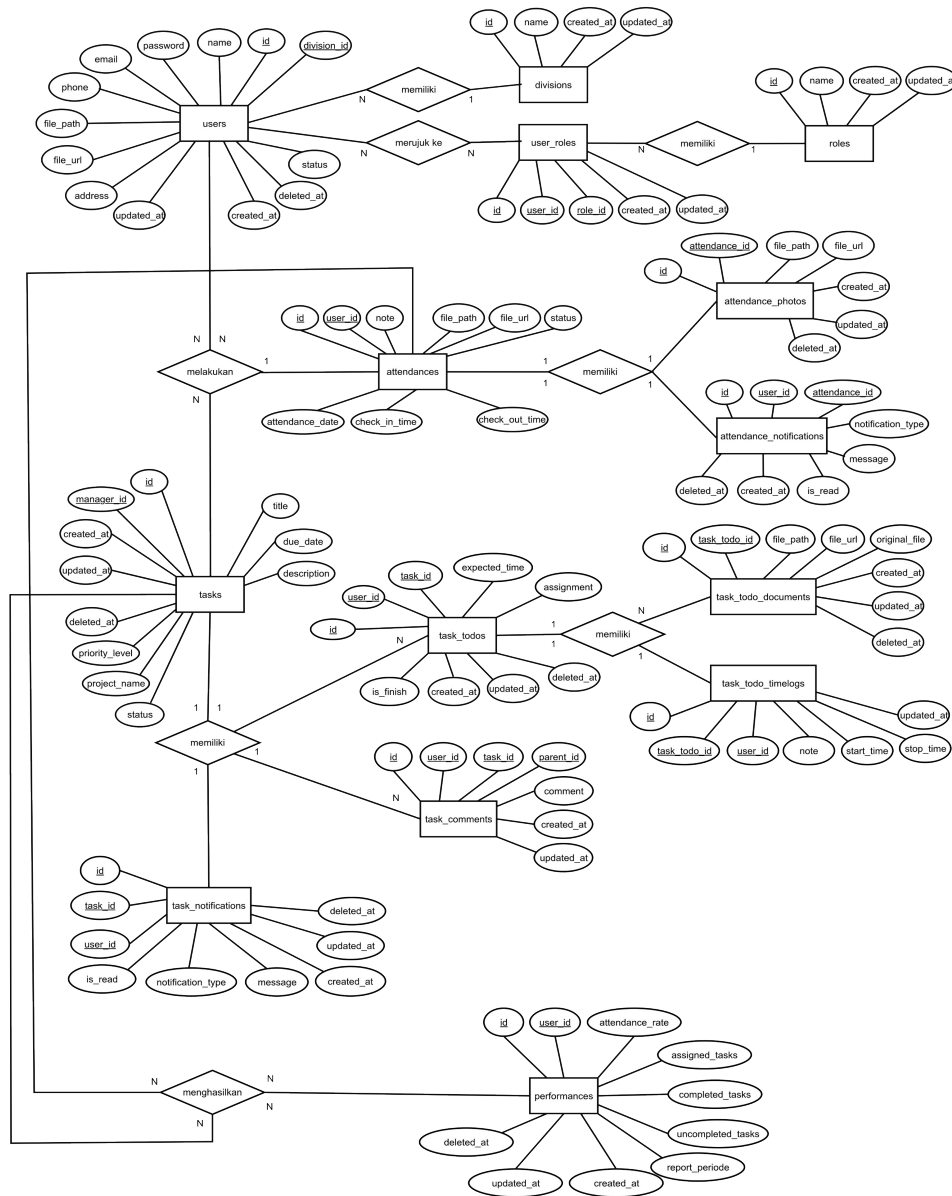
Gambar 7. Gambar Activity Diagram Karyawan & Peserta Magang

Gambar di atas ini adalah activity diagram yang menunjukkan aktivitas karyawan dan peserta magang dalam aplikasi Rancang Bangun Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Web. Dalam aplikasi ini, karyawan dan peserta magang berperan sebagai user yang

memiliki hak akses untuk mengatur user dan task. Proses dimulai ketika karyawan dan peserta magang login ke dalam aplikasi dengan cara mengisi form login dengan data kredensial yang telah dibuat dan menekan tombol login, kemudian sistem akan memvalidasi data kredensial yang dikirimkan, jika valid maka sistem akan memunculkan halaman utama karyawan dan peserta magang dan apabila tidak valid maka sistem tetap pada halaman login. Kemudian, karyawan dan peserta magang dapat mengelola *task* dan *attendance*, dengan melakukan pengaturan sesuai gambar diagram yang telah dipaparkan di bawah.

3.2.5. Entity Relationship Diagram

Menurut Dedy Rahman Prehanto (2020:117) *Entity Relationship Diagram (ERD)* adalah *high level conceptual* untuk memfasilitasi perancangan database. Konsep-konsep dasar dari *Entity Relationship Diagram* mencakup *Entity*, *Relation* dan *Attribute*. Berdasarkan pengertian tersebut, maka peneliti akan memaparkan gambar *ERD* sebagai berikut.



Gambar 8. Gambar Diagram Use Case

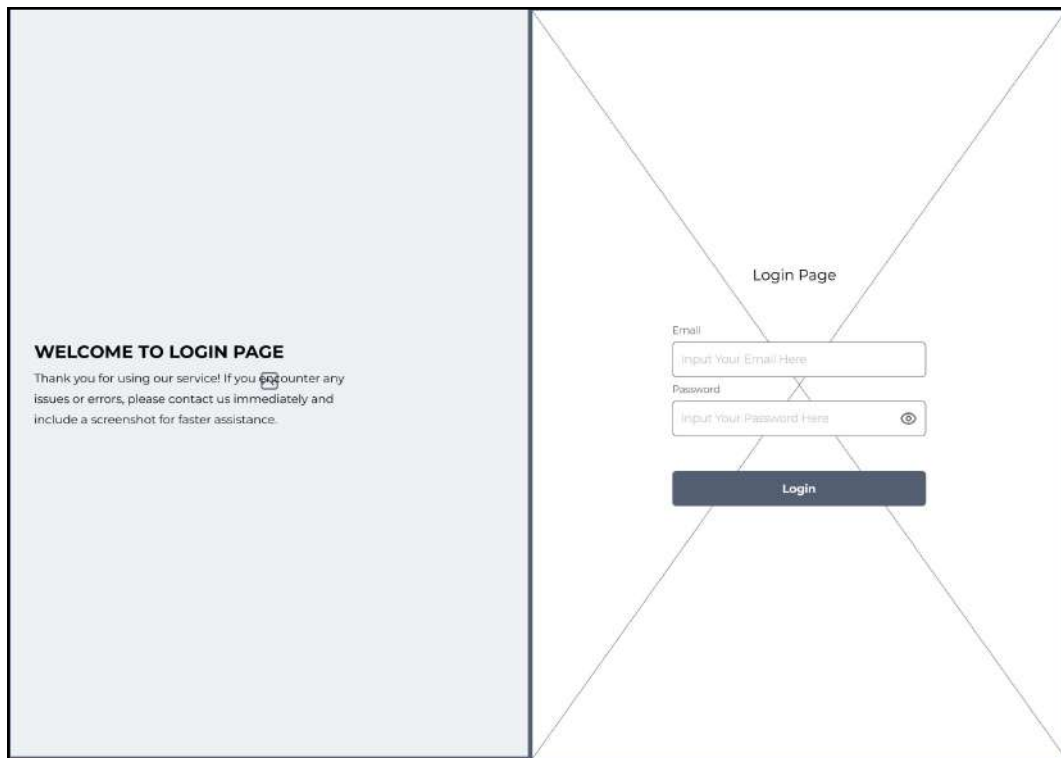
3.2.6. Perancangan Antarmuka (Wireframe)

Pengertian Wireframe menurut Rio Andika Malik dan Marta Riri Frimadani (2021:52) adalah skema atau kerangka yang dapat memberikan gambaran kasar (*low-fidelity*) setiap halaman yang terdapat dalam sebuah aplikasi sebelum memasuki tahapan visual mockup. Berdasarkan

pengertian tersebut maka peneliti menyusun wireframe sebagai gambaran kasar sistem sebagai berikut.

1. *Login Page*

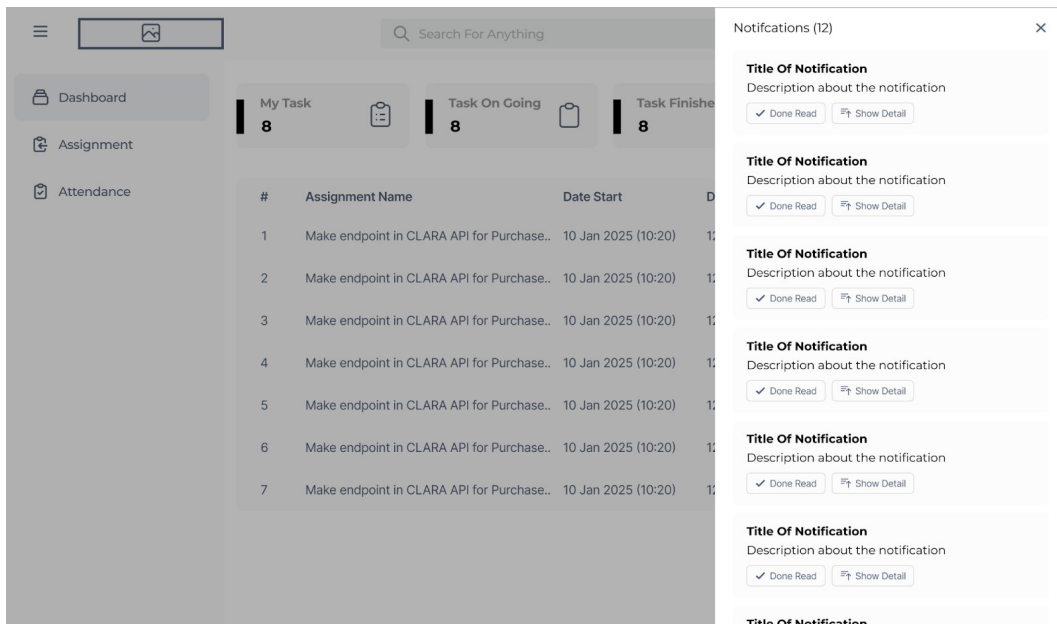
Gambar dibawah merupakan wireframe dari login page yang terdiri dari form login di sebelah kanan dan kata selamat datang di sebelah kiri halaman



Gambar 9. Gambar Diagram Use Case

2. *Notification Layout*

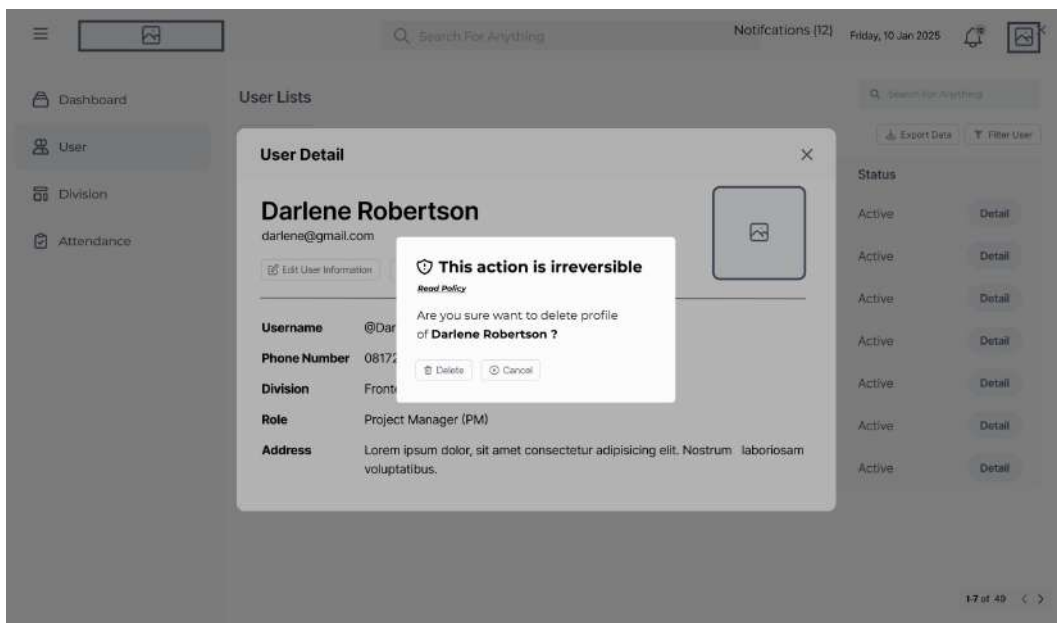
Gambar dibawah merupakan *wireframe* dari *notification layout* yang akan ada di semua pengguna aplikasi ini.



Gambar 10. Notification Layout

3. Confirmation Modal

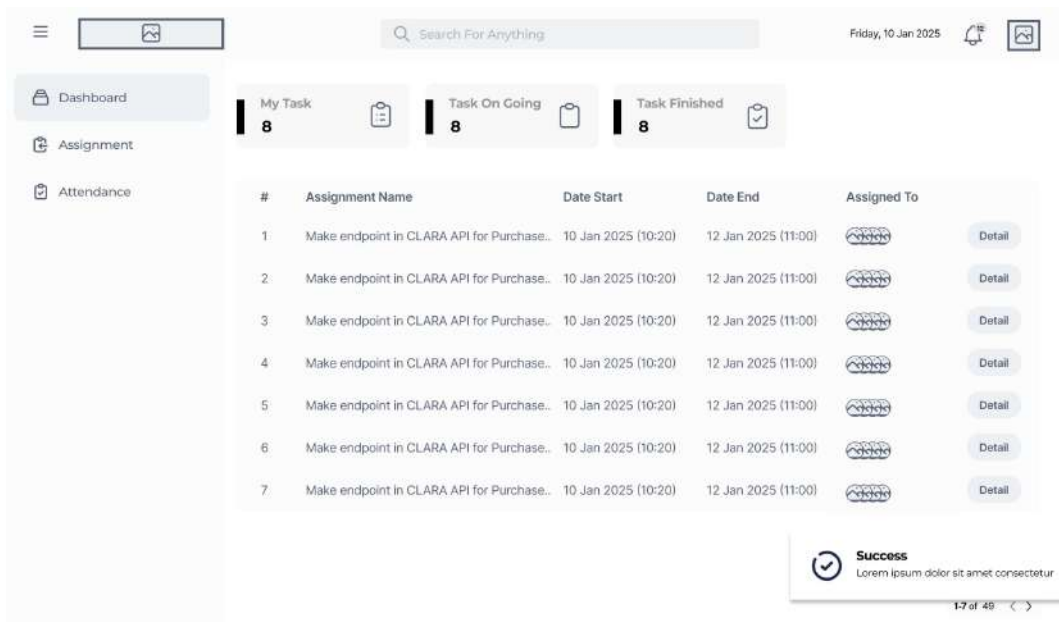
Gambar dibawah merupakan *wireframe* dari *confirmation modal* yang akan ada di semua pengguna aplikasi ini.



Gambar 11. Confirmation Modal

4. Message Toast

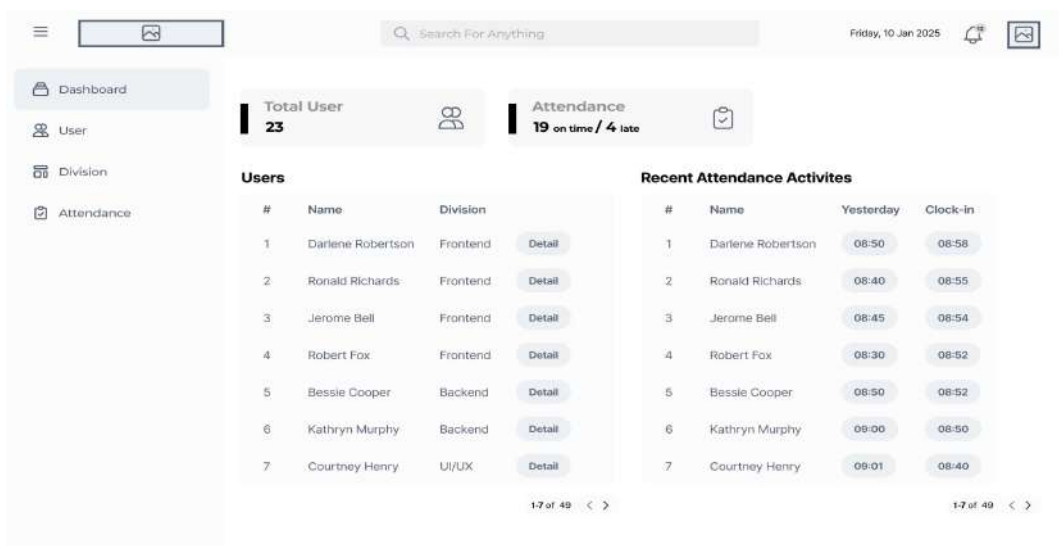
Gambar dibawah merupakan *wireframe* dari *message toast* yang akan ada di semua pengguna aplikasi ini.



Gambar 12. Message Toast

5. Dashboard

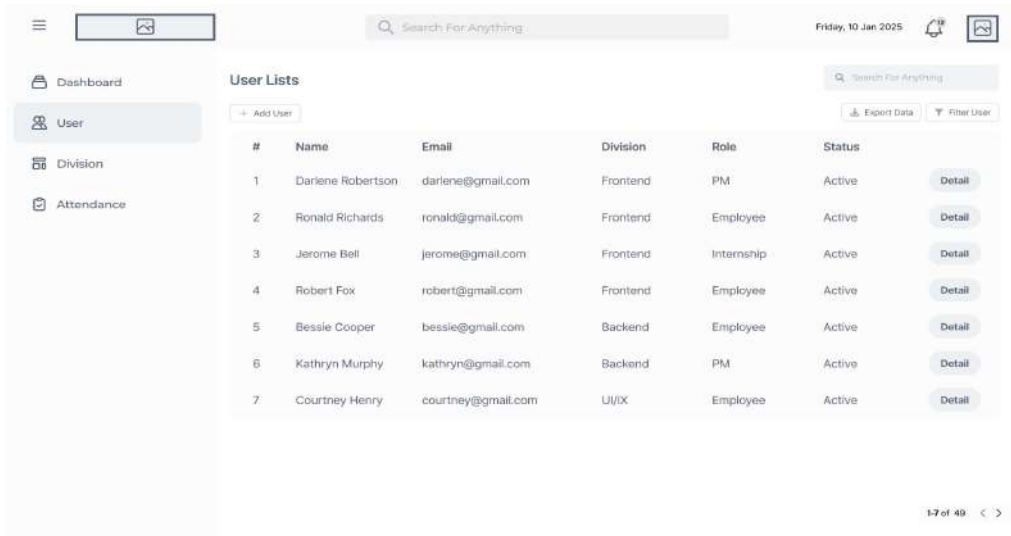
Gambar dibawah merupakan wireframe dashboard dari manager yang menampilkan beberapa rekapitulasi data dari *user* dan *attendance*.



Gambar 13. Dashboard

6. User Page

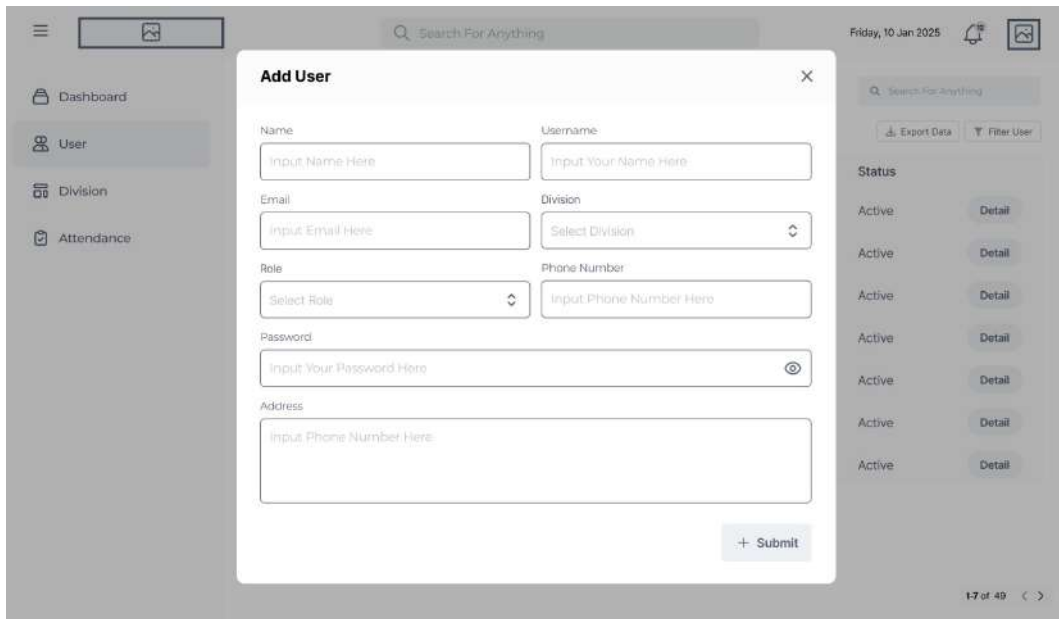
Gambar dibawah merupakan *wireframe user page* dari manager yang menampilkan list dari *user*.



Gambar 14. User Page

7. Add User Modal

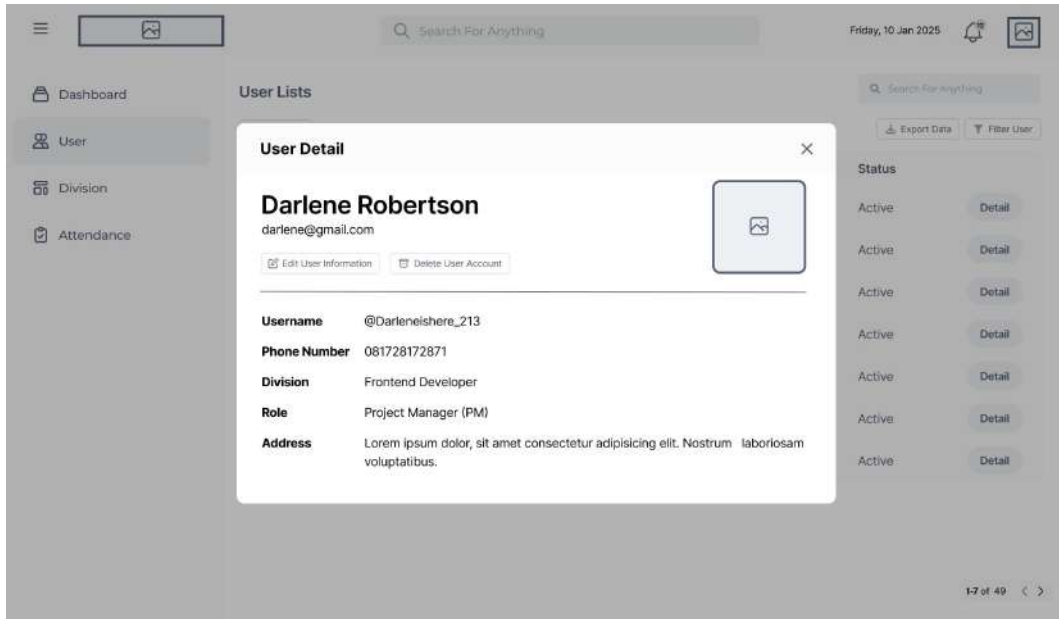
Gambar dibawah merupakan *wireframe add user modal* dari manager yang menampilkan *form* penambahan *user*.



Gambar 15. Add User Modal

8. User Detail Page

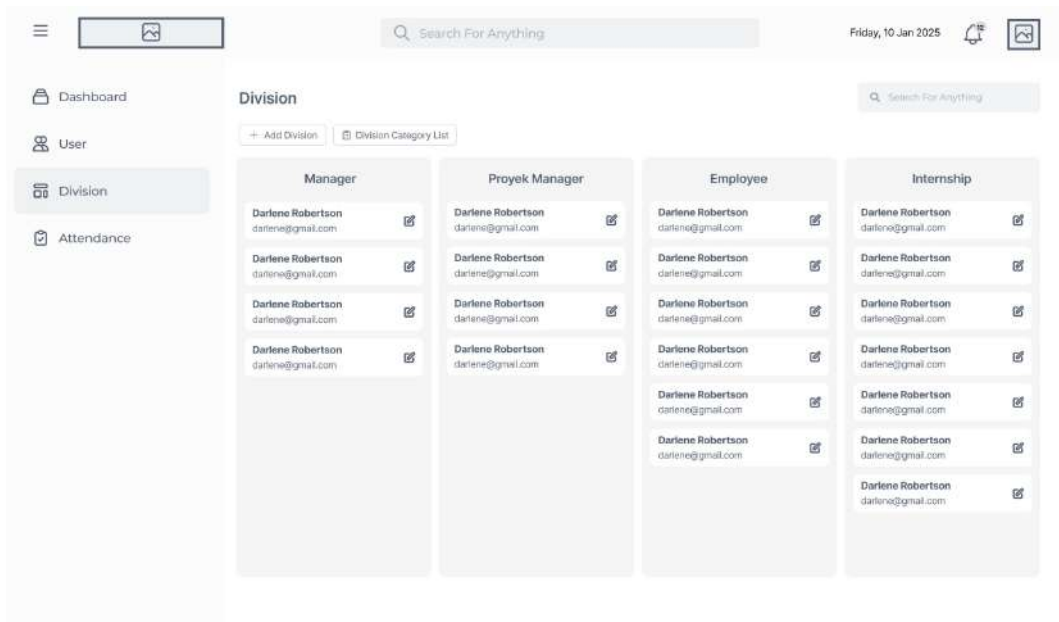
Gambar dibawah merupakan *wireframe User Detail Page* dari manager yang menampilkan *detail user*.



Gambar 16. User Detail Modal

9. Division Page

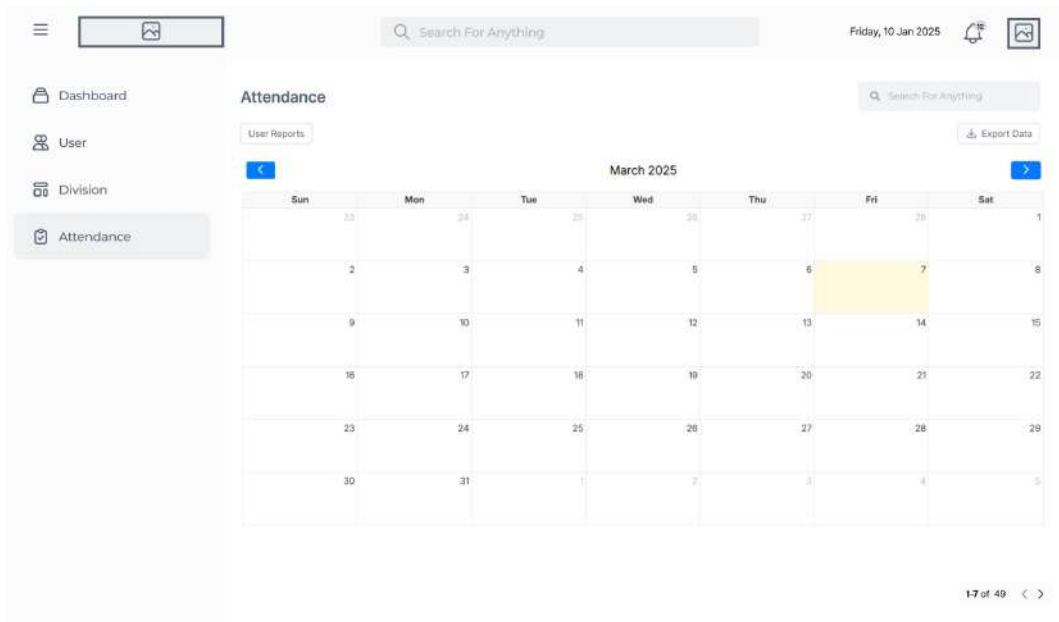
Gambar dibawah merupakan *wireframe Division Page* dari manager yang menampilkan divisi dari *user*.



Gambar 17. Division Page

10. Attendance Page

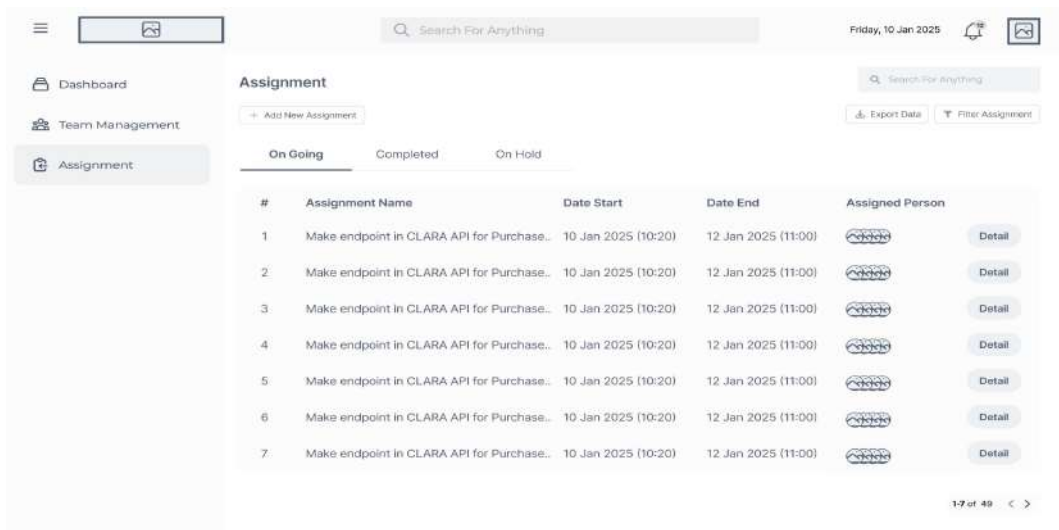
Gambar dibawah merupakan *wireframe attendance page* yang menampilkan *grid* kalender.



Gambar 18. Attendance Page

11. Assignment Page

Gambar dibawah merupakan *wireframe assignment pag* dari *project manager* yang berisi *list* dari tugas atau project yang sedang dikerjakan.



Gambar 19. Assignment Page

12. Add Assignment Modal

Gambar dibawah merupakan *wireframe add assignment page* dari *project manager* yang berisi form.

The image shows a wireframe of a web application interface. On the left is a sidebar with navigation items: Dashboard, Team Management, and Assignment. The main content area features a modal window titled "Add Assignment". The modal contains the following form elements: "Assignment Title" (input field), "Start Date" (input field), "End Date" (input field), and "Team of Assignment" (dropdown menu). Below these is a text area for "Assignment Description Overview". At the bottom of the modal is an "Assignment To Do" section with a "+ Add To Do" button and a message "Assignment To Do Not Added Yet". A "+ Submit" button is located at the bottom right of the modal. The background shows a blurred dashboard with a search bar, "Export Data" and "Filter Assignment" buttons, and a list of task cards.

Gambar 20. Add Assignment Modal

13. Add Assignment Todo Modal

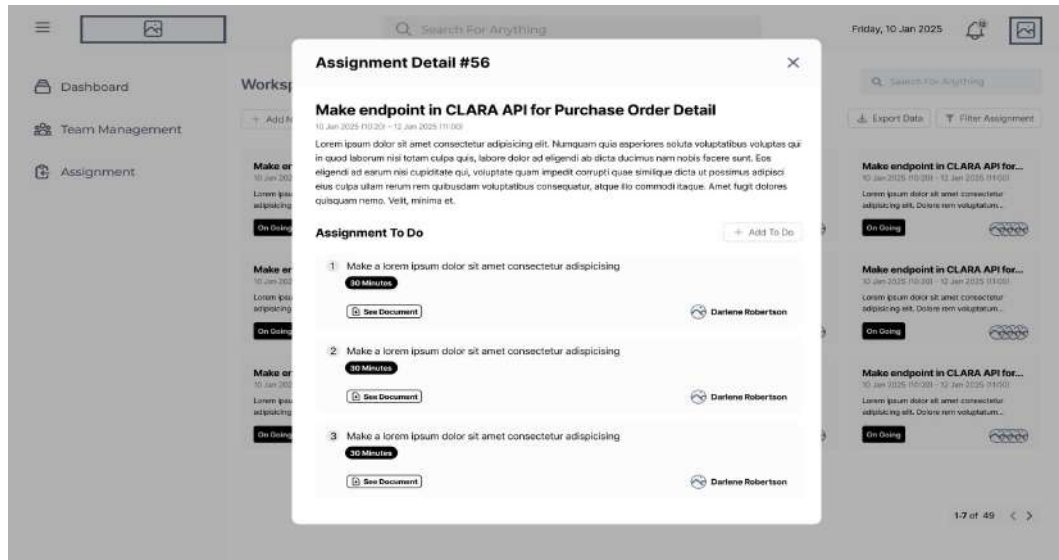
Gambar dibawah merupakan *wireframe add assignment todo modal* dari *project manager* yang berisi form.

The image shows a wireframe of a web application interface, similar to the previous one. It features the same sidebar and background dashboard. The main focus is a modal window titled "Add Assignment" which has a smaller modal nested inside it titled "Add Assignment Todo". The "Add Assignment Todo" modal contains: "Assignment" (input field), "Expected Minutes" (input field), and "Document" (text area). It includes a "+ Assign User" button and a "+ Submit" button. The background "Add Assignment" modal is partially visible, showing its "Assignment Title", "Start Date", "End Date", "Team of Assignment", and "Assignment Description Overview" fields.

Gambar 21. Add Assignment Todo Modal

14. Assignment Detail Modal

Gambar dibawah merupakan *wireframe assignment detail modal* dari *project manager* yang berisi detail.



Gambar 22. Assignment Detail Modal

3.3. Mekanisme Alur Kerja Pengguna

3.3.1. Absensi

Pada proses absensi, pengguna yang diharuskan melakukan absensi hanya dapat mengakses fitur presensi melalui aplikasi dalam rentang waktu pukul 08.00–09.15 WIB. Di luar waktu tersebut, fitur absensi dinonaktifkan dan pengguna tidak dapat melakukan absensi dan tidak dapat memasuki aplikasi. Untuk melakukan absensi, pengguna diwajibkan mengambil foto langsung melalui kamera perangkat pada halaman yang telah ditentukan dalam aplikasi. Foto harus menampilkan pengguna bersama perangkat laptop yang digunakan, sebagai bukti bahwa kehadiran dilakukan di lingkungan kantor. Sistem dirancang sedemikian rupa sehingga pengguna tidak dapat memilih atau mengunggah gambar dari galeri perangkat. Pembatasan ini diterapkan untuk mencegah potensi kecurangan, seperti penggunaan foto lama atau foto yang tidak mencerminkan kondisi kehadiran sebenarnya pada saat absensi dilakukan.

3.3.2. Penugasan

Pada proses penugasan, manajer proyek terlebih dahulu menetapkan tugas melalui halaman Tasks dalam aplikasi. Tugas tersebut kemudian dapat diakses oleh karyawan dan peserta magang melalui halaman yang sama. Selain itu, notifikasi sistem dikirimkan secara otomatis kepada pengguna yang bersangkutan sebagai pemberitahuan atas tugas baru yang telah diberikan. Saat memulai pengerjaan tugas, pengguna dapat mengaktifkan fitur *timelog* untuk mencatat durasi pengerjaan secara otomatis. Fitur ini merekam waktu mulai pengerjaan dan terus berjalan selama tugas dalam proses pengerjaan. Setelah tugas selesai dikerjakan, pengguna menonaktifkan *timelog* dan mengisi keterangan singkat mengenai hasil atau status penyelesaian tugas tersebut. Data dari *timelog* dan keterangan tugas kemudian disimpan sebagai bagian dari laporan kinerja pengguna.

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Hasil Implementasi

Tahap implementasi sistem dimulai dengan pengembangan komponen backend dan struktur basis data terlebih dahulu, sebelum beralih ke pengembangan antarmuka pengguna (frontend). Basis data aplikasi dirancang dan diimplementasikan secara otomatis melalui Sequelize CLI, sebuah ORM (Object-Relational Mapping) yang diintegrasikan dengan framework NestJS. Dalam proses ini, struktur basis data tidak dibuat secara manual melalui antarmuka database, melainkan didefinisikan melalui serangkaian file migration dan model. File migration berfungsi untuk mendefinisikan skema tabel seperti kolom, tipe data, kunci primer, dan relasi. Sementara itu, file model merepresentasikan entitas dalam sistem dan menangani logika interaksi antara aplikasi dan tabel database.

Selain itu, arsitektur backend mengikuti pola modular NestJS yang terstruktur, terdiri atas Controller, Service, dan Module. Controller bertanggung jawab atas penanganan permintaan HTTP dan respons API. Service mengelola logika bisnis dan operasi data (termasuk pemanggilan model Sequelize), sedangkan Module mengorganisasi dan menghubungkan komponen-komponen tersebut secara terenkapsulasi. Setelah seluruh migration dan model didefinisikan, struktur basis data fisik dihasilkan melalui perintah terminal `npx sequelize db:migrate`, yang mengeksekusi semua file migrasi dan membuat tabel-tabel sesuai spesifikasi ERD yang telah dirancang pada tahap desain.

Kemudian, setelah seluruh endpoint API berhasil diuji dan divalidasi melalui Postman sebagai bagian dari verifikasi komponen backend pengembangan dilanjutkan ke tahap frontend. Pada tahap ini, antarmuka pengguna dibangun berdasarkan wireframe dan desain UI yang telah disusun sebelumnya, dengan memastikan keselarasan antara tampilan visual, alur interaksi, dan kebutuhan fungsional yang telah ditetapkan. Frontend dikembangkan menggunakan Angular, sebuah framework JavaScript berbasis TypeScript. Untuk kebutuhan desain visual dan responsivitas, digunakan Tailwind CSS sebuah utility-first CSS framework

yang memungkinkan pengembangan antarmuka yang konsisten, ringan, dan mudah dikustomisasi tanpa perlu menulis CSS dari nol. Selain itu, untuk mempercepat implementasi komponen UI yang kompleks (seperti tabel data, kalender, dialog, dan form control), diintegrasikan pula pustaka PrimeNG, yaitu kumpulan komponen UI siap pakai yang dapat digunakan dengan Angular. Integrasi frontend dengan backend dilakukan melalui pemanggilan HTTP request ke endpoint API yang telah diuji sebelumnya. Setiap interaksi pengguna seperti login, pengisian formulir, atau penampilan data dinamis diterjemahkan menjadi permintaan ke server, yang kemudian merespons dengan data dalam format JSON.

4.2. Pengujian *User Acceptance Testing* (UAT)

Tabel 20. Hasil *Blackbox Testing* Manajer

| Skenario Pengujian | Ekspektasi Sistem | Hasil Sistem | Status |
|---|--|---|-------------|
| Manajer menginput username dan password dengan benar | Menuju halaman dashboard | Menampilkan halaman dashboard | Valid |
| Manajer menginput username dan password dengan salah | Memunculkan pesan <i>please login using your valid credential</i> | Muncul pesan <i>please login using your valid credential</i> | Tidak Valid |
| Manajer menginput <i>form add user</i> dan menekan tombol <i>submit</i> | Menutup form, memunculkan pesan <i>successfully add user</i> dan user baru bertambah | Menutup form, muncul pesan <i>successfully add user</i> dan user baru bertambah | Valid |
| Manajer menekan salah satu list nama pengguna | Membuka halaman informasi <i>detail</i> pengguna | Menampilkan halaman informasi detail pengguna | Valid |

| | | | |
|---|---|---|-------|
| Manajer menekan tombol edit dan mengubah data kemudian menekan tombol <i>save changes</i> | Membuka <i>form edit</i> dan menyimpan perubahan | Membuka <i>form edit</i> dan menyimpan perubahan | Valid |
| Manajer mencari nama pengguna di input cari pengguna | Memunculkan data pengguna berdasarkan pencarian | Memunculkan data pengguna berdasarkan pencarian | Valid |
| Manajer menyeret box nama dari <i>division lists</i> | Memunculkan konfirmasi untuk melanjutkan aksi | Muncul konfirmasi untuk melanjutkan aksi | Valid |
| Manajer menekan tombol show calendar pada menu attendance dan kemudian hide calendar | Memunculkan halaman tanggal <i>attendance</i> dan menghilangkan halaman | Memunculkan halaman tanggal <i>attendance</i> dan menghilangkan halaman | Valid |
| Manajer menekan tombol <i>pin calendar</i> | Halaman di pin dan tidak dapat ditutup | Halaman di pin dan tidak dapat ditutup | Valid |
| Manajer menekan tombol <i>today</i> | Mengarahkan tanggal ke hari pada saat itu | Menampilkan tanggal ke hari pada saat itu | Valid |
| Menekan salah satu tanggal di <i>calendar</i> | Memunculkan data <i>attendance</i> pada sesuai tanggal yang di pilih | Memunculkan data <i>attendance</i> pada sesuai tanggal yang di pilih | Valid |
| Mencari nama dari pengguna di pencarian attendance | Memunculkan data <i>attendance</i> pengguna berdasarkan pencarian | Memunculkan data <i>attendance</i> pengguna berdasarkan pencarian | Valid |

Tabel 21. Hasil *Blackbox Testing* Manajer Proyek

| Skenario Pengujian | Ekspektasi Sistem | Hasil Sistem | Status |
|---|--|--|---------------|
| Manajer Proyek menginput username dan password dengan benar | Menuju halaman dashboard | Menampilkan halaman dashboard | Valid |
| Manajer Proyek menginput username dan password dengan salah | Memunculkan pesan <i>please login using your valid credential</i> | Muncul pesan <i>please login using your valid credential</i> | Tidak Valid |
| Manajer Proyek menekan tombol <i>new assignment</i> | Membuka modal <i>add assignment</i> | Menampilkan modal <i>add assignment</i> dan | Valid |
| Manajer Proyek mengisi <i>form</i> dengan sesuai kemudian menekan tombol <i>create assignment</i> | Menyimpan data <i>assignment</i> dan menutup modal <i>add assignment</i> | Menyimpan data <i>assignment</i> dan menutup modal <i>add assignment</i> | Valid |
| Manajer Proyek menekan <i>card assignment</i> | Membuka modal <i>edit assignment</i> | Menampilkan modal <i>edit assignment</i> | Valid |
| Manajer Proyek menekan tombol ikon pensil | Menyediakan <i>Form</i> untuk perubahan data | Memunculkan input <i>Form</i> untuk perubahan data | Valid |
| Manajer Proyek menekan tombol <i>save</i> | Menyimpan perubahan | Menyimpan perubahan | Valid |

Tabel 22. Hasil *Blackbox Testing* Karyawan dan Peserta Magang

| Skenario Pengujian | Ekspektasi Sistem | Hasil Sistem | Status |
|---------------------------|--------------------------|---------------------|---------------|
|---------------------------|--------------------------|---------------------|---------------|

| | | | |
|--|---|---|-------------|
| Karyawan dan Peserta Magang menginput username dan password dengan benar | Menuju halaman dashboard, apabila belum presensi maka menuju halaman presensi | menampilkan halaman dashboard, apabila belum presensi maka menampilkan halaman presensi | Valid |
| Karyawan dan Peserta Magang menekan tombol foto <i>dan submit attendance</i> | Menyimpan data presensi dan mengarahkan ke <i>dashboard</i> | Menyimpan data presensi dan mengarahkan ke dashboard | Valid |
| Karyawan dan Peserta Magang menginput username dan password dengan salah | Memunculkan pesan <i>please login using your valid credential</i> | Muncul pesan <i>please login using your valid credential</i> | Tidak Valid |
| Karyawan dan Peserta Magang menekan <i>card assignment</i> | Membuka modal <i>view assignment</i> | Menampilkan modal view assignment | Valid |
| Karyawan dan Peserta Magang menekan tombol <i>start timelog</i> | Memulai timelog | Memulai timelog | Valid |
| Karyawan dan Peserta Magang menekan tombol <i>stop timelog</i> | Menghentikan timelog dan memunculkan modal <i>stop timelog</i> | Menghentikan timelog dan memunculkan modal <i>stop timelog</i> | Valid |

BAB V KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa proses pencatatan kehadiran dan pelaporan kinerja di PT. Mitra Integritas Indonesia yang sebelumnya bersifat manual dan terdesentralisasi melalui Slack serta dokumen Microsoft Word telah berhasil diintegrasikan ke dalam sebuah Sistem Informasi Absensi dan Pelaporan Kinerja Karyawan Berbasis Website. Sistem yang dihasilkan mampu memenuhi kebutuhan fungsional dan non-fungsional yang telah ditetapkan, termasuk dukungan terhadap empat peran pengguna (Manajer, Project Manager, Karyawan, dan Peserta Magang), kontrol akses berbasis peran, pencatatan absensi berbasis unggah foto, serta pembuatan laporan kerja dalam satu platform terpadu. Pengujian blackbox yang dilakukan terhadap seluruh skenario penggunaan menunjukkan bahwa sistem berfungsi sesuai spesifikasi—memberikan respons yang akurat, menolak akses tidak sah, dan menyajikan data secara real-time.

5.2 Saran

Berdasarkan sistem yang telah dibuat, masih terdapat hal-hal yang dapat ditingkatkan untuk meningkatkan kecanggihan sistem. Salah satunya adalah menambahkan mekanisme validasi tambahan pada proses absensi berbasis foto. Pertama, sistem dapat dilengkapi dengan validasi lokasi geografis (geolocation) saat pengguna mengunggah foto kehadiran, sehingga memastikan bahwa absensi dilakukan dari lokasi kerja yang sah atau dalam radius yang ditentukan. Kedua, sistem dapat menerapkan validasi format dan metadata foto, seperti memastikan foto diambil secara langsung melalui kamera (bukan dipilih dari galeri), memverifikasi timestamp eksif, atau memeriksa resolusi minimum guna mencegah penggunaan gambar yang tidak valid atau dimanipulasi.

DAFTAR PUSTAKA

- Ahmad Taufik, Gunawan Sudarsono, Budiyantara, I Ketut Sudaryana, & Tupan Tri Muryono. (2022). Pengantar Teknologi Informasi. *Yayasan Drestanta Pelita Indonesia*, 6(1), 1–113.
- Cherny, B. (2019). *Programming TypeScript* (1st ed.). O'Reilly Media.
- Nayrolles, M., Gunasundaram, R., Rao, S. (2017). *Mastering Angular*. Britania Raya: Packt Publishing, Limited.
- Jonna, S., & Varaksin, O. (2017). *Angular UI Development with PrimeNG*. Packt Publishing Ltd.
- Pham, A. D. (2020). *Developing back-end of a web application with NestJS framework: Case - Integrify Oy's student management system* (Bachelor's thesis, Bachelor of Business Administration).
- Dyer, R. J. T. (2015). *Learning MySQL and MariaDB: Heading in the right direction with MySQL and MariaDB*. O'Reilly Media.
- Kujala, A. (2023). *Development of a modern full stack web application* (Bachelor's thesis). Turku University of Applied Sciences.
- Ivanova, V., Boneva, A., & Vasilev, P. (2024). Unified modelling language application for laparoscopic instrument design. *International Journal of Bioautomation*, 28(3).
- Maulani, G., Santoso, L. H., Ibrahim, A., Chafid, N., Hidayat, A., Mustakim, Fitriyadi, F., Anyan, Soleh, O., & Suparman, A. (2024). *Rekayasa Perangkat Lunak*. Paput Tri Cahyono (Ed.). Yayasan Cendekia Mulia Mandiri. ISBN: 978-623-8382-90-3.
- Tiun, S., Mokhtar, U.A., Bakar, S.H., & Saad, S. (2020). Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text. *Journal of Physics: Conference Series*, 1529.
- Mulyani, S. (2016). *Metode analisis dan perancangan sistem* (2nd ed.). Abdi Sistematika.
- BUKU AJAR KONSEP SISTEM INFORMASI. (2020). (n.p.): SCOPINDO MEDIA PUSTAKA.
- Lean UX: Pemahaman dan Penerapan Metodologi Desain UI yang Efektif. (n.d.). (n.p.): Jejak Pusta

- Wowor, G. M. (2021). *Aplikasi pelaporan kinerja pegawai berbasis web (Studi kasus: Dinas Pendidikan dan Kebudayaan Daerah Kota Tomohon)*.
- Rahmawati, R. (2018). *Sistem pelaporan kinerja pegawai berbasis web pada Kementerian Agama Kabupaten Luwu Utara*.
- Valerian, M. E. D., & Dahlia. (2024). *Analisa penerapan sistem absensi pegawai berbasis website pada DPRD Provinsi DKI Jakarta*.
- Larasati, P. D., & Sa'ba, N. F. (2021). *Perancangan sistem pelaporan absensi berbasis web pada PT. Solar Control Specialist*

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement

Tujuan pengumpulan *requirement* adalah untuk memahami kebutuhan pengguna dan memastikan bahwa sistem yang dikembangkan dapat memenuhi tujuan bisnis serta memberikan nilai tambah bagi organisasi. Berikut beberapa poin mengenai tujuan pengumpulan *requirement*

1. Memahami alur kerja saat ini dalam pelaporan kinerja harian dan mingguan karyawan lapangan.
2. Mengidentifikasi kebutuhan fitur utama sistem pelaporan kinerja berbasis web.
3. Menentukan indikator kinerja (KPI) yang relevan: kehadiran, tugas selesai, tugas tertunda, dll.
4. Mengumpulkan masukan terkait antarmuka pengguna (UI/UX) dan aksesibilitas.

Tabel 23. Tabel lampiran requirement

| ITEM | DESKRIPSI |
|------------------|--|
| Tanggal Kejadian | 7 Oktober 2024 |
| Lokasi | PT Mitra Integritas Indonesia Lantai 3 |
| Metode | Diskusi dan Wawancara singkat |
| Peserta | Penulis dan Mentor |

Tabel 24. Tabel lampiran pertanyaan

| No | Pertanyaan | Tujuan Pertanyaan |
|----|--|--|
| 1 | Bagaimana alur pelaporan kehadiran dan penyelesaian tugas karyawan saat ini? | Untuk mengetahui alur manual yang berjalan saat ini. |
| 2 | Apa saja kendala yang sering muncul | Untuk mengidentifikasi |

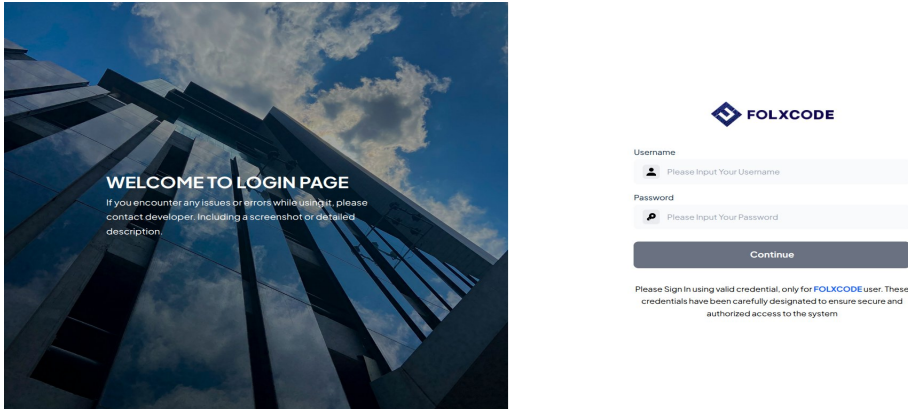
| | | |
|---|--|---|
| | dalam proses pelaporan kinerja (kehadiran & tugas) secara manual? | permasalahan pada sistem lama. |
| 3 | Siapa saja pihak yang terlibat dalam proses pelaporan dan verifikasi kinerja karyawan? | Untuk mengetahui aktor sistem yang relevan (misal: karyawan, manajer, admin). |
| 4 | Bagaimana data kehadiran dan tugas dikumpulkan, disimpan, dan direkap setelah periode pelaporan (harian/mingguan)? | Untuk memahami alur penyimpanan dan rekap data. |
| 5 | Apakah ada kebutuhan untuk memantau kinerja karyawan secara real-time atau hanya ringkasan mingguan? | Untuk menentukan cakupan fitur dashboard dan frekuensi pembaruan data. |

Lampiran B. Link Produk

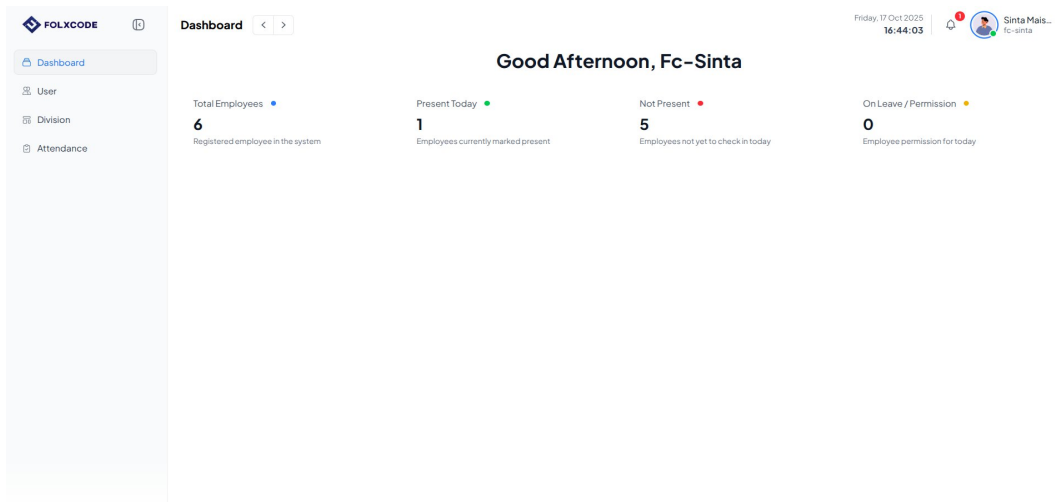
https://gitlab.com/Hugo_Hamonangan103/folxcode-attendance-task-frontend/-/tree/c021413d950fc8f873589812af993f1445ef0bfb

https://gitlab.com/Hugo_Hamonangan103/folxcode-attendance-task-backend/-/tree/master?ref_type=heads

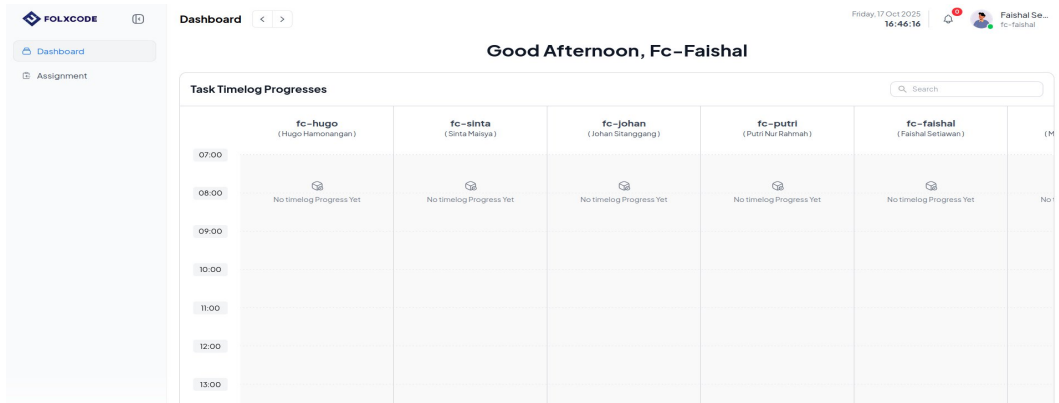
Lampiran C. Dokumen Pengujian UAT



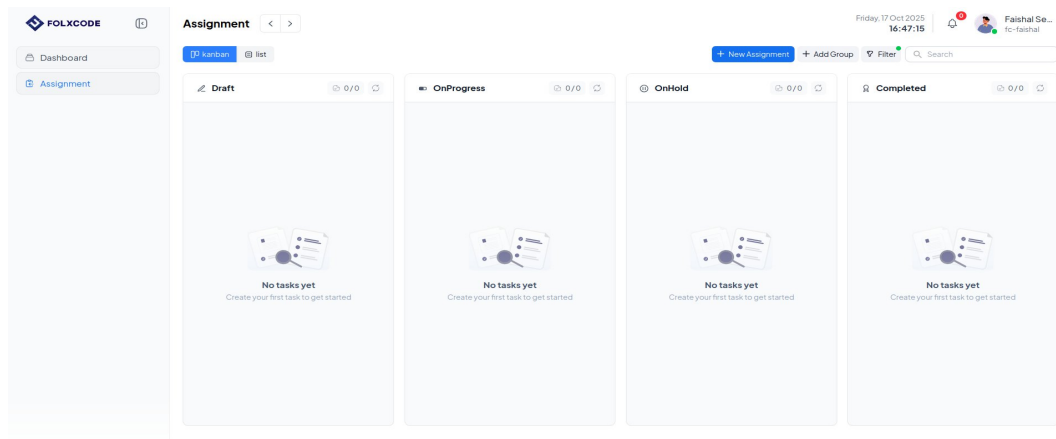
Gambar 23. Lampiran halaman Login



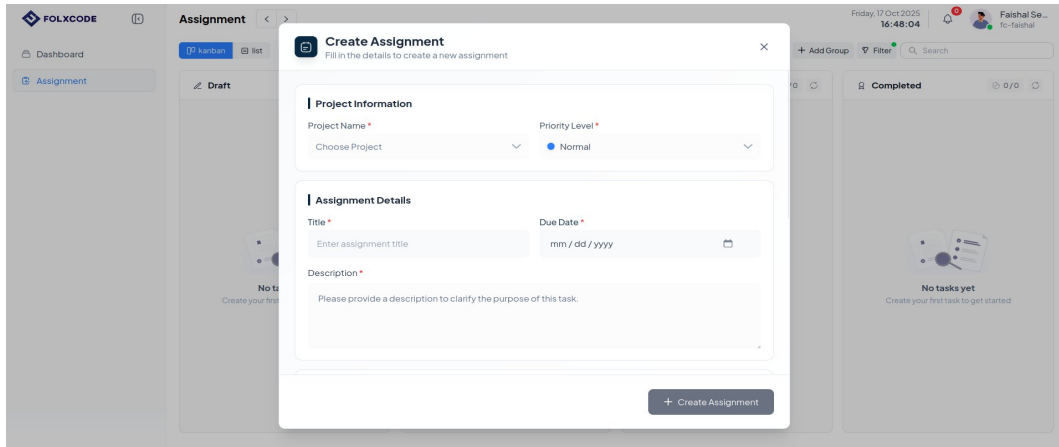
Gambar 24. Lampiran halaman Dashboard Manajer



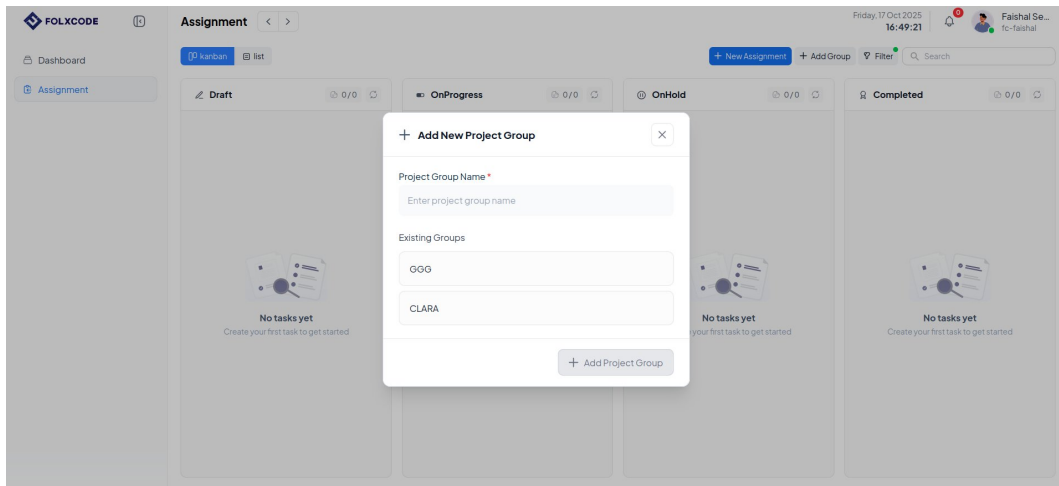
Gambar 25. Lampiran halaman *Dashboard* Manajer Proyek



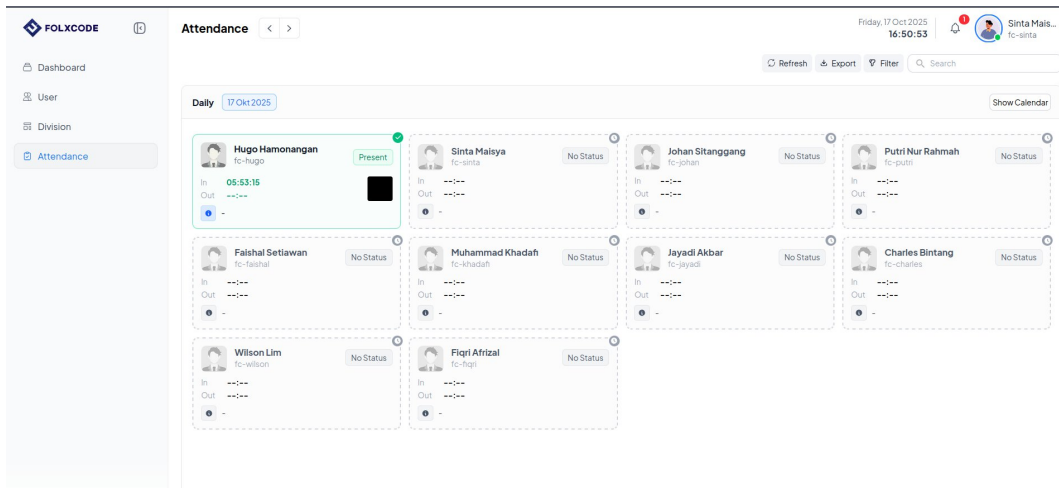
Gambar 26. Lampiran halaman *Assignment*



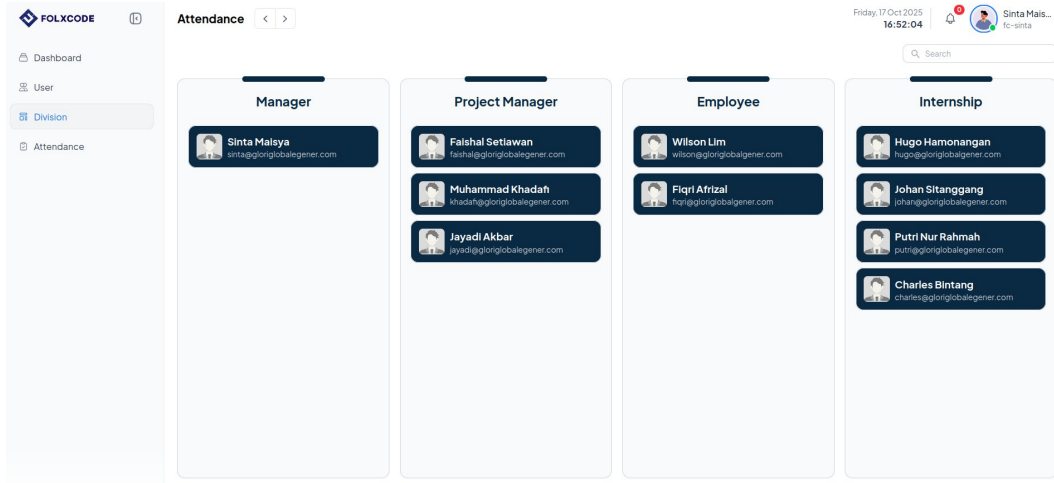
Gambar 27. Lampiran halaman Add assignment



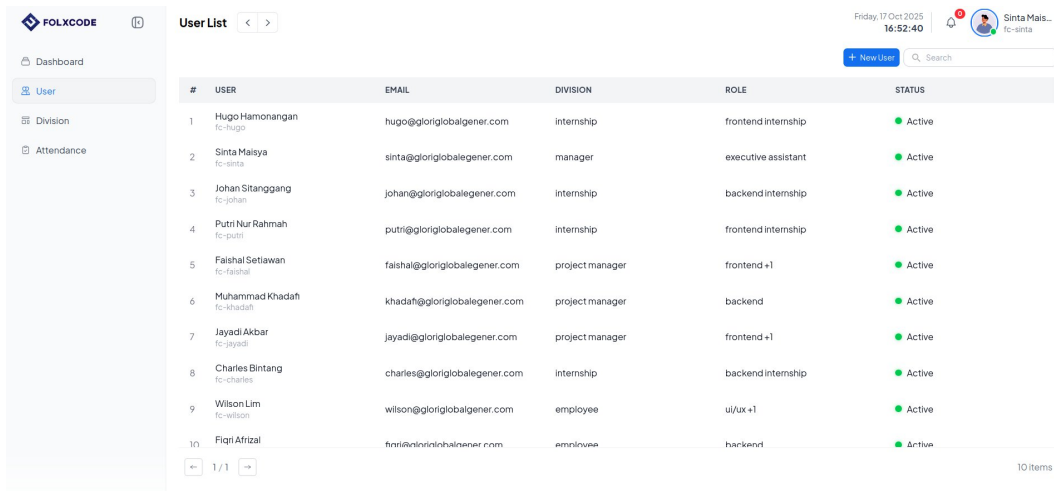
Gambar 28. Lampiran halaman Add Project Groups



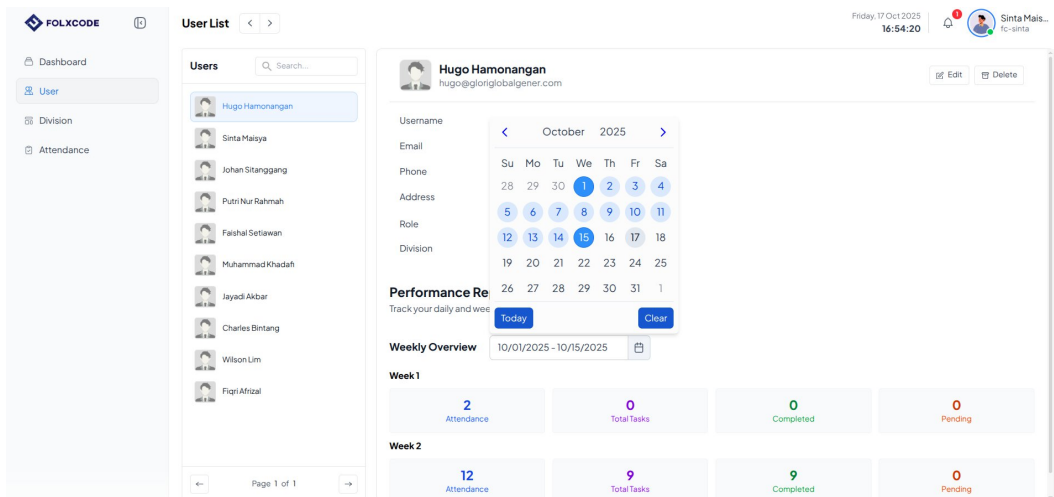
Gambar 29. Lampiran halaman Attendance



Gambar 30. Lampiran halaman Division



Gambar 31. Lampiran halaman User



Gambar 32. Lampiran halaman User Detail