**LAPORAN**

**Implementasi dan Dokumentasi**

**“Project sistem manajemen karyawan”**

Dosen Pengampu:

****

Disusun oleh:

**KELOMPOK 2**

|  |  |
| --- | --- |
| Darean Ahmad Raffi | F55124086 |
| Farel Grady Toding | F55124099 |
| Fauzan Abdi Mudzacky | F55124118 |
| Dini Zahra | F55124113 |
| Claudya Christy Koloay | F55124116 |

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TADULAKO**

**2025**

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Perkembangan teknologi informasi saat ini sangat membantu berbagai bidang pekerjaan, termasuk dalam pengelolaan data karyawan di perusahaan atau instansi. Pengelolaan data karyawan yang masih dilakukan secara manual sering menimbulkan berbagai permasalahan, seperti pencarian data yang lambat, risiko kesalahan pencatatan, serta kesulitan dalam pembuatan laporan absensi dan penggajian.

Manajemen karyawan sendiri tidak hanya berkaitan dengan pendataan identitas, tetapi juga mencakup pengelolaan jabatan, departemen, absensi, dan payroll. Jika proses tersebut tidak didukung oleh sistem yang baik, maka dapat menghambat kinerja dan efisiensi kerja. Oleh karena itu, dibutuhkan sebuah sistem informasi yang mampu mengelola data karyawan secara terstruktur dan terintegrasi.

Berdasarkan hal tersebut, kelompok kami mengembangkan Sistem Informasi Manajemen Karyawan berbasis web yang bertujuan untuk membantu proses administrasi karyawan agar lebih cepat, rapi, dan aman. Sistem ini dilengkapi dengan fitur pengelolaan data karyawan, absensi, payroll, serta profil karyawan.

**1.2 Rumusan Masalah**

1. Bagaimana perancangan dan pembuatan Sistem Informasi Manajemen Karyawan berbasis web?
2. Bagaimana proses pengelolaan data karyawan, absensi, dan payroll dalam sistem?
3. Bagaimana penerapan autentikasi dan hak akses antara admin dan karyawan?
4. Bagaimana proses penyimpanan source code dan dokumentasi aplikasi melalui repository GitHub?
5. Bagaimana proses pengujian sistem pada sisi backend dan frontend?

**1.3 Tujuan**

1. Membuat Sistem Informasi Manajemen Karyawan berbasis web.
2. Mengelola data karyawan, absensi, dan payroll secara terkomputerisasi.
3. Menerapkan sistem login dan pembagian hak akses pengguna.
4. Menyediakan dokumentasi penggunaan sistem melalui file README.md di GitHub.
5. Melakukan pengujian sistem untuk memastikan aplikasi berjalan dengan baik.

**1.4 Manfaat**

1. Mempermudah pengelolaan data karyawan secara cepat dan terstruktur.  
   Mengurangi kesalahan dalam pencatatan data karyawan, absensi, dan gaji.
2. Mempercepat proses pembuatan laporan oleh admin.
3. Memberikan kemudahan bagi karyawan dalam mengelola data profil dan melihat absensi serta payroll.
4. Menjadi sarana pembelajaran dalam pengembangan sistem informasi berbasis web.

**1.5 Ruang Lingkup**

1. Manajemen Data Karyawan
2. Manajemen Absensi
3. Manajemen Divisi / Departemen
4. Autentikasi & Hak Akses Pengguna
5. Laporan Karyawan
6. Antarmuka Pengguna (UI/UX)
7. Batasan Sistem

**BAB II**

**DESKRIPSI DAN ANALISIS SISTEM**

**2.1 Deskripsi Umum Aplikasi**

**2.1.1 Nama Aplikasi** Aplikasi ini diberi nama “Sistem Manajemen Karyawan”, yaitu sebuah platform digital yang dikembangkan untuk mendukung kegiatan operasional perusahaan dalam mengelola informasi karyawan secara terintegrasi. Sistem ini dirancang untuk menjadi pusat data yang menyimpan seluruh informasi penting terkait karyawan, mulai dari biodata, jabatan, status pekerjaan, riwayat penilaian, hingga aktivitas operasional sehari-hari.

Pengembangan aplikasi ini bertujuan untuk menggantikan proses manual yang sering memakan waktu, berpotensi menimbulkan kesalahan pencatatan, dan menyulitkan proses pencarian data. Dengan memanfaatkan aplikasi ini, perusahaan dapat memastikan bahwa manajemen data dilakukan secara lebih akurat, cepat, dan aman. Proses seperti input data karyawan, pembaruan informasi, pengecekan riwayat kehadiran, hingga pengelolaan akses pengguna dapat dilakukan hanya melalui beberapa langkah sederhana.

**2.1.2 Deskripsi Aplikasi**

Aplikasi Sistem Manajemen Karyawan adalah sebuah platform yang dirancang untuk membantu perusahaan dalam mengelola data dan aktivitas karyawan secara lebih efisien, terstruktur, dan terpusat. Sistem ini menyediakan berbagai fitur utama seperti pengelolaan data karyawan, pencatatan absensi, pengaturan jadwal kerja, manajemen cuti, hingga pelaporan kinerja. Dengan adanya aplikasi ini, proses administrasi yang sebelumnya dilakukan secara manual dapat dilakukan dengan lebih cepat dan akurat.

Aplikasi ini mengintegrasikan fungsi antara pihak karyawan dan admin. Karyawan dapat melakukan absensi, melihat jadwal kerja, mengajukan cuti, dan memperbarui data pribadi secara mandiri. Sementara itu, admin atau HR dapat mengelola seluruh data karyawan, memantau kehadiran secara real-time, menyetujui atau menolak pengajuan cuti, serta menghasilkan laporan bulanan maupun tahunan.

Dengan memanfaatkan teknologi berbasis web, aplikasi ini dapat diakses kapan saja melalui komputer maupun perangkat mobile, sehingga mendukung fleksibilitas kerja. Selain itu, sistem ini dibangun dengan memperhatikan aspek keamanan data, sehingga seluruh informasi karyawan tersimpan secara aman dan hanya dapat diakses oleh pengguna yang berwenang.Secara keseluruhan, aplikasi ini bertujuan untuk meningkatkan produktivitas perusahaan, mempermudah proses administrasi, serta memberikan transparansi antara karyawan dan pihak manajemen.

**2.1.3 Fungsi Utama Aplikasi**

Aplikasi ini memiliki fungsi utama untuk:

1. Mengelola data karyawan, meliputi pencatatan identitas, jabatan, divisi, dan status kepegawaian.
2. Menyediakan fitur tambah, edit, hapus, dan pencarian data karyawan.
3. Menyediakan tampilan informasi karyawan yang dapat dilihat oleh karyawan sendiri.
4. Mengelompokkan karyawan berdasarkan jabatan atau divisi.
5. Menghasilkan laporan sederhana terkait informasi data karyawan.
6. Menyediakan sistem login untuk membedakan hak akses antara karyawan dan admin.

**2.1.4 Target Pengguna**

Aplikasi ini digunakan oleh dua jenis pengguna, yaitu:

1. **Admin**, yang bertanggung jawab dalam mengelola seluruh data karyawan, melakukan perubahan data, serta mengakses seluruh fitur aplikasi.
2. **Karyawan**, yang dapat melihat profil pribadi, informasi jabatan, dan data kepegawaian lainnya sesuai hak akses yang diberikan.
   1. **Spesifikasi Teknis**

**2.1.2 Arsitektur Sistem**

Berikut Adalah bagian dari arsitektur sistem:

1. Aplikasi memakai arsitektur Client–Server: frontend (client) berinteraksi dengan backend (server API).
2. Sistem bersifat full-stack web application dengan pemisahan antara frontend dan backend.
3. Pola komunikasi antara frontend dan backend kemungkinan menggunakan REST API (karena backend berperan sebagai API server).

**2.1.3 Teknologi yang Digunakan**

|  |  |
| --- | --- |
| **Bagian** | **Teknologi/stack** |
| **Frontend** | Vite + React 19, Tailwind CSS, React-Query (@tanstack/react-query), axios, dayjs, lucide-react |
| **Backend** | Node.js 20, Express 5, ORM Prisma, database SQLite (default). |
| **Database** | SQLite (default) via Prisma ORM |
| **Autentikasi & Keamanan** | JSON Web Token (JWT) untuk login/auth, bcrypt untuk hashing password, middleware guard role (requireAuth, requireRole) |

**BAB III**

**ANALISIS DAN PENGUJIAN SISTEM**

**2.3 Analisis sistem**

Analisis sistem dilakukan untuk mengetahui sejauh mana sistem yang dibangun dapat berjalan sesuai dengan kebutuhan fungsional yang telah dirancang. Pada tahap ini dilakukan pengujian terhadap sistem dari sisi backend dan frontend, serta pengujian terhadap kondisi ekstrem (*edge cases*) untuk melihat ketahanan sistem terhadap kesalahan input maupun penyalahgunaan akses. Hasil dari analisis ini menjadi acuan untuk menentukan apakah sistem sudah layak digunakan atau masih memerlukan perbaikan.

**2.3.1 Pengujian Backend (Postman / Thunder Client)**

Pengujian backend adalah proses pemeriksaan pada bagian server aplikasi yang meliputi database, API, dan logika bisnis untuk memastikan seluruh proses berjalan dengan benar, aman, dan stabil. Tujuan utama pengujian ini adalah menjaga keakuratan data, memastikan logika sistem sesuai kebutuhan, menguji keamanan akses, serta menilai kinerja dan integrasi antar komponen backend agar aplikasi dapat berfungsi optimal tanpa

**Tabel Pengujian**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Endpoint** | **Method** | **Tujuan Pengujian** | **Input/Request** | **Output/Respons** | **Status** | **Kesimpulan** |
| **1.** | /api/auth/login | POST | Menguji autentikasi menggunakan email & password. | Body JSON: { "email": "admin@example.com", "password": "PasswordRahasia123" } | 200 OK + token JWT + informasi user. | ✅ | Backend berhasil memverifikasi login dan menghasilkan token JWT. |
| 2. | /api/employees | GET | Menguji endpoint terproteksi dengan token valid. | Header: Authorization: Bearer <token> | 200 OK + list data karyawan (id, code, name, email, dsb). | ✅ | Endpoint hanya dapat diakses jika token valid. |
| 3. | /api/employees | GET | Menguji akses tanpa token atau token salah. | Header: Authorization: Bearer token-salah | 401 Unauthorized | ✅ | Middleware menolak request tanpa kredensial valid. |
| 4. | /api/profile | GET | Melihat data profil user dari token login. | Header: Authorization: Bearer <token> | 200 OK + data profil user | ✅ | Berfungsi berdasarkan token user. |
| 5. | /api/payroll | GET | Menguji akses modul payroll. | Header token | Data payroll (JSON) | ✅ | Modul payroll berhasil diakses melalui token. |
| 6. | /api/attendance | POST/GET | Menguji absensi karyawan. | Token + data absensi | Response JSON | ✅ | Fitur absensi berjalan sesuai proses bisnis. |

**Keterangan:**

✅ Berhasil

**2.3.2 Pengujian Frontend**

Pengujian frontend adalah proses memastikan bahwa tampilan antarmuka pengguna (UI) dan segala interaksi yang terjadi di sisi klien berfungsi dengan baik, tampil konsisten, serta memberikan pengalaman pengguna yang optimal sesuai desain dan kebutuhan sistem. Pengujian ini mencakup pengecekan elemen visual, responsivitas layout, navigasi, validasi input, kompatibilitas pada berbagai perangkat dan browser, serta performa saat pengguna berinteraksi dengan aplikasi. Tujuan utamanya adalah memastikan bahwa pengguna dapat menggunakan aplikasi dengan mudah, tanpa error, bebas bug visual maupun fungsional, serta mendapatkan pengalaman yang stabil dan nyaman.

**TabelPengujian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Kategori** | **Skenario Pengujian** | **Langkah-Langkah Pengujian** | **Hasil Yang Diharapkan** | **Status** |
| 1. | Autentikasi | Login Admin & Validasi Navigasi | 1. Login menggunakan akun Admin. 2. Periksa menunavigasi yang tampil. | Menu navigasi menampilkan tab lengkap: Dashboard, Karyawan, Absensi, Penggajian. | ✅PASS |
| 2. | Autentikasi | Login Karyawan & Validasi Navigasi | 1. Login akun karyawan (user: Mamank). 2. Periksa menunavigasi. | Menu navigasi menyesuaikan hak akses (Hanya muncul: Absensi, Penggajian, Profil). Dashboard admin tidak terlihat. | ✅PASS |
| 3. | Dashboard | Monitoring KPI & Grafik (Refresh Query) | 1. Cek KPI awal di Dashboard (Total: 4). 2. Tambahdata/karyawan baru. 3. Refresh Dashboard. | Angka pada KPI dan grafik dashboard berubah real-time (Total naik menjadi 6) mengikuti data baru. | ✅PASS |
| 4. | Karyawan | Tambah Data Karyawan (Admin) | 1. Login sebagai Admin → Menu Karyawan. | Form tambah karyawan berfungsi. Data baru muncul di tabel (contoh: Mamank, Ijat, dll.) dan status “ACTIVE”. | ✅PASS |

**2.3.3 Pengujian Edge Cases**

Pengujian *edge cases* merupakan pengujian terhadap kondisi ekstrem atau tidak normal yang kemungkinan besar dapat terjadi dalam penggunaan sistem. Pengujian ini bertujuan untuk memastikan bahwa sistem mampu memberikan respon yang tepat ketika menerima input yang salah atau akses yang tidak sesuai. Adapun beberapa pengujian edge case yang telah dilakukan adalah sebagai berikut:

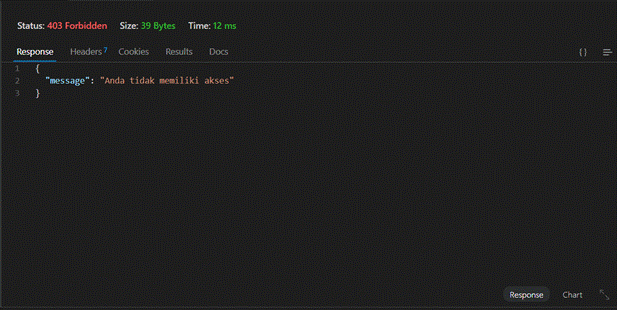
**Tabel Pengujian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **Skenario Pengujian** | **Langkah Pengujian** | **Hasil Diharapkan** | **Hasil Aktual** | **Status** |
| 1. | Akses endpoint admin tanpa token | Mengakses endpoint admin tanpa Authorization Header | Sistem menolak akses (401 Unauthorized) | Sistem mengembalikan status 401 Unauthorized | ✅ Pass |
| 2. | akses endpoint admin dengan token karyawan | Login sebagai karyawan → dapat token → akses endpoint admin | Sistem menolak akses (403 Forbidden) | Sistem mengembalikan status 403 Forbidden | ✅ Pass |
| 3. | Input absensi tanpa employeeId sebagai admin | Login admin → dapat token → input absensi tanpa employeeId | Sistem menampilkan error 400 dan pesan “employeeId wajib diisi” | Sistem menampilkan error validasi | ✅ Pass |
| 4. | Jalankan payroll dengan tanggal kosong | Login admin → POST payroll tanpa periodStart & periodEnd | Sistem menolak data (400 Bad Request) | Sistem mengembalikan error validasi Zod dengan status 500 | ✅ Pass (Validasi berjalan) |

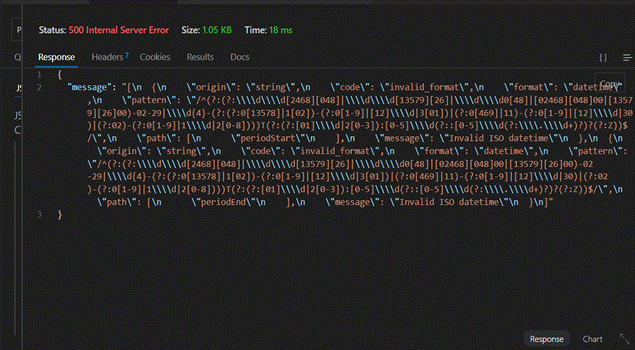
**Lampiran Screenshot & Response Penting**

**1.** **Screenshot response 401 saat akses admin tanpa token**

****

**2.** **Screenshot response 403 saat akses admin dengan token karyawan**

**3.** **Screenshoterror validasi absensi tanpa employeeId**

**4.** **Screenshot error Invalid ISO datetime saat payroll tanpa tanggal**

**Catatan Bug & Status Perbaikannya**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Bug Ditemukan** | **Penyebab** | **Dampak** | **Status** |
| **1.** | Payroll kosong mengembalikan status 500 | Error validasi Zod belum ditangani dengan status 400 | Status HTTP kurang sesuai standar | ⏳ Belum diperbaiki |
| **2.** | Validasi periode payroll wajib format ISO | Format input salah saat uji coba | Payroll gagal diproses | ✅ Diketahui |

Berdasarkan hasil pengujian terhadap prosedur autentikasi, otorisasi, absensi, dan payroll:

* ✅ Sistem berhasil melakukan validasi akses berdasarkan role pengguna
* ✅ Sistem berhasil mendeteksi kesalahan input data absensi
* ✅ Sistem berhasil memvalidasi format tanggal payroll
* ⚠️ Ditemukan bahwa status HTTP untuk error validasi masih perlu disesuaikan dari 500 menjadi 400

**BAB IV**

**DOKUMENTASI PENBJALANAN APLIKASI**

**4.1 Pembuatan dan Upload README.md**

**4.1.1 Isi file README.md**

**4.1.2 Proses commit ke repository**

**4.1.3 Screenshot commit history**

**4.2 Proses Clone Repository**

**4.2.1 Langkah awal: git clone [URL]**

**4.2.2 Langkah akhir: verifikasi file**

**4.2.3 Screenshot proses cloning**

**4.3 Panduan Menjalankan Aplikasi**

**4.3.1 Prasyarat sistem**

**4.3.2 Langkah instalasi**

**4.3.3 Cara menjalankan aplikasi**

**4.4 Demonstrasi Sistem (Screenshots)**

**BAB V**

**PENUTUP**

**5.1 Kesimpulan**

Berdasarkan hasil implementasi yang telah dilakukan, dapat disimpulkan bahwa proses penerapan CI/CD pipeline pada proyek TBP-Dashboard Monitoring System berhasil berjalan dengan baik. Pipeline yang dibangun mampu menjalankan proses build, testing dasar, hingga deployment otomatis ke GitHub Pages tanpa hambatan berarti. Selain itu, dashboard monitoring yang dikembangkan juga dapat di-*deploy* secara otomatis setiap kali terdapat perubahan pada branch utama, sehingga mempermudah proses pemeliharaan dan pembaruan website. Implementasi ini secara keseluruhan mampu meningkatkan efisiensi proses pengembangan, karena developer tidak lagi perlu melakukan langkah build dan upload secara manual, sehingga risiko kesalahan manusia dapat diminimalkan.

**5.2 Kendala Pengembangan**

Untuk pengembangan lebih lanjut, terdapat beberapa aspek yang dapat ditingkatkan agar sistem menjadi lebih stabil dan berkualitas. Pertama, perlu ditambahkan automated testing yang lebih komprehensif, seperti pengujian unit, integrasi, serta validasi struktur file yang lebih mendalam, sehingga potensi error dapat terdeteksi lebih awal sebelum sistem di-*deploy*. Selanjutnya, penerapan *quality gates* seperti pengecekan linting dan pengukuran code coverage dapat meningkatkan standar kualitas kode sekaligus menjaga konsistensi antar developer. Selain itu, integrasi dashboard dengan tools monitoring eksternal seperti Grafana, Prometheus, atau API data real-time lainnya dapat memberikan data yang lebih kaya serta meningkatkan kapabilitas visualisasi dan analisis. Dengan berbagai pengembangan ini, sistem diharapkan dapat menjadi lebih robust, informatif, dan siap untuk digunakan pada skala yang lebih besar.

**LAMPIRAN**