

INTEGRASI SISTEM ABSENSI MAHASISWA MAGANG BERBASIS TELEGRAM BOT DENGAN WEB MONITORING

Amelia Cristin Ginting, Evaluata Br Sembiring, S.Kom., M.Cs

Program Studi Teknik Informatika, Politeknik Negeri Batam

Email: ameliacristin718@gmail.com, eva@polibatam.ac.id

Article Info

Article history:

Keyword:

Digital Attendance, Telegram Bot, Node Red, Web Monitoring; Waterfall

ABSTRACT

Manual attendance for internship students is still commonly used, such as paper signatures or spreadsheet input. These methods are prone to data manipulation and slow recapitulation. This study developed a digital attendance system using a Telegram Bot integrated with a web dashboard through the prototyping method.

Students submit attendance via Telegram, processed through Node-RED, and stored in a PostgreSQL database. Data is displayed in real-time via a Next.js dashboard. The system, built with React.js, Node.js, and PostgreSQL, was tested and showed 70% faster attendance processing, zero input errors, and instant report generation. This system effectively supports real-time internship attendance monitoring for companies and educational institutions.

Absensi manual mahasiswa magang masih umum digunakan, seperti tanda tangan di kertas atau input spreadsheet. Metode ini rawan manipulasi data dan lambat dalam rekapitulasi. Penelitian ini mengembangkan sistem absensi digital berbasis Telegram Bot yang terintegrasi dengan dashboard web menggunakan metode prototyping. Mahasiswa melakukan absensi melalui Telegram, diproses oleh Node-RED, dan disimpan di PostgreSQL. Data ditampilkan secara real-time di dashboard Next.js. Sistem ini, yang dibangun dengan React.js, Node.js, dan PostgreSQL, telah diuji dan menunjukkan proses absensi 70% lebih cepat, tanpa kesalahan input, serta laporan yang langsung tersedia. Sistem ini efektif dalam mendukung pemantauan absensi magang secara real-time bagi perusahaan dan institusi pendidikan.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

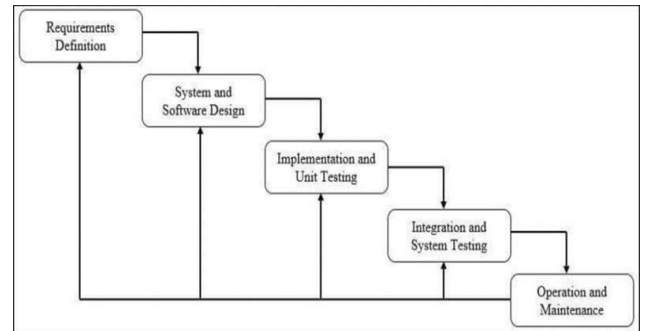
Di era digital saat ini, transformasi teknologi informasi telah mendorong banyak perusahaan, termasuk perusahaan manufaktur, untuk beralih dari sistem administrasi manual ke sistem digital. Salah satu proses penting yang memerlukan digitalisasi adalah pencatatan kehadiran atau absensi. Sayangnya, dalam praktiknya, proses absensi mahasiswa magang di perusahaan masih sering dilakukan secara manual, seperti tanda tangan pada lembar kertas atau pengisian spreadsheet. Metode ini memiliki beberapa kekurangan, antara lain potensi kesalahan pencatatan, manipulasi data, serta kurang efisien dalam proses monitoring dan rekapitulasi. [1]

Data dari observasi langsung serta hasil wawancara dengan bagian HRD perusahaan menunjukkan bahwa proses manual tersebut menyulitkan dalam hal pengawasan kehadiran mahasiswa secara real-time. Selain itu, pencatatan manual juga rawan terhadap kehilangan data, tidak terintegrasi, dan sulit diakses oleh pihak kampus. Oleh karena itu, dibutuhkan sebuah sistem yang mampu menjawab kebutuhan efisiensi, transparansi, dan keakuratan dalam pencatatan kehadiran mahasiswa magang. [2] Salah satu solusi potensial yang dapat diterapkan adalah penggunaan *Telegram Bot* sebagai media input kehadiran, serta *Node-RED* sebagai sistem pengolah logika data. Telegram dipilih karena ringan, cepat, bersifat multiplatform, serta sudah umum digunakan oleh mahasiswa

Sementara itu, Node-RED memungkinkan integrasi dan pemrosesan data melalui antarmuka visual tanpa penulisan kode kompleks, sehingga memudahkan pengembangan sistem secara cepat dan fleksibel. [3]. Data kehadiran yang dikirim melalui Telegram Bot akan diproses oleh *Node-RED* dan disimpan ke dalam database, kemudian ditampilkan melalui platform web monitoring. Sistem ini memungkinkan pihak perusahaan maupun dosen pembimbing dari kampus untuk memantau kehadiran mahasiswa secara real-time, fleksibel, dan terbuka. Pendekatan ini juga telah digunakan dalam beberapa penelitian terdahulu, seperti pengembangan sistem absensi berbasis *Telegram Bot* dan *RFID*, yang terbukti mampu meningkatkan akurasi dan efisiensi pencatatan kehadiran karena bersifat otomatis dan langsung terhubung dengan backend. [4] Dengan mempertimbangkan kebutuhan sistem yang efisien dan berbasis digital, maka penelitian ini bertujuan untuk mengembangkan sistem absensi mahasiswa magang berbasis *Telegram Bot* yang terintegrasi dengan web monitoring. Sistem ini diharapkan mampu menjawab permasalahan dalam pencatatan kehadiran mahasiswa secara modern dan terkontrol. Penggunaan sistem absensi otomatis berbasis *Telegram Bot* dan *Node-RED* telah banyak dibahas dalam penelitian dan implementasi terdahulu. Beberapa penelitian menunjukkan bahwa penggunaan *Telegram Bot* dalam sistem absensi mampu meningkatkan efisiensi dan meminimalisir kecurangan dalam pencatatan kehadiran. Penelitian lainnya menggabungkan penggunaan *Telegram Bot* dengan teknologi *RFID* untuk absensi otomatis yang terintegrasi langsung dengan sistem backend, menunjukkan bahwa pendekatan ini mempercepat proses rekap dan pelaporan kehadiran. Dengan demikian, pendekatan yang diusulkan dalam penelitian ini memiliki dasar teoritis dan empiris yang kuat. [5]. Dalam pengembangannya, penelitian ini menggunakan *metode waterfall*, yaitu metode rekayasa perangkat lunak yang dilakukan secara bertahap dan terstruktur, mulai dari analisis kebutuhan hingga pemeliharaan. Setiap tahap dilakukan secara linear agar meminimalisir kesalahan dan memastikan bahwa hasil akhir dapat tercapai sesuai rencana pengembangan. Model ini dipilih karena cocok untuk proyek dengan ruang lingkup dan kebutuhan yang jelas sejak awal. [6]

II. METODE

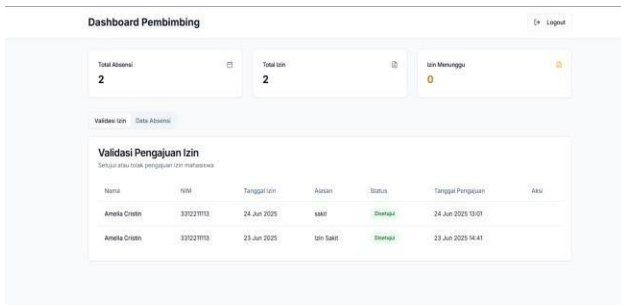
Penelitian ini menggunakan metode pengembangan perangkat lunak *Waterfall*, yaitu metode yang bersifat linier dan sistematis. Pada metode ini, setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya (Pressman, 2010). Metode *Waterfall* cocok digunakan dalam proyek yang memiliki kebutuhan yang jelas dan stabil sejak awal pengembangan, tahapan metode *waterfall* dapat dilihat pada gambar 1.



Gambar 1 Metode *Waterfall*

Metode *Waterfall* bertujuan sebagai pendekatan pengembangan sistem yang terstruktur dan sistematis, di mana setiap tahapan dilakukan secara berurutan dan tidak dapat dilanjutkan ke tahap berikutnya sebelum tahap sebelumnya selesai. [7]

1. Tahapan Analisis Kebutuhan, Tahap ini adalah tahap awal dalam pembangunan sebuah sistem. Kebutuhan aplikasi yang dikembangkan meliputi perangkat lunak yang digunakan, bahasa pemrograman yang digunakan, dan data yang diperlukan. Penulis melakukan analisis kebutuhan dan batasan sistem agar sistem yang nantinya dibuat sesuai dengan yang diharapkan. Informasi tersebut didapat melalui metode pengumpulan data yaitu observasi, wawancara, dan studi literatur. Hasil analisis kebutuhan adalah gambaran awal mengenai spesifikasi sistem yang mencakup fitur-fitur yang dibutuhkan, alur kerja sistem, serta sumber daya yang diperlukan agar sistem dapat berjalan secara optimal.
2. Membangun Desain, membangun Desain yang bertujuan memberikan gambaran lengkap tentang bagian apa saja yang harus dikerjakan dan bagaimana tampilan dari sebuah sistem, sehingga membantu peneliti dalam mendefinisikan arsitektur sistem yang akan dibuat secara keseluruhan. Tahapan Desain meliputi *Use Case Diagram* untuk menggambarkan interaksi pengguna dengan sistem, *Activity Diagram* untuk menunjukkan alur proses absensi, *ERD* untuk mendesain struktur database dan *Desain antarmuka Telegram Bot dan Web Monitoring* untuk menampilkan fitur yang akan digunakan oleh pengguna



- Implementasi Dan Pengujian Unit Tahap ini merupakan tahap pengembangan dari sistem berdasarkan hasil perancangan sebelumnya. Setiap modul atau komponen dari sistem mulai dikembangkan menggunakan bahasa pemrograman dan tools yang telah ditentukan Dalam penelitian ini, pengembangan dilakukan dengan menggunakan bahasa pemrograman *TypeScript/JavaScript* pada platform *Next.js* untuk web monitoring, dan *Node.js* untuk bot Telegram, serta *PostgreSQL* sebagai basis data.
- Integrasi Dan Pengujian Sistem Setelah seluruh modul diuji secara terpisah, tahap selanjutnya adalah melakukan integrasi antar modul dan melakukan pengujian sistem secara menyeluruh. Pada tahap ini, seluruh bagian sistem digabungkan dan diuji untuk melihat apakah sistem bekerja sebagaimana mestinya saat digunakan secara utuh.
- Operasional Dan Pemeliharaan, Tahap akhir dari metode Waterfall adalah implementasi sistem ke lingkungan operasional dan proses pemeliharaan sistem setelah digunakan. Tahap ini bersifat berkelanjutan agar sistem tetap relevan dan dapat digunakan dalam jangka panjang

III. HASIL DAN PEMBAHASAN

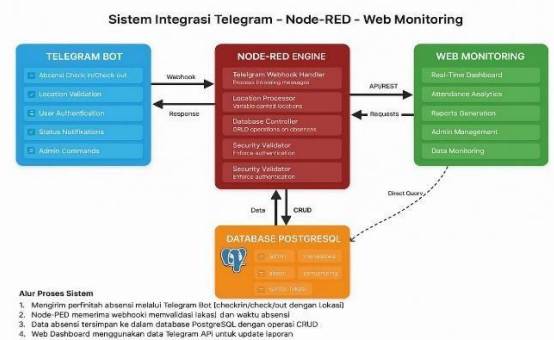
A. Tahapan Requirement

Sistem Absensi Mahasiswa Magang ini dibangun menggunakan arsitektur sistem terintegrasi yang dapat diakses melalui dua antarmuka utama, yaitu Telegram Bot untuk mahasiswa dan *Web Monitoring* untuk admin dan pembimbing. Tahapan requirement dilakukan dengan melakukan observasi, wawancara, serta studi literatur untuk memahami kebutuhan pengguna dan proses absensi yang ada saat ini. Berdasarkan hasil analisis, sistem yang akan dibangun memuat beberapa kebutuhan utama, di antaranya:

- Mahasiswa dapat melakukan absensi melalui perintah di Telegram Bot.
- Sistem memvalidasi identitas pengguna dan waktu

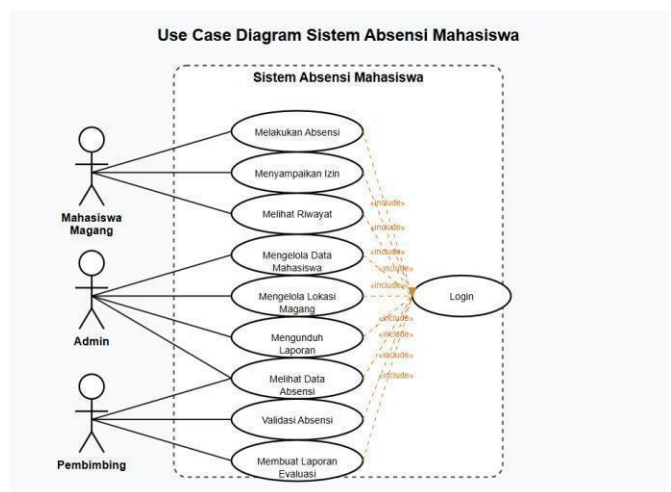
- absensi secara otomatis melalui Node-RED.
- Data absensi disimpan secara otomatis ke dalam database.
- Admin dan pembimbing dapat melihat data absensi mahasiswa melalui web dashboard.
- Sistem menyediakan fitur unduh laporan absensi dalam format Excel atau PDF.

Pada sistem ini memiliki 2 roles yaitu ada Admin untuk memantau dan merekap data kehadiran mahasiswa magang dan Mahasiswa Magang untuk melakukan absensi harian menggunakan perintah di Telegrambot serta dapat melihat riwayat kehadiran mereka sendiri jika di aktifkan



Gambar 3. 2 Gambaran Umum Sistem

2. Use Case Diagram



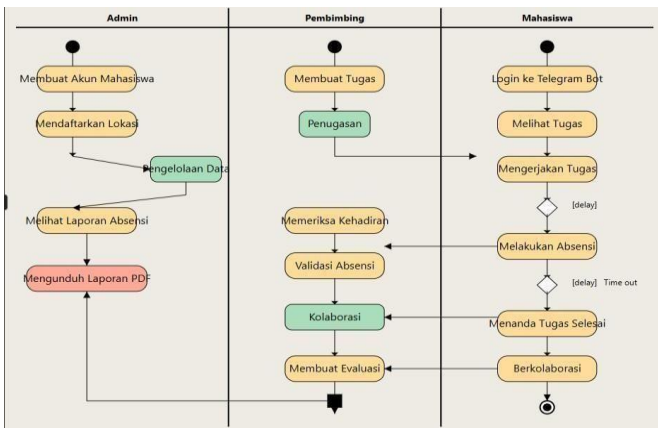
Gambar 3. 3 Use Case Diagram

Diagram use case di atas menunjukkan interaksi antara tiga aktor utama, yaitu Mahasiswa Magang, Admin dan Pembimbing. sebagai simbol dari peran mereka dan mengilustrasikan berbagai fungsi yang dapat diakses

Berikut adalah penjelasan mengenai alur dalam diagram tersebut:

- a. Mahasiswa Magang merupakan aktor utama yang menggunakan sistem untuk mencatat kehadirannya selama menjalani program magang. Adapun tugas mahasiswa dalam system yaitu Melakukan absens, Menyampaikan ijin dan Melihat Riwayat. Peran mahasiswa dalam sistem lebih kepada pengguna aktif yang melakukan input data kehadiran, baik hadir maupun izin.
- b. Admin berperan sebagai pengelola sistem dan data mahasiswa secara menyeluruh. Tugas admin yaitu Mengelola Data Mahasiswa, Mengelola Lokasi Magang
- c. Mengunduh Laporan, Melihat Data Absensi Secara umum, Admin bertindak sebagai pengelola dan pengawas administratif sistem absensi.
- d. Pembimbing memiliki peran sebagai pengawas atau evaluator kehadiran mahasiswa magang Tugas pembimbing yaitu Melihat Data Absensi, Validasi Absensi, Membuat Laporan Evaluasi.

3. Activity Diagram



Gambar 3. 4 Alur Activity Diagram

Dalam sistem ini, proses absensi dilakukan oleh mahasiswa magang melalui Telegram Bot. Mahasiswa mengirimkan perintah absensi menggunakan pesan tertentu, seperti /absen, ke dalam aplikasi Telegram. Pesan tersebut kemudian diproses oleh Bot dan diteruskan ke sistem Node-RED. Node-RED akan melakukan validasi terhadap data yang diterima, seperti mengecek ID pengguna, waktu absensi, dan format pesan. Jika data valid, maka Node-RED akan menyimpan data kehadiran ke dalam database PostgreSQL. Setelah proses penyimpanan selesai, sistem akan mengirimkan notifikasi balasan kepada mahasiswa bahwa absensi berhasil dilakukan.

Kemudian, data yang sudah tersimpan dalam database dapat ditampilkan secara real-time melalui dashboard web monitoring yang dibangun menggunakan Next.js. Admin atau pembimbing dapat mengakses dashboard tersebut untuk memantau kehadiran mahasiswa, melihat rekap data, serta mengunduh laporan absensi Bot. Perintah ini dapat berupa /absen atau perintah lain yang telah disediakan dalam sistem. Bot Telegram kemudian memproses data tersebut.

1. Mahasiswa Melakukan Absensi

Mahasiswa magang mengirimkan perintah absensi melalui Telegram Bot. Perintah ini dapat berupa *Absen* atau perintah lain yang telah disediakan dalam system. Bot Telegram kemudian memproses data tersebut.

2. Pemrosesan Absensi oleh Telegram Bot

Telegram Bot menerima perintah absensi dari mahasiswa, kemudian mengirimkan data tersebut ke *Node Red* untuk di proses. Node Red akan memvalidasi data, seperti memastikan waktu absensi dan *Id* mahasiswa sesuai

3. Validasi dan Penyimpanan Data Absensi

Node Red memeriksa apakah data absensi valid. Jika valid, *Node Red* akan menyimpan data absensi tersebut kedalam *PostgreSQL*. Jika data tidak valid, system akan mengirimkan pesan kesalahan kepada mahasiswa

4. Konfirmasi Absensi

Setelah absensi berhasil diproses dan disimpan didatabase, *Telegram Bot* mengirimkan pesan konfirmasi ke mahasiswa yang telah melakukan absensi, memberitahukan bahwa absensi telah tercatat dengan sukses.

5. Monitoring Absensi oleh Admin/Pembimbing

Admin/Pembimbing dapat mengakses dashboard web monitoring yang dibangun menggunakan *Next JS*. Pada dashboard ini, admin/ pembimbing dapat memonitor kehadiran mahasiswa secara real time, melihat laporan absensi, dan mengunduh laporan dalam format tertentu (PDF atau *Excel*)

6. Rekap Absensi

Admin dapat melakukan rekap absensi berdasarkan periode tertentu (harian, mingguan atau bulanan). Data absensi yang telah disimpan di *PostgreSQL* akan ditampilkan dalam bentuk table atau grafik yang mudah dipahami.

7. Penyelesaian Absensi

Jika seluruh proses sudah dilakukan dengan baik dan data tersimpan dengan benar, admin/pembimbing dapat menandai system sebagai selesai atau melakukan audit untuk memastikan tidak ada kesalahan dalam proses absensi.

1. Kebutuhan Fungsional

a. Functional Requirement (FR)

Kebutuhan fungsional merupakan pernyataan layanan sistem yang harus disediakan, bagaimana sistem harus bereaksi terhadap input tertentu, dan bagaimana sistem harus berperilaku dalam situasi tertentu. Berikut adalah hasil analisis mengenai kebutuhan fungsional pada sistem Absensi Mahasiswa Magang Berbasis Telegram Bot dengan Integrasi Web Monitoring.

Table 3. 2

| NO | KEBUTUHAN FUNGSIONAL |
|------|---|
| F001 | Pengguna (Admin,Mahasiswa Magang, Pembimbing Magang) melakukan login agar dapat mengakses sistem |
| F002 | Admin mengelola data mahasiswa magang seperti menambah, mengedit, atau menghapus data mahasiswa |
| F003 | Mahasiswa melakukan absensi melalui Telegram Bot dengan perintah tertentu |
| F004 | Telegram Bot memberikan konfirmasi sukses atau gagal setelah menerima perintah absensi |
| F005 | Sistem menyimpan data absensi mahasiswa ke dalam database PostgreSQL secara otomatis |
| F006 | Admin mengakses dan melihat dashboard monitoring absensi mahasiswa secara real-time. |
| F007 | Admin mengunduh laporan absensi dalam format PDF atau Excel dari web monitoring |
| F008 | Admin memantau kehadiran mahasiswa magang berdasarkan periode tertentu (harian/mingguan) |
| F009 | Sistem menampilkan data absensi mahasiswa dalam bentuk tabel atau grafik di dashboard web |
| F010 | Admin mengelola akses pengguna untuk Telegram Bot dan Web berdasarkan peran (Admin, Mahasiswa) |
| F011 | Mahasiswa melihat riwayat absensi mereka melalui Telegram Bot atau dashboard web |

b. Non Functional Requirement (NFR)

Kebutuhan non fungsional tidak secara langsung terkait pada fitur tertentu. Kebutuhan non fungsional memberikan batasan pada kebutuhan fungsional. Berikut adalah analisis kebutuhan non fungsional pada sistem.

Table 3. 2

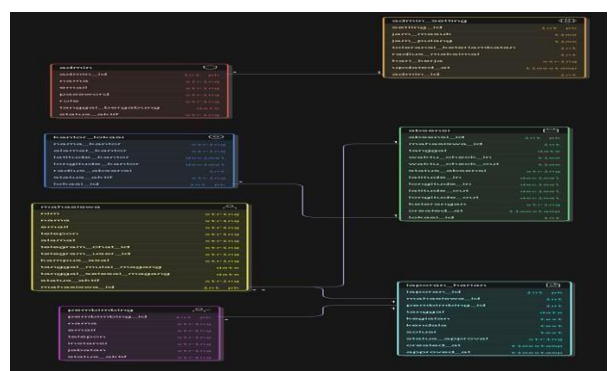
| KRITERIA | PARAMETER |
|--------------|--|
| Availability | Sistem dapat diakses apabila terhubung ke jaringan internet Sistem ini dapat beroperasi selama 24 jam tanpa henti, kecuali jika sedang dilakukan perawatan atau pembaruan system. |
| Bahasa | Sistem ini menggunakan Bahasa Inggris & Indonesia |
| Security | Sistem memastikan bahwa yang digunakan harus terlindungi dari akses yang data tidak berwenang |
| Ergonomy | Sistem dirancang dengan mudah digunakan oleh pengguna |

B. Membangun Desain

Tahap ini merupakan lanjutan dari proses analisis kebutuhan, di mana hasil dokumentasi kebutuhan pengguna diterjemahkan ke dalam bentuk rancangan sistem yang akan dibangun. Perancangan ini mencakup struktur sistem, alur proses, desain database, serta antarmuka pengguna yang akan digunakan dalam implementasi system.[8].

1. Entity Relationship Diagram

ERD (Entity Relationship Diagram) adalah sebuah diagram yang digunakan untuk memodelkan data dalam sebuah sistem informasi. ERD dapat membantu dalam memahami bagaimana entitas seperti objek atau tabel di dalam sistem berhubungan satu sama lain. Berikut ini menggambarkan tabel serta atribut database yang akan digunakan pada penelitian ini.



3. 5 ERD

Diagram di atas menggambarkan struktur basis data untuk system Absensi Mahasiswa Magang berbasis Telegram Bot dengan Web Monitoring. Terdapat tujuh tabel utama: *Mahasiswa*, *Absensi*, *Laporan Harian*, *Admin*, *Pembimbing*, *Kantor Lokasi* dan *Admin Setting*.

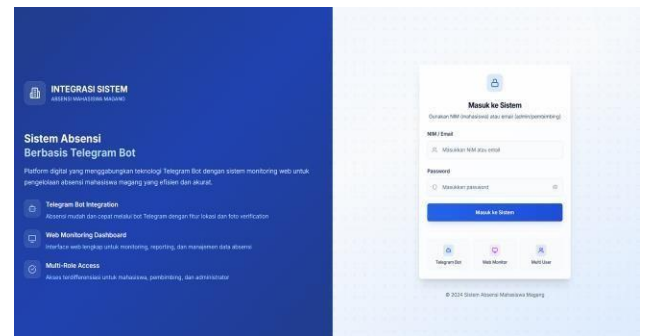
- a. Admin: Entitas admin menyimpan data pengguna yang memiliki hak akses sebagai administrator sistem. Kolom-kolom di dalamnya mencakup:
 1. Admin id: Id unik untuk admin (Primary key)
 2. Role :.Menyimpan peran admin
 3. Status aktif: Menunjukkan apakah admin tersebut aktif..
- b. Admin Setting: Menyimpan konfigurasi waktu dan aturan absensi yg ditetapkan oleh admin. Kolom-kolom di dalamnya mencakup:
 1. Jam masuk, jam pulang: batas waktu kerja.
 2. Keterlambatan: Batas keterlambatan.
 3. Radius maksimal: Radius berbasis lokasi.
 4. Hari kerja: Hari aktif bekerja
 5. Admin Id: Relasi ke table admin yang membuat pengaturan tersebut
- c. Mahasiswa: Menyimpan semua informasi mahasiswa yang mengikuti program magang. Kolom-kolom di dalamnya mencakup:
 1. Nim, nama, email, kampus: identitas dasar mahasiswa.
 2. Telegram_chat_id, telegram user_id: mengidentifikasi akun telegram mahasiswa
 3. Tanggal_mulai_magang, tanggal_selesai_magang: periode magang.
 4. Status_aktif: menandai mahasiswa yang sudah tidak aktif.
- d. Absensi: mencatat semua aktivitas absensi mahasiswa. Kolom-kolom didalamnya mencakup:
 1. Waktu_check_in, waktu_check_out: waktu datang dan pulang
 2. Latitude_in, longitude_out: lokasi saat absensi masuk
 3. Longitude_out, latitude_in: Lokasi absensi saat pulang
 4. Status_absensi: Seperti "hadir"
 5. Lokasi_id: relasi ke Lokasi kantor tempat absensi dilakukan
- e. Kantor Lokasi: Menyimpan informasi Lokasi Perusahaan Lokasi digunakan untuk memvalidasi presensi mahasiswa melalui perbandingan koordinat saat absen.
 1. Nama_kantor, Alamat_kantor: informasi Lokasi
 2. Latitude_kantor, longitude_kantor: titik koordinat lokasi
 3. Radius_absensi: radius maksimal untuk Lokasi

4. Status_aktif: apakah lokasi tersebut masi digunakan
- f. Pembimbing: pihak Perusahaan yang menjadi pembimbing mahasiswa magang. Kolom-kolom didalamnya yaitu:
 1. Nama, email, telfon, instansi, jabatan : Data pribadi
 2. Status_aktif: Menunjukkan apakah masih jadi pembimbing aktif
- g. Laporan harian: Mencatat laporan kegiatan mahasiswa magang setiap harinya. Kolom-kolom didalamnya yaitu:
 1. Tanggal: Tanggal laporan dibuat
 2. Kegiatan, kendala, Solusi: Isi laporan
 3. Status_approval: Status laporan
 4. Approved_at: Waktu persetujuan laporan
 5. Mahasiswa_id, pembimbing_id: Relasi ke mahasiswa yang membuat dan pembimbing yang memverifikasi

C. Implementasi Kode

1. Implementasi Antar Muka

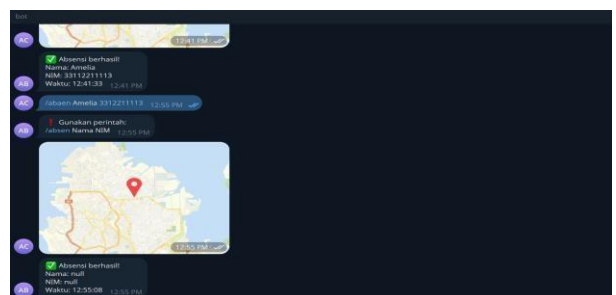
Halaman landing page merupakan halaman awal yang dapat diakses secara umum tanpa harus melakukan login. Halaman ini berfungsi sebagai tampilan depan dari sistem Absensi Mahasiswa Magang Implementasi halaman landing page ini dapat dilihat pada gambar dibawah ini.



Gambar 3.6 Landing Page

2. Implementasi Halaman Validasi Absensi Mahasiswa Halaman ini hanya dapat diakses oleh pembimbing atau admin yang telah berhasil login ke sistem. Pada halaman ini, pembimbing akan melakukan validasi terhadap data absensi yang dikirimkan oleh mahasiswa melalui Telegram Bot. Validasi dilakukan untuk memastikan bahwa absensi benar-benar dilakukan dalam waktu dan lokasi yang sesuai dengan aturan yang telah ditentukan.

Gambar 3.7 Form Input Data Absensi



3. Implementasi Halaman *History* Hasil validasi
 Pada halaman Riwayat Absensi, Mahasiswa dapat melihat hasil absensi yang telah mereka lakukan melalui Telegram Bot. Setiap data absensi yang ditampilkan pada halaman ini telah melewati proses validasi, baik secara otomatis oleh sistem maupun oleh pembimbing/admin jika diperlukan.

Gambar 3. 8 Halaman *Dashboard Admin*



D. Testing

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur dan alur sistem berfungsi dengan baik sesuai dengan kebutuhan pengguna. Sistem ini dibangun menggunakan metode *Waterfall*, sehingga pengujian dilakukan setelah seluruh tahapan pengembangan selesai, tepatnya setelah tahap implementasi. Sistem diuji secara bertahap, mulai dari pengujian tiap komponen, pengujian integrasi antar komponen, hingga pengujian sistem secara keseluruhan.[9]

Berikut hasil dari testing pada sistem ini :

| No | Fungsional | Skenario Pengujian | Role | Indikator Keberhasilan | Hasil Pengujian |
|----|------------|--|-----------|---|-----------------|
| 1 | Login | Mahasiswa memasukkan email & password dengan benar | Mahasiswa | Sistem menampilkan pesan "Berhasil login" dan pengguna diarahkan ke dashboard | Sesuai |
| | | Mahasiswa memasukkan email benar & password salah | Mahasiswa | Sistem menampilkan pesan "Password salah" | Sesuai |

| | | | | | |
|---|----------------------|--|-----------|--|--------|
| 2 | Absensi via Telegram | Mahasiswa mengirim perintah /absen Nama NIM | Mahasiswa | Sistem meminta lokasi, lalu menampilkan pesan "Kirim lokasi Anda untuk verifikasi" | Sesuai |
| | | Mahasiswa mengirim lokasi valid (dalam radius dan waktu kerja) | Mahasiswa | Sistem menampilkan pesan "Absensi berhasil" disertai nama dan waktu | Sesuai |
| | | Mahasiswa mengirim lokasi di luar radius kantor | Mahasiswa | Sistem menampilkan pesan "Lokasi tidak valid untuk absensi" | Sesuai |
| | | Mahasiswa mengirim perintah /absen tanpa format yang benar | Mahasiswa | Sistem menampilkan pesan "Gunakan perintah: /absen Nama NIM" | Sesuai |

| | | | | | |
|---|-----------------|---|-------|---|--------|
| 3 | Dashboard Admin | Admin login dan melihat data mahasiswa di dashboard | Admin | Sistem menampilkan tabel daftar mahasiswa aktif dan total absensi | Sesuai |
| | | Admin membuka tab "Data Absensi" | Admin | Sistem menampilkan riwayat absensi seluruh mahasiswa | Sesuai |

| | | | | | |
|---|--------------------------------|---|-----------|--|--------|
| 4 | Dashbo ard Mahasi swa | Mahasiswa login dan melihat halaman dashboard pribadi | Mahasiswa | Sistem menampilkan total absensi, pengajuan izin, dan riwayat absensi pribadi | Sesuai |
| | | Mahasiswa membuka tab "Pengajuan Izin" | Mahasiswa | Sistem menampilkan form pengajuan izin dan daftar status izin yang pernah diajukan | Sesuai |

E. Evaluasi Sistem

Evaluasi sistem dilakukan untuk memastikan bahwa seluruh fitur pada sistem Absensi Mahasiswa Magang berbasis Telegram Bot dengan Web Monitoring berjalan dengan baik dan sesuai kebutuhan pengguna. Evaluasi mencakup keakuratan absensi berdasarkan lokasi dan waktu, kemudahan penggunaan bot oleh mahasiswa, serta kejelasan tampilan data pada dashboard admin dan mahasiswa. Sistem akan terus diperbarui secara bertahap berdasarkan hasil pengujian dan masukan dari pengguna. Setelah sistem dinyatakan stabil, maka aplikasi ini siap digunakan sepenuhnya oleh mahasiswa, admin, dan pembimbing dalam proses absensi dan monitoring kehadiran.

IV. KESIMPULAN

Sistem absensi Mahasiswa Magang berbasis Telegram Bot dengan Web Monitoring telah berhasil dibangun dan berjalan sesuai dengan tujuan. Sistem ini mempermudah mahasiswa dalam melakukan absensi secara otomatis melalui Telegram, serta memudahkan admin dan pembimbing dalam memantau kehadiran melalui dashboard web. Fitur validasi waktu dan lokasi membuat absensi lebih akurat dan terpercaya. Berdasarkan hasil pengujian, seluruh fungsi utama bekerja dengan baik. Meskipun begitu, sistem ini masih dapat dikembangkan lebih lanjut untuk menambah fitur-fitur pendukung agar lebih optimal.

DAFTAR PUSTAKA

A. Referensi Akademik dan Karya Ilmiah Lokal

Arifin, H. N., & Sari, N. L. D. M. (2024). Rancang bangun sistem informasi manajemen proyek berbasis web pada PT. ABISEKA. IN-FEST 2024 - Seminar Nasional Informatika, FTI UPGRIS, 2, 58–65.

Fatman, Y., Hidayat, I. S., & Anadhiya, N. I. (2024). Rancang bangun sistem informasi tugas akhir menggunakan metode Waterfall di Fakultas Teknik Universitas Islam Nusantara. *Edusaintek: Jurnal Pendidikan, Sains dan Teknologi*, 11(1), 290–299. <https://doi.org/10.47668/edusaintek.v11i1.988>

Hariono, & Widya. (2022). Implementasi Telegram Bot API untuk Informasi Kehadiran Siswa di Sekolah [Skripsi, Universitas Tidak Disebutkan].

Kasanova, Nurraharjo, Budiarmo, & Utomo. (2020). Presensi siswa berbasis RFID terintegrasi web dengan notifikasi bot Telegram [Laporan Penelitian].

Kurnia, M. R. (2022). Aplikasi purchase order outsole pada industri outsole sepatu dan sandal (Studi kasus: UD. Solindo) [Tugas akhir, Universitas Widyatama].

Lai, A., & Ramadhan, I. (2024). Rancang bangun aplikasi manajemen proyek berbasis web menggunakan framework Laravel dan Livewire. *Jurnal Ilmu Komputer*, 9(1), 9–18. <https://doi.org/10.47007/komp.v7i01.xxxxx>

Mame, S. M. A. (2024). Rancang bangun backend aplikasi Karir IO berbasis web menggunakan NestJS [Skripsi, STT Nurul Fikri].

Muhamad Annas Rivai. (2021). Implementasi metode Waterfall untuk sistem informasi e-voting pemilihan ketua OSIS [Tugas akhir, Universitas Semarang].

Nurjaman, A. S., & Yasin, V. (2020). Konsep desain aplikasi sistem manajemen kepegawaian berbasis web. *JISICOM*, 4(2), 143–165. <http://journal.stmikjayakarta.ac.id/index.php/jisicom>

Paramita, D. (2015). Rancang bangun sistem informasi kolaboratif berbasis web. *Jurnal Buana Informatika*, 6(3), 195–202.

Pradana, A. A. (2024). Studi independen: Rancang bangun website pemesanan tiket pesawat menggunakan React [Skripsi, Universitas Dinamika].

Pratomo, A. B. (2023). Pengembangan website manajemen tugas berbasis web. *TEKNIMEDIA*, 4(2), 139–143.

Puspita, A., Yuningsih, Y., Amalia, H., & Lestari, A. F. (2023). Penerapan metode waterfall dalam sistem pembelian alat kesehatan. *Infotek*, 6(2), 311–318.
<https://dx.doi.org/10.29408/jit.v6i2.12974>

Purwandari, N., & Fauzi, A. (2020). Perancangan sistem informasi manajemen pada Toko XYZ. *Jurnal Sistem Informasi*, 1(2), 54–64.

Raharja, M. P., & Wijayanto, H. (2023). Pengembangan prototype backend pelaporan barang hilang. *JBegaTI*, 4(2), 204–213.
<http://begawe.unram.ac.id/index.php/JBTI/>

Rohfikha, N., et al. (2022). Perancangan sistem informasi kependudukan berbasis web menggunakan metode waterfall. *Science and Engineering National Seminar 7 (SENS 7)*, Semarang.

Saputra, M. N. R. (2024). Rancang bangun platform konsultasi mental online menggunakan ReactJS [Skripsi, STT Nurul Fikri].

Sumarsono. (2021). Absensi Sidik Jari Portable dengan Telegram [Skripsi, Universitas Tidak Disebutkan].

Syaputra, S. M., Anwar, S., & Malau, Y. (2023). Sistem informasi manajemen tugas harian berbasis website. *Jurnal INSAN*, 3(2), 62–71.
<https://jurnal.bsi.ac.id/index.php/jinsan>

Trisianto, C. (2018). Penggunaan metode waterfall untuk sistem monitoring pedesaan. *Jurnal Teknologi Informasi ESIT*, XII(01), 7–8.

B. Referensi Internasional & Standar Akademik

Stallings, W. (2017). *Network Security Essentials: Applications and Standards* (6th ed.). Pearson.

Fette, I., & Melnikov, A. (2011). The WebSocket Protocol. RFC 6455. Internet Engineering Task

Force (IETF).
<https://datatracker.ietf.org/doc/html/rfc645g>

Al-Fuqaha, A., et al. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376.
<https://doi.org/10.1109/COMST.2015.2444095>

Li, S., Xu, L. D., & Zhao, S. (2018). The Internet of Things: A Survey. *Information Systems Frontiers*, 17(2), 243–259.
<https://doi.org/10.1007/s10796-014-9492-7>

Kouicem, D. E., Bouabdallah, A., & Lakhlef, H. (2018). Internet of Things Security: A Top-Down Survey. *Computer Networks*, 141, 199–221.
<https://doi.org/10.1016/j.comnet.2018.03.012>

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

C. Dokumentasi Resmi Teknologi

Telegram. (2023). Telegram Bot API Documentation. <https://core.telegram.org/bots/api>

IBM. (2022). Node-RED Documentation. <https://nodered.org/docs>

Vercel. (2023). Next.js Documentation. <https://nextjs.org/docs>

PostgreSQL Global Development Group. (2024). PostgreSQL 16 Documentation. <https://www.postgresql.org/docs/>

Mozilla Developer Network (MDN). (2024). HTTPS Protocol Overview. <https://developer.mozilla.org/en-US/docs/Web/HTTPS>