

|  |
| --- |
| [Company name] |
| Laporan Akhir |
| Sistem Informasi Jaringan dan Aplikasi |

|  |
| --- |
| Favian Dwi Yuliansyah  [Date] |

**MEMBANGUN SISTEM ABSENSI BERBASIS *WEBSITE* MENGGUNAKAN NEXT JS CHAKRA UI DAN GOOGLE SPREADSHEET SEBAGAI *DATABASE***

Hak Cipta 2024 Pada Penulis.

|  |  |  |
| --- | --- | --- |
| Penulis | : | Favian Dwi Yuliansyah |
| Editor | : | Favian Dwi Yuliansyah |
| Desain Cover | : | Favian Dwi Yuliansyah |
| Korektor | : | Dendi Mochamad Jabar, S.Pd. |

|  |  |  |
| --- | --- | --- |
| Redaksi : | | |
| **SMK NEGERI 1 CIMAHI** | | |
| Jl. Mahar Martanegara No.48, Utama,  Kec. Cimahi Sel., Kota Cimahi, Jawa Barat 40533 | | |
| Telp. | : | (022) 6629683 |
| Email | : | [info@smkn1-cmi.sch.id](mailto:info@smkn1-cmi.sch.id) |

KATA PENGANTAR

Puji Syukur penulis penajtkan kepada Allah SWT. karena dengan rahmat dan karunia-Nya penulis dapat menyelesaikan praktik kerja industri yang dilaksanakan di PT Telkom Indonesia. Tak lupa sholawat serta salam selalu tercurah kepada Nabi Muhammad SAW., keluraga, para sahabat, dan umatnya hingga akhir zaman.

Adapun tujuan dari penulisan laporan ini adalah untuk memenuhi salah satu syarat kelulusan dalam menempuh pembelajaran di SMK Negeri 1 Cimahi sekaligus sebagai laporan pertanggungjawaban tertulis mengenai pelaksanaan Praktik Kerja Lapangan (PKL) di PT. Telkom Indonesia selama kurang lebih 6 (enam) bulan yang dimulai dari tanggal 3 Juli 2023 sampai dengan 31 Desember 2023.

Dalam karya tulis ini penulis mengambil judul “**MEMBANGUN SISTEM ABSENSI BERBASIS *WEBSITE* MENGGUNAKAN NEXT JS CHAKRA UI DAN GOOGLE SPREADSHEET SEBAGAI *DATABASE* DI PT TELKOM INDONESIA**”.

Aplikasi yang penulis bangun merupakan aplikasi yang dapat mempermudah manajemen dalam merekap absensi pegawai dan juga meningkatkan efisiensi kerja manajemen dalam merekap absensi pegawai.

Penulis menyadari dalam penyususan laporan ini banyak mendapat dukungan, bimbingan, bantuan dan kemudahan dari berbagai pihak sehingga laporan ini dapat terselesaikan. Dengan ketulusan hati, penulis menyampaikan ucapan terima kasih kepada :

1. Kedua orang tua dan seluruh keluarga yang senantiasa mendoakan dan memberi dukungan moral, spiritual, serta materil.
2. Bapak Agus Priyatmono Nugroho, S.Pd., M.Si., selaku Kepala Sekolah Kejuruan Negeri 1 Cimahi.
3. Bapak Antoni Budiman, M.Pd., selaku Ketua Kompetensi Keahlian Sistem Informatika Jaringan dan Aplikasi.
4. Bapak Diky Ridwan, S.Kom., selaku Wali Kelas XIII Sistem Informatika Jaringan dan Aplikasi A.
5. Bapak Dendi Mochamad Jabar, S.Pd., selaku pembimbing dari pihak sekolah yang telah banyak memberikan ilmu, nasihat, dan semangat untuk kelancaran pelaksanaan PKL dan penyusunan laporan PKL ini.
6. Seluruh staf Hubungan Industri SMK Negeri 1 Cimahi.
7. Mas Pepen Supriatna, selaku *Head* *of* *Engineering* *Tribe* TCO dan pembimbing dari pihak industri yang telah membina dan memberikan ilmu dan masukan selama Praktik Kerja Lapangan.
8. Mas Gegy Faskal Nur Hamdani, selaku *Lead* *Frontend Tribe* TCO dari pihak industri yang telah membimbing dan memberikan ilmu dan masukan mengenai *frontend developer* selama Praktik Kerja Lapangan.
9. Seluruh staf karyawan dan rekan di Tribe TCO PT. Telkom Indonesia yang bersedia memberikan pengarahan selama Praktik Kerja Lapangan.
10. Teman-teman SIJA Angkatan 47 yang telah membantu dan memberikan dukungan serta saran kepada penulis dalam penyusunan laporan ini.
11. Semua pihak yang tidak dapat penulis sebutkan satu-persatu yang telah membantu dalam penyelesaian penulisan laporan ini.

Akhir kata penulis berharap laporan ini dapat bermanfaat dan menjadi sesuatu yang dapat menambah wawasan pengetahuan bagi penulis dan setiap orang yang membacanya.

Bandung, Februari 2024

Penulis

DAFTAR ISI

[KATA PENGANTAR i](#_Toc158566356)

[DAFTAR ISI iv](#_Toc158566357)

[DAFTAR GAMBAR vi](#_Toc158566358)

[DAFTAR TABEL ix](#_Toc158566359)

[BAB I PENDAHULUAN 10](#_Toc158566360)

[1.1. Latar Belakang Masalah 10](#_Toc158566361)

[1.2. Tujuan 11](#_Toc158566362)

[1.3. Pembatasan Masalah 11](#_Toc158566363)

[1.4. Sistematika Penulisan 12](#_Toc158566364)

[BAB II MEMBANGUN SISTEM ABSENSI BERBASIS WEBSITE MENGGUNAKAN NEXT JS DI PT TELKOM INDONESIA 14](#_Toc158566365)

[2.1. Analisa Sistem Absensi Pegawai 14](#_Toc158566366)

[2.2. Analisa Kebutuhan Alat & Perangkat 15](#_Toc158566367)

[2.3. Perancangan Sistem Absensi Karyawan 17](#_Toc158566368)

[2.3.1. Perancangan *Flowmap* 18](#_Toc158566369)

[2.3.2. Perancangan *Data* *Flow* *Diagram* 22](#_Toc158566370)

[2.4. Perancangan *Database* 24](#_Toc158566371)

[2.5. Perancangan Layout (Mock Up) 25](#_Toc158566372)

[2.6. Persiapan Pembuatan Aplikasi 29](#_Toc158566373)

[4.1. Pembuatan Basis Data 33](#_Toc158566374)

[2.7. Pembuatan Antarmuka *Website* 38](#_Toc158566375)

[2.5.1. Halaman Absensi 38](#_Toc158566376)

[2.5.2. Halaman *Login* 41](#_Toc158566377)

[2.5.3. Halaman Dashboard 41](#_Toc158566378)

[2.5.4. Halaman Daily Summary 43](#_Toc158566379)

[2.5.5. Halaman Monthly Summary 44](#_Toc158566380)

[BAB III PENGUJIAN 46](#_Toc158566381)

[3.1. Pengujian Website Absensi Karyawan 46](#_Toc158566382)

[BAB IV PENUTUP 64](#_Toc158566383)

[4.2. Kesimpulan 64](#_Toc158566384)

[4.3. Saran 64](#_Toc158566385)

DAFTAR GAMBAR

[Gambar 2. 1 *Flowmap* Absensi 18](#_Toc158566299)

[Gambar 2. 2 *Flowmap* *Login* 19](#_Toc158566300)

[Gambar 2. 3 *Flowmap* *Daily* *Summary* 20](#_Toc158566301)

[Gambar 2. 4 *Flowmap* *Monthly* *Summary* 21](#_Toc158566302)

[Gambar 2. 5 DFD *Level* 0 22](#_Toc158566303)

[Gambar 2. 6 DFD *Level* 1 23](#_Toc158566304)

[Gambar 2. 7 *Layout* Halaman Absensi 26](#_Toc158566305)

[Gambar 2. 8 *Layout* Halaman *Login* 26](#_Toc158566306)

[Gambar 2. 9 *Layout* Halaman *Dashboard* *Admin* 27](#_Toc158566307)

[Gambar 2. 10 *Layout* Halaman *Daily* *Summary* 28](#_Toc158566308)

[Gambar 2. 11 *Layout* Halaman *Monthly* *Summary* 28](#_Toc158566309)

[Gambar 2. 12 Instalasi Node.JS 29](#_Toc158566310)

[Gambar 2. 13 *Setup* *Project* 30](#_Toc158566311)

[Gambar 2. 14 Instalasi *library* *postgres* 30](#_Toc158566312)

[Gambar 2. 15 Instalasi library axios 31](#_Toc158566313)

[Gambar 2. 16 Instalasi *library* *prisma* dan *prisma* *client* 32](#_Toc158566314)

[Gambar 2. 17 Instalasi *library* *Chakra* *UI* dan *Sweetalert2* 32](#_Toc158566315)

[Gambar 2. 18 Instalasi *library* *React Chart JS 2* 33](#_Toc158566316)

[Gambar 2. 19 Mengunduh *PostgreSQL* 34](#_Toc158566317)

[Gambar 2. 20 Instalasi *PostgreSQL* 34](#_Toc158566318)

[Gambar 2. 21 Instalasi *PostgreSQL* 35](#_Toc158566319)

[Gambar 2. 22 Inisialisasi *prisma* 35](#_Toc158566320)

[Gambar 2. 23 Isi *file* *schema*.*prisma* 36](#_Toc158566321)

[Gambar 2. 24 Isi *file* .*env* 36](#_Toc158566322)

[Gambar 2. 25 Proses migrasi *database* menggunakan *prisma* 37](#_Toc158566323)

[Gambar 2. 26 *Database* pada *PostgreSQL* 37](#_Toc158566324)

[Gambar 2. 27 Halaman *form* absensi 40](#_Toc158566325)

[Gambar 2. 28 Halaman sukses ketika berhasil absen 40](#_Toc158566326)

[Gambar 2. 29 Halaman *login* 41](#_Toc158566327)

[Gambar 2. 30 Tampilan halaman *dashboard* 42](#_Toc158566328)

[Gambar 2. 31 Tampilan halaman daily summary 43](#_Toc158566329)

[Gambar 2. 32 tampilan halaman *monthly* *summary* 44](#_Toc158566330)

[Gambar 3. 33 Pengujian halaman absensi 1 48](#_Toc158566331)

[Gambar 3. 34 Pengujian halaman absensi 2 49](#_Toc158566332)

[Gambar 3. 35 Pengujian halaman absensi 3 49](#_Toc158566333)

[Gambar 3. 36 Pengujian halaman absensi 4 50](#_Toc158566334)

[Gambar 3. 37 Pengujian halaman absensi 5 50](#_Toc158566335)

[Gambar 3. 38 Pengujian halaman *login* 1 52](#_Toc158566336)

[Gambar 3. 39 Pengujian halaman *login* 2 53](#_Toc158566337)

[Gambar 3. 40 Pengujian halaman *login* 3 53](#_Toc158566338)

[Gambar 3. 41 Pengujian halaman *login* 4 54](#_Toc158566339)

[Gambar 3. 42 Pengujian halaman *dashboard* 1 56](#_Toc158566340)

[Gambar 3. 43 Pengujian halaman *dashboard* 2 56](#_Toc158566341)

[Gambar 3. 44 Pengujian halaman *dashboard* 3 57](#_Toc158566342)

[Gambar 3. 45 Pengujian halaman *daily* *summary* 1 59](#_Toc158566343)

[Gambar 3. 46 Pengujian halaman *daily* *summary* 2 59](#_Toc158566344)

[Gambar 3. 47 Pengujian halaman *daily* *summary* 3 60](#_Toc158566345)

[Gambar 3. 48 Pengujian halaman *monthly* *summary* 1 62](#_Toc158566346)

[Gambar 3. 49 Pengujian halaman *monthly* *summary* 2 62](#_Toc158566347)

[Gambar 3. 50 Pengujian halaman monthly summary 3 63](#_Toc158566348)

DAFTAR TABEL

[Tabel 2. 1 Analisa kebutuhan perangkat keras (*Hardware*) 15](#_Toc158566349)

[Tabel 2. 2 Perancangan basisi data tabel absensi 24](#_Toc158566350)

[Tabel 3. 4 Pengujian *black* *box* pada halaman absensi 46](#_Toc158566351)

[Tabel 3. 5 Pengujian *black* *box* pada *login* 51](#_Toc158566352)

[Tabel 3. 6 Pengujian *black* *box* sistem pada halaman *dashboard* 54](#_Toc158566353)

[Tabel 3. 7 Pengujian *black* *box* pada fitur *daily* *summary* 57](#_Toc158566354)

[Tabel 3. 8 Pengujian *black* *box* pada *monthly* *summary* 60](#_Toc158566355)

BAB I PENDAHULUAN

1. Latar Belakang Masalah

Dalam era perkembangan teknologi yang pesat, penggunaan teknologi sebagai alat bantu untuk mempermudah berbagai aktivitas menjadi suatu keharusan. Pencatatan absensi pegawai, yang pada awalnya dilakukan secara konvensional, kini membutuhkan perubahan untuk beralih ke sistem yang lebih modern dan efisien. PT Telkom Indonesia, khususnya Tribe Technology Commercialization (Tribe TCO), saat ini masih mengandalkan Google Form sebagai sistem absensi dan rekap absensinya yang masih membutuhkan input manual ke Spreadsheets. Hal ini memunculkan kebutuhan untuk memperbarui sistem tersebut agar pekerjaan menjadi lebih efisien.

Pencatatan data secara konvensional seringkali berpotensi menyebabkan kesalahan dan kehilangan data, yang pada akhirnya dapat merugikan perusahaan. Melihat potensi masalah ini, Tribe TCO di PT Telkom Indonesia merencanakan perubahan pada sistem absensinya. Tujuan utamanya adalah untuk menggantikan penggunaan Google Form dengan sistem aplikasi website yang dikembangkan secara internal. Selain itu, rekap absensi juga akan diotomatisasi, menghilangkan kebutuhan untuk melakukan input manual ke dalam Spreadsheets.

Dalam upaya untuk meningkatkan efisiensi dan kehandalan sistem, PT Telkom Indonesia memutuskan untuk mengubah database yang digunakan. Langkah ini diambil untuk memperkuat keamanan dan skalabilitas sistem, serta memungkinkan pengelolaan data yang lebih efisien dan terstruktur.

Penggunaan aplikasi web dan mobile diusulkan sebagai solusi yang lebih efisien. Keuntungan utamanya adalah penghematan waktu yang signifikan, di mana karyawan yang bertugas merekap tidak perlu lagi menghabiskan waktu untuk melakukan input data absensi secara manual. Dengan demikian, perubahan ini tidak hanya akan memberikan manfaat dalam hal keakuratan data, tetapi juga akan meningkatkan efisiensi pengelolaan data absensi karyawan.

1. Tujuan

Tujuan dibuatnya aplikasi web absensi kepegawaian adalah untuk:

1. Membangun sistem website absensi yang efisien dengan menggunakan *framework Javascript, Next.JS* juga menggunakan *Prisma ORM* dan *PostgreSQL* sebagai *database*.
2. Menggantikan sistem rekapitulasi dari manual menjadi layanan otomatis berbasis *website* yang terintegrasi dengan *database*.
3. Meningkatkan efisiensi kerja dalam merekap absensi karyawan untuk memperbaiki pengelolaan data dan mengurangi kesalahan dalam rekapitulasi.
4. Pembatasan Masalah

Adapun penulisan ini dibatasi dengan hal sebagai berikut:

1. Aplikasi berbasis *web* yang dibuat hanya mencakup *website* sistem absensi karyawan serta sebuah *dashboard* *admin* yang menyajikan *daily* *summary* dan *monthly* *summary* secara sederhana.
2. *Tools* yang digunakan dalam membangun website tersebut menggunakan *framework* *Next.JS* 14.1.0, Chakra UI 2.8.0, dan aplikasi yang digunakan untuk menyimpan *database* adalah *Prisma* *ORM* dan *PostgreSQL* dengan koneksi *Next.JS* menggunakan *Axios*.
3. Penggunaan *website* ini terbatas pada pihak manajemen *Tribe* *TCO* yang merekap data sebagai *admin* dan karyawan yang melakukan absensi sebagai *user*.
4. Aplikasi *website* tersebut dilengkapi dengan beberapa fitur umum yang sederhana. Pada akun *admin*, terdapat fitur *daily* *summary* untuk merekam absensi karyawan berdasarkan tanggal, serta fitur *monthly* *summary* untuk merekam absensi karyawan berdasarkan bulan. Setiap fitur tersebut memungkinkan pengguna untuk mengekspor data dalam format .*xls* melalui akun *admin*. Serta terdapat fitur authentikasi yaitu *login*.
5. Tampilan *user* dan *admin website* absensi karyawan sepenuhnya menggunakan *component* dari *Chakra* *UI*.
6. Projek kali ini tidak memiliki keterlibatan dalam proses *deployment* *website* ke ranah publik.
7. Sistematika Penulisan

Sistematika penulisan laporan ini terdiri dari beberapa bab disertai dengan kata pengantar, daftar isi, daftar gambar, daftar tabel, serta daftar Pustaka. Berikut pemaparan dari sistematika laporan ini.

**BAB I PENDAHULUAN**

Pada bab ini berisi tentang latar belakang masalah yaitu membahas hal-hal yang menjadi persoalan dalam judul laporan, tujuan yaitu membahas tentang tujuan dari judul yang penulis ambil, pembatasan masalah yaitu membatasi pembahasan yang diangkat oleh penulis dalam laporan, dan diakhiri dengan sistematika penulisan yang dapat memudahkan penulis dalam menyusun laporan.

**BAB II MEMBANGUN SISTEM ABSENSI BERBASIS *WEBSITE* MENGGUNAKAN NEXT JS CHAKRA UI DAN GOOGLE SPREADSHEET SEBAGAI *DATABASE* DI PT TELKOM INDONESIA**

Bab ini berisi tentang penjelasan mengenai skenario perancangan sistem dan proses *script*, meliputi analisa masalah, analisis kebutuhan perangkat, *flowmap, Data flow Diagram,* dan langkah pembuatan.

**BAB III PENGUJIAN**

Bab ini berisi pengujian aplikasi *website* sistem absensi menggunakan metode *black* *box* *testing*.

**BAB IV PENUTUP**

Bab ini berisi tentang kesimpulan dari kegiatan pada bab II dan bab III, serta terdapat saran-saran yang bersifat membangun terhadap proyek yang telah dibuat.

BAB II MEMBANGUN SISTEM ABSENSI BERBASIS WEBSITE MENGGUNAKAN NEXT JS DI PT TELKOM INDONESIA

1. Analisa Sistem Absensi Pegawai

Sistem absensi karyawan di *Tribe Technology Commercialization (TCO)* PT Telkom Indonesia masih mengandalkan *Google Form*. Namun, penggunaan *form* tersebut menjadi kurang efektif karena keterbatasan fitur seperti input teks, chekcbox, pilihan ganda, dan tanggal. Keterbatasan ini membuat kustomisasi *form* absensi menjadi kurang optimal.

Proses rekapitulasi absensi karyawan masih dilakukan secara manual dengan menginputkan data ke *Google* *Spreadsheet*. Pendekatan manual ini tidak hanya kurang efisien bagi *staff* tetapi juga meningkatkan risiko kesalahan dalam rekapitulasi data absensi.

Untuk meningkatkan efisiensi, diperlukan sebuah aplikasi berbasis *website* untuk sistem absensi karyawan di *Tribe* *TCO* PT Telkom Indonesia. Aplikasi ini akan memungkinkan *staff* dan karyawan untuk melakukan absensi serta rekapitulasi secara efektif. Diharapkan dengan adanya aplikasi ini, proses absensi dan rekapitulasi data absensi dapat dilakukan secara fleksibel, tanpa harus secara manual memasukkan data ke dalam sistem.

Dengan demikian, aplikasi ini diharapkan dapat mempermudah proses absensi dan rekapitulasi bagi semua pihak yang terlibat, serta mengurangi potensi kesalahan dalam pengelolaan data absensi karyawan.

1. Analisa Kebutuhan Alat & Perangkat

Setelah menganalisa masalah, tentunya perlu dilakukan analisis kebutuhan yaitu menganalisis apa saja yang dibutuhkan dalam pembuatan website ataupun dalam penggunaannya. Analisis kebutuhan ini dibagi menjadi 3 (tiga), yaitu kebutuhan software, kebutuhan hardware, dan kebutuhan brainware.

* + 1. Kebutuhan Perangkat Keras (Hardware)

Pengertian hardware adalah segala piranti atau komponen dari sebuah komputer yang sifatnya bisa dilihat secara kasat mata dan bisa diraba secara langsung. Dengan kata lain hardware merupakan komponen yang memiliki bentuk nyata. Spesifikasi *hardware* yang digunakan dalam pembuatan sistem absensi karyawan dijelaskan pada tabel 2.1 di bawah ini:

Tabel 2. 1 Analisa kebutuhan perangkat keras (*Hardware*)

|  |  |  |
| --- | --- | --- |
| **No.** | **Jenis** | **Keterangan** |
|  | *Processor* | AMD Ryzen 7 5800H, Up to 4.4GHz |
|  | RAM | 8GB |
|  | *Storage* | 1TB |

Seperti itu, spesifikasi yang dibutuhkan sesuai dengan tabel 4.1 diatas. Selain itu, juga dibutuhkan perangkat pendukung lainnya seperti *keyboard*, *mouse*, dan *monitor*.

* + 1. Kebutuhan Perangkat Lunak (*Software*)

Selain *Hardware* (perangkat keras) dibutuhkan juga *software* (perangkat lunak) yang dapat menunjang penggunaan perangkat keras tadi. Spesifikasi *software* yang digunakan dalam pembuatan *website* ini, diantaranya :

1. Windows 11 sebagai sistem operasi yang digunakan dalam proses pengerjaan aplikasi berbasis *web*.
2. Visual Studio Code versi 1.85.2 merupakan *software* yang digunakan untuk penulisan *script* *HTML,* *CSS* maupun *Javascript* dalam pembuatan *website* ini.
3. Javascript sebagai bahasa pemrograman yang digunakan untuk menjalankan *framework* Next*.*JS14.1.0.
4. Chakra UI sebagai *library* Javascript yang menyediakan berbagai macam komponen *user* *interface*.
5. Prisma ORM sebagai *Object-relational* *mapping* yang berfungsi menginisialisasi dan memetakan struktur data.
6. Postgresql yang difungsikan sebagai *database* dan di authentikasi dan dihubungkan melalui *library Axios*.
7. Draw.io digunakan untuk membuat perancangan *Flowmap* dan *Data Flow Diagram.*
8. Canva adalah *platform* desain grafis yang digunakan untuk membuat grafis media sosial, presentasi, poster, dokumen dan konten visual lainnya. Pada laporan ini menggunakan canva untuk membuat *layout* (*mockup*) untuk desain *website*.
9. *Web browser* digunakan untuk menampilkan apa yang dihasilkan dari script yang telah dibuat. Selain itu, juga digunakan untuk mencari referensi dan dokumentasi terkait pengerjaan *website*.
   * 1. Kebutuhan *Brainware*

*Brainware* adalah orang yang menggunakan atau mengoperasikan sebuah perangkat komputer. Dengan kata lain, *brainware* adalah pengguna/*user* yang menggunakan atau memanajemen atau mengelola *website* ini. Pengguna atau *admin* *website* tentunya harus dapat mengoperasikan komputer dengan baik. *Brainware* pada pembuatan *website* absensi karyawan ini dibagi menjadi dua (2) jenis *user*, berikut di bawah ini penjelasannya:

1. Admin memiliki hak akses untuk mengelola data absensi karyawan. Admin juga dapat melihat data absensi dan *men-download* laporan hasil rekap *daily summary* dan *monthly summary*.
2. User memiliki hak akses hanya untuk melukan absensi, yaitu mengisi *form* yang ada di *website*.
3. Perancangan Sistem Absensi Karyawan

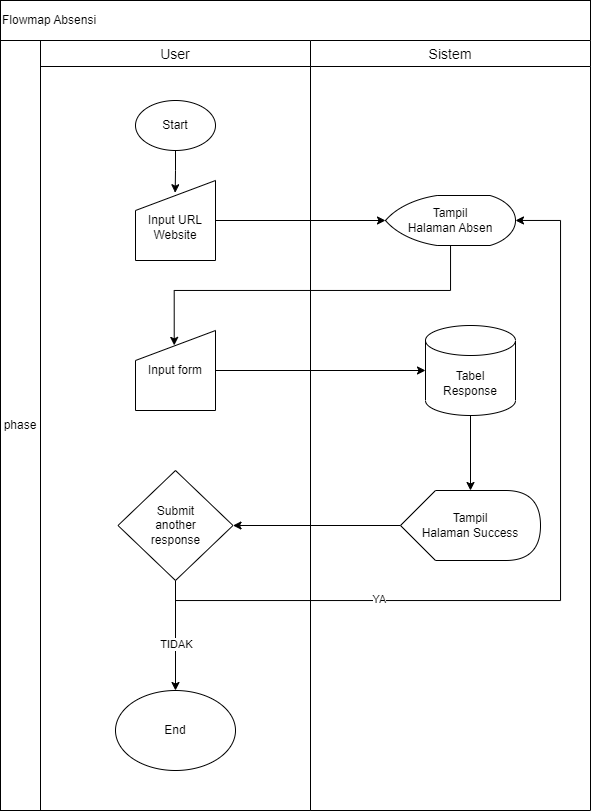
Perancangan sistem adalah suatu proses menyusun atau mengembangkan sistem informasi yang baru. Perancangan sistem merupakan tahap selanjutnya setelah Analisa sistem. Setelah mendapatkan gambaran dengan jelas tentang apa yang dikerjakan pada Analisa sistem, dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut agar lebih efektif. Hasil sistem yang dirancang harus sesuai dengan kebutuhan pemakai untuk mendapatkan informasi.

Selanjutnya pada perancangan sistem ini akan dijelaskan mengenai, perancangan flowmaр, perancangan Data Flow Diagram (DFD), perancangan Layout (Mock Up), dan perancangan basis data. Berikut penjelasannya :

* + 1. Perancangan *Flowmap*

1. *Flowmap* absensi

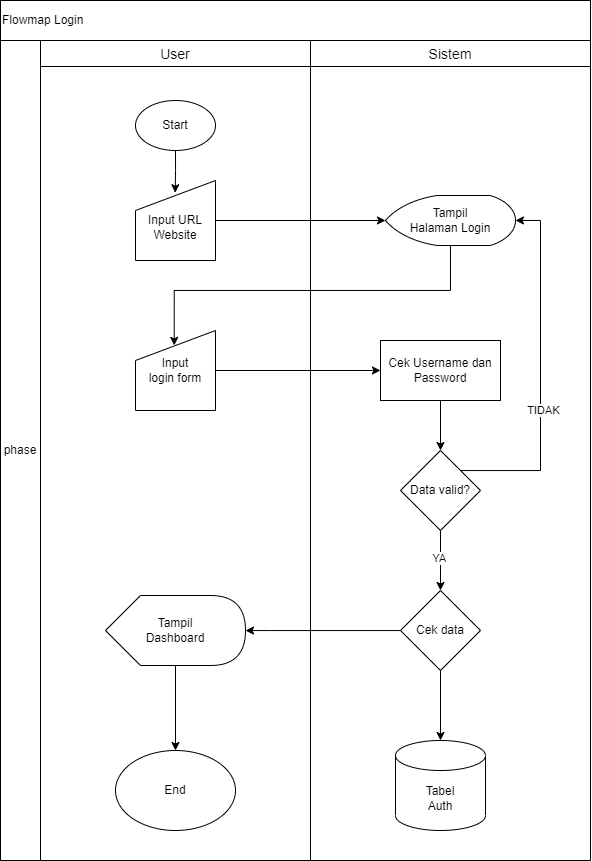
Berikut merupakan *flowmap* dari sistem absensi yang ditunjukan pada gambar 2.1 di bawah.



Gambar 2. 1 *Flowmap* Absensi

1. *Flowmap* *login*

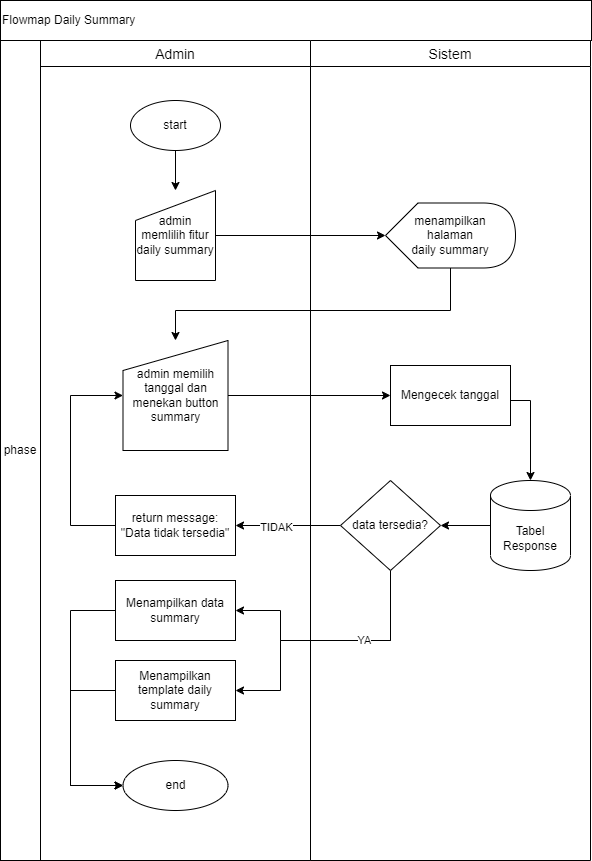
Berikut merupakan *flowmap* dari sistem *login* yang ditunjukan pada gambar 2.2 di bawah.



Gambar 2. 2 *Flowmap* *Login*

1. *Flowmap daily summary*

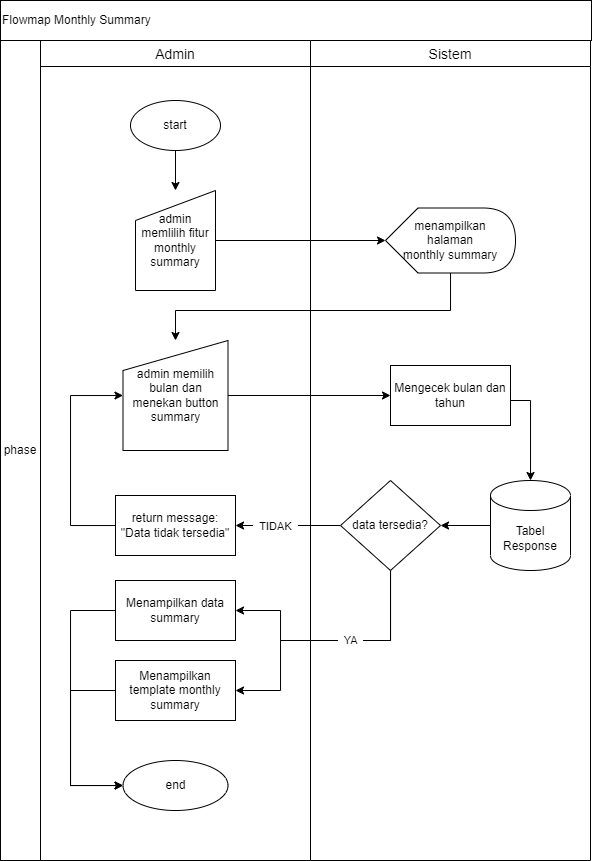
Berikut merupakan *flowmap* dari fitur *daily summary* yang ditunjukan pada gambar 2.3 di bawah.



Gambar 2. 3 *Flowmap* *Daily* *Summary*

1. *Flowmap monthly summary*

Berikut merupakan *flowmap* dari fitur *Monthly summary* yang ditunjukan pada gambar 2.4 di bawah.

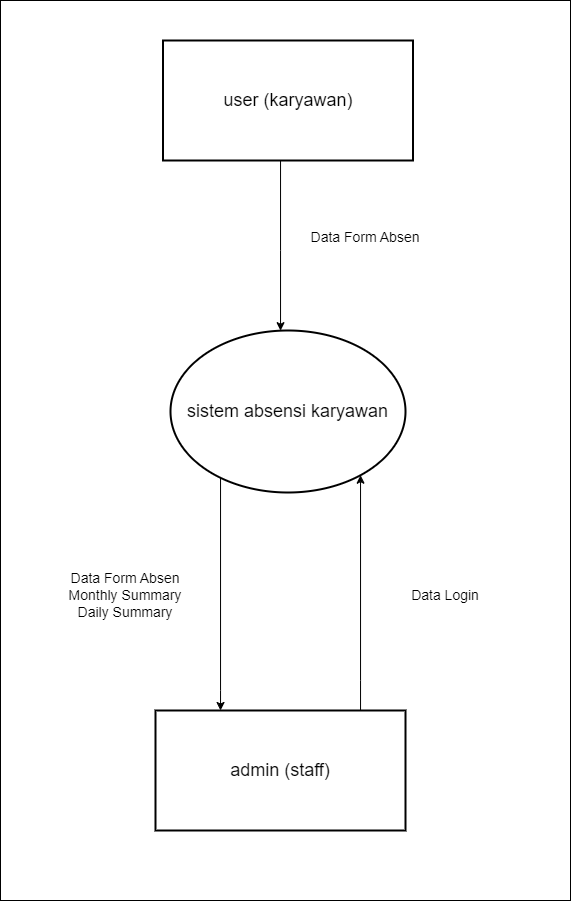


Gambar 2. 4 *Flowmap* *Monthly* *Summary*

* + 1. Perancangan *Data* *Flow* *Diagram*

Salah satu model dari pembahasan *website* sistem absensi karyawan ini dengan menggunakan pemodelan *Data* *Flow* *Diagram* (DFD) level 0 dan DFD level 1.

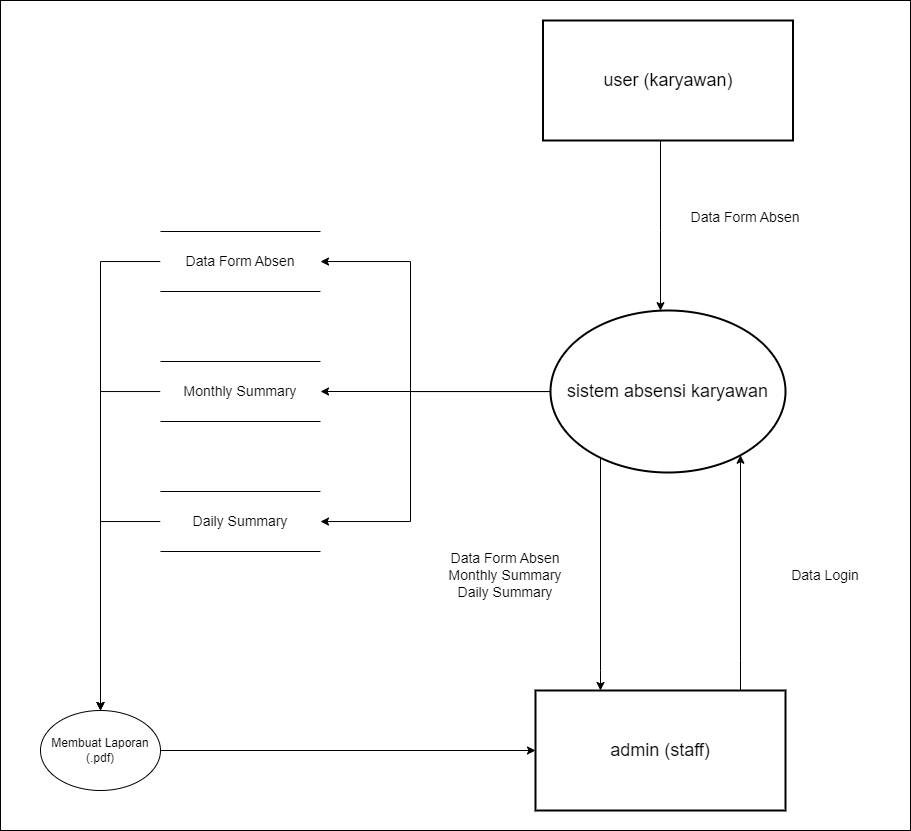
Untuk DFD yang pertama yaitu *context* *diagram*. Berikut pada gambar 2.5 di bawah ini adalah gambar dari *context* *diagram*:



Gambar 2. 5 DFD *Level* 0

Berdasarkan pada gambar 2.5 *context diagram* diatas, menunjukan alur bahwa admin dapat melakukan pengelolaan absensi yaitu *Daily Summary* dan *Monthly Summary*. Untuk user dapat memberikan data form absen pada aplikasi.

Selanjutnya adalah DFD *level* 1, yang dimana semua proses yang ada pada DFD *level* 0 dirinci dengan lengkap sehingga lebih lengkap dan detail. Proses-proses utama yang ada akan dipecah menjadi sub proses. Berikut pada gambar 2.6 di bawah ini adalah gambar dari DFD *level* 1:



Gambar 2. 6 DFD *Level* 1

Berdasarkan pada gambar 2.6 diatas, menunjukan bahwa DFD *level* 1 ini merupakan pengenmbangan lebih lanjut dari DFD *level* 0. Pada DFD *level* 1 ini terlihat bahwa *admin* dapat mengelola data *form* absen, data *daily* *summary* dan data *monthly* *summary*.

1. Perancangan *Database*

Perancangan database atau basis data merupakan proses yang bertujuan untuk menentukan isi serta pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Salah satu tahap yang dilakukan dalam perancangan basis data yaitu pembuatan tabel. Berikut merupakan struktur tabel yang digunakan pada pembuatan Website Absensi Karyawan.

1. Tabel Absensi

Tabel ini berfungsi sebagai tempat penyimpanan data dari *user* yang melakukan *input* pada *form* absensi. Tabel absensi terdiri beberapa *field*, seperti id, *createdAt*, *email*, *date*, *name*, *employeeStatus*, *squad*, *condition*, *conditionDesc*, *workFrom*, *location*, dan *workPlan*. Berikut merupakan struktur tabel absensi yang ditunjukan pada tabel 2.2.

Tabel 2. 2 Perancangan basisi data tabel absensi

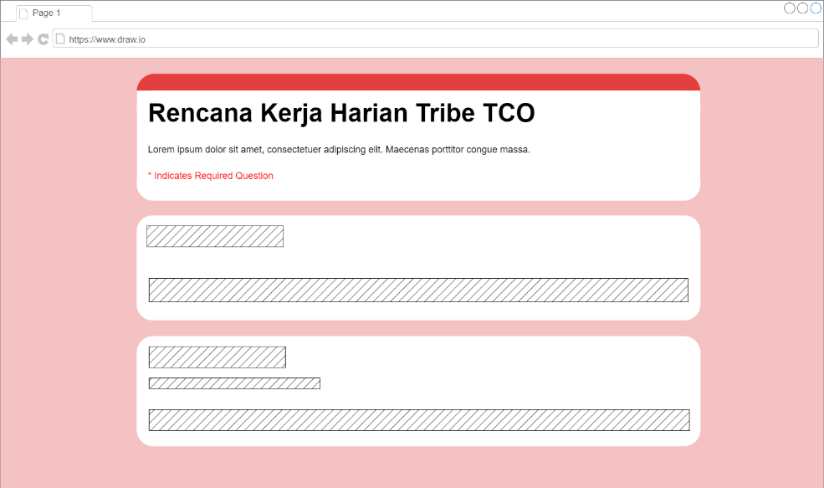
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Absensi** | | | | |
| **Field** | **Type** | **Null** | **Key** | **Extra** |
| id | *int*(11) | *NO* | *PRIMARY* *KEY* | *AUTO\_INCREMENT* |
| *createdAt* | *Date* | *NO* |  | *DEFAULT*(*NOW*) |
| *email* | *varchar*(128) | *NO* |  |  |
| *date* | *Date* | *NO* |  |  |
| *name* | *varchar*(128) | *NO* |  |  |
| *employeeStatus* | *varchar*(128) | *NO* |  |  |
| *squad* | *varchar*(128) | *NO* |  |  |
| *condition* | *varchar*(128) | *NO* |  |  |
| *conditionDesc* | *varchar*(128) | *NO* |  |  |
| *workFrom* | *varchar*(128) | *NO* |  |  |
| *location* | *varchar*(128) | *NO* |  |  |
| *workPlan* | *varchar*(128) | *NO* |  |  |

1. Perancangan Layout (Mock Up)

Desain layout atau tampilan antarmuka berfungsi untuk menentukan bagaimana tampilan dari website sistem pengajuan permintaan barang.

1. Tampilan Halaman Absensi

Berikut merupakan tampilan perancangan *layout* *mockup* halaman absensi yang akan ditampilkan padagambar 2.7 di bawah ini.

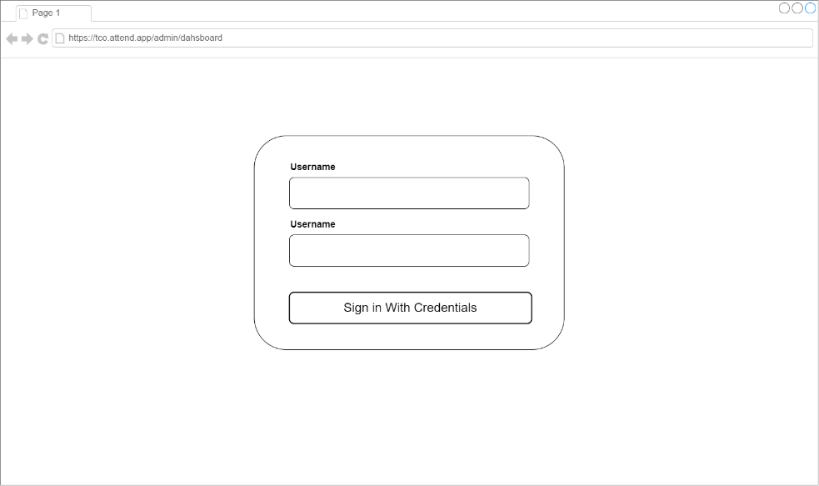


Gambar 2. 7 *Layout* Halaman Absensi

1. Tampilan Halaman *Login*

Tampilan halaman login terdiri dari form username dan password, tombol login yang akan mengalihkan ke halaman dashboard.

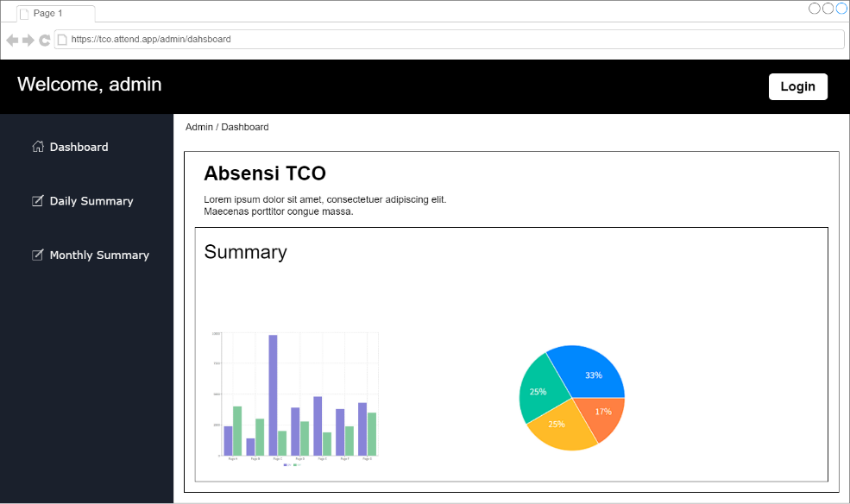
Desain tampilan halaman login dapat dilihat pada gambar 2.8.



Gambar 2. 8 *Layout* Halaman *Login*

1. Tampilan Halaman *Dashboard* *Admin*

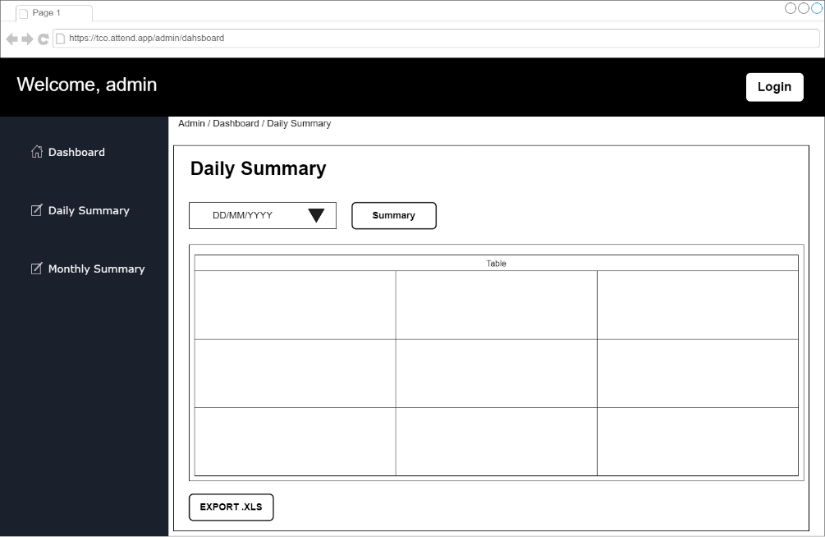
Halaman *dashboard* merupakan halaman yang akan tampil setelah *admin* berhasil *login* ke dalam sistem. Desain tampilan *dashboard* *admin* dapat dilihat pada gambar 2.9.



Gambar 2. 9 *Layout* Halaman *Dashboard* *Admin*

1. Tampilan Halaman *Daily* *Summary*

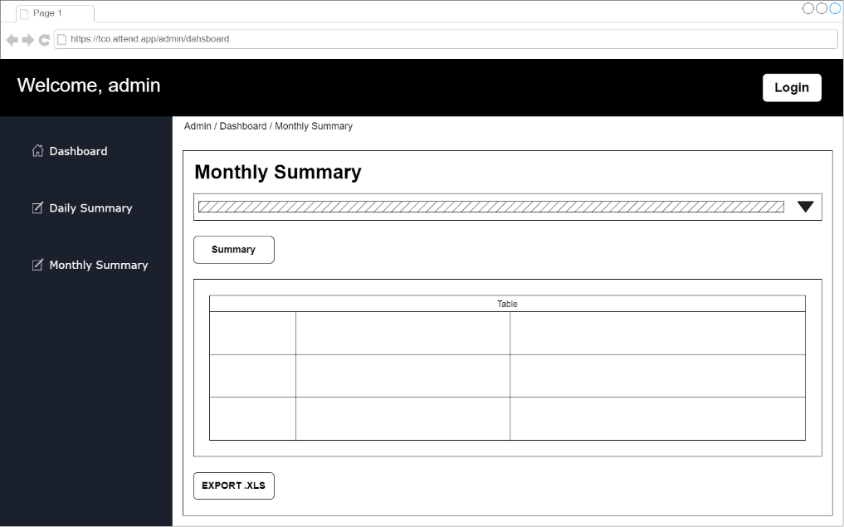
Tampilan halaman daily summary adalah halaman yang menampilkan *field* data *response* yang difilter berdasarkan tanggal, bulan dan tahun*.* Terdapat fitur *input* untuk menentukan tanggal. Desain tampilan halaman *daily* *summary* dapat dilihat pada gambar 2.10.



Gambar 2. 10 *Layout* Halaman *Daily* *Summary*

1. Tampilan Halaman Monthly Summary

Tampilan halaman monthly summary adalah halaman yang menampilkan field data response yang difilter berdasarkan bulan dan tahun. Terdapat fitur input untuk menentukan bulan. Desain tampilan halaman monthly summary dapat dilihat pada gambar 2.11.

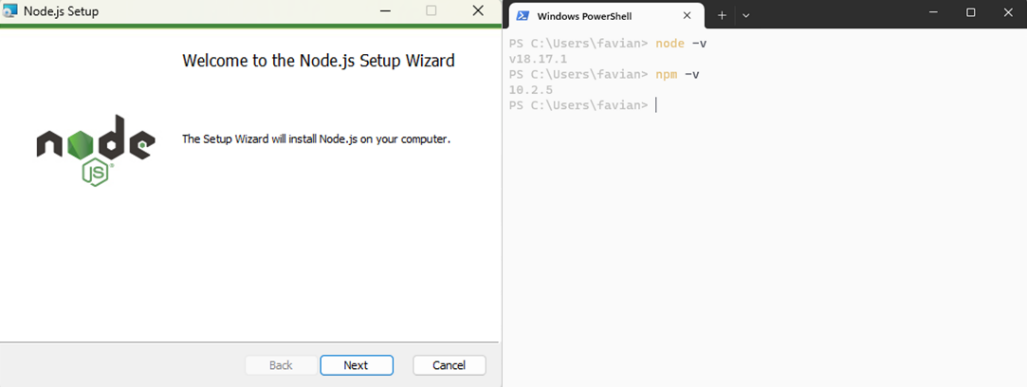


Gambar 2. 11 *Layout* Halaman *Monthly* *Summary*

1. Persiapan Pembuatan Aplikasi

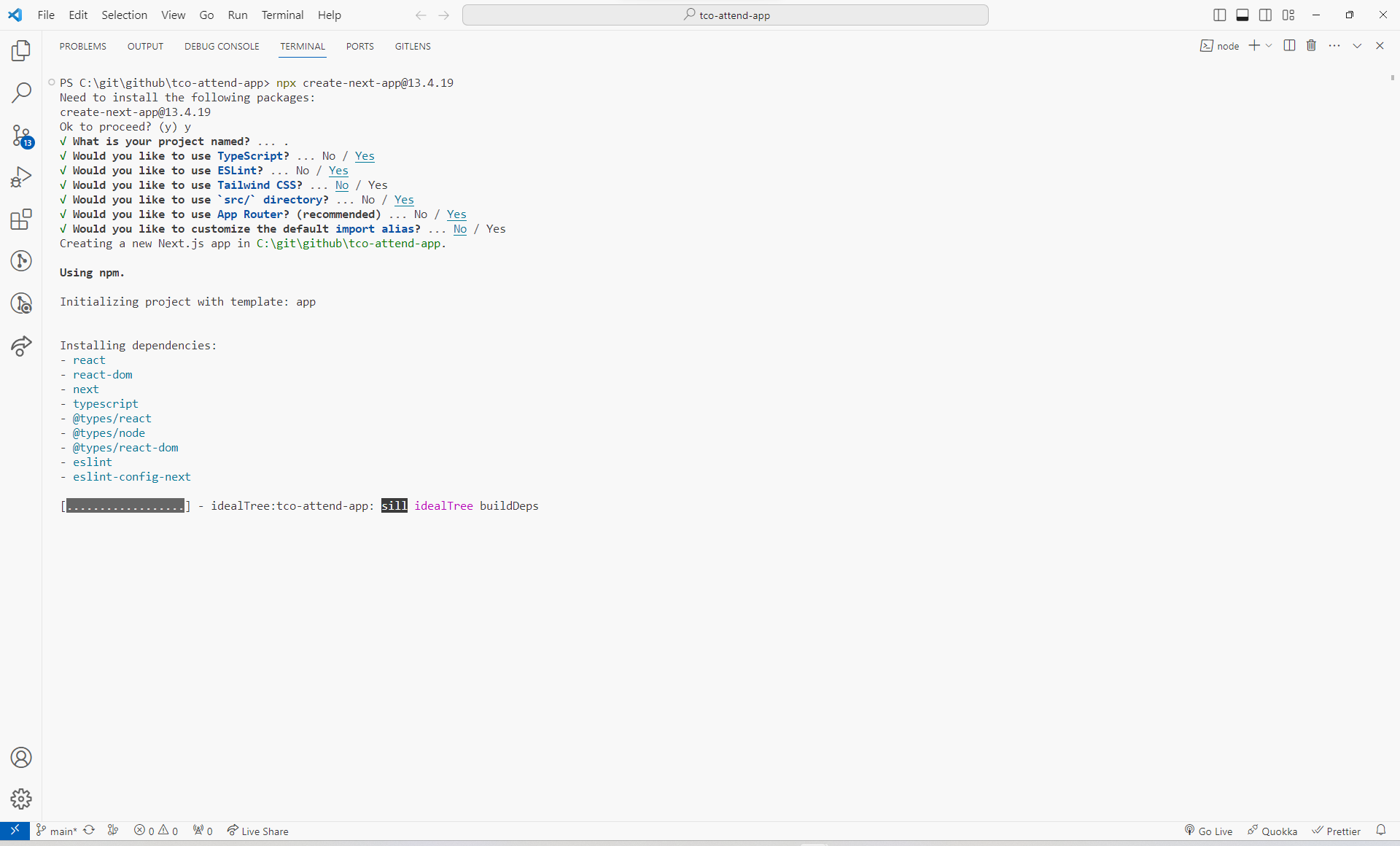
Dalam pembuatan aplikasi sistem absensi karyawan berbasis *website* menggunakan *framework* *Next*.*js* 14.1.0. Berikut penjelasan mengenai tahapan yang dilakukan dalam membuat aplikasi berbasis *website* ini:

1. Menginstal terlebih dahulu dan menggunakan Node.JS untuk menginstal beberapa library seperti next.js dan library lainnya. Berikut proses instalasi node.js beserta pengecekan versi Node.JS dapat dilihat di gambar 2.12.



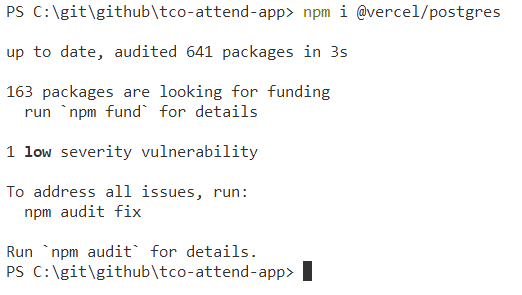
Gambar 2. 12 Instalasi Node.JS

1. *Setup* *project* dengan membuka *visual* *studio* *code* lalu pada *terminal* ketikan “npx create-next-app@14.1.0”. Pada setup project kali ini penulis memilih menggunakan *Typescript*, *ESLint*, direktori src dan menggunakan *App* *Router*. Setup project bisa dilihat pada gambar 2.13 di bawah ini.



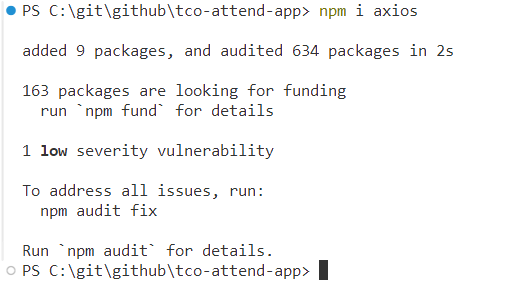
Gambar 2. 13 *Setup* *Project*

1. Menginstal beberapa *library* dan dependensi melalui *terminal*. Library dan dependensi yang disebut diantaranya:
   1. *Postgres*, hal pertama yang penulis adalah melakukan instalasi *postgres* sebagai *client* *library,* untuk menghubungkan *postgresql* dengan project seperti pada gambar 2.14.



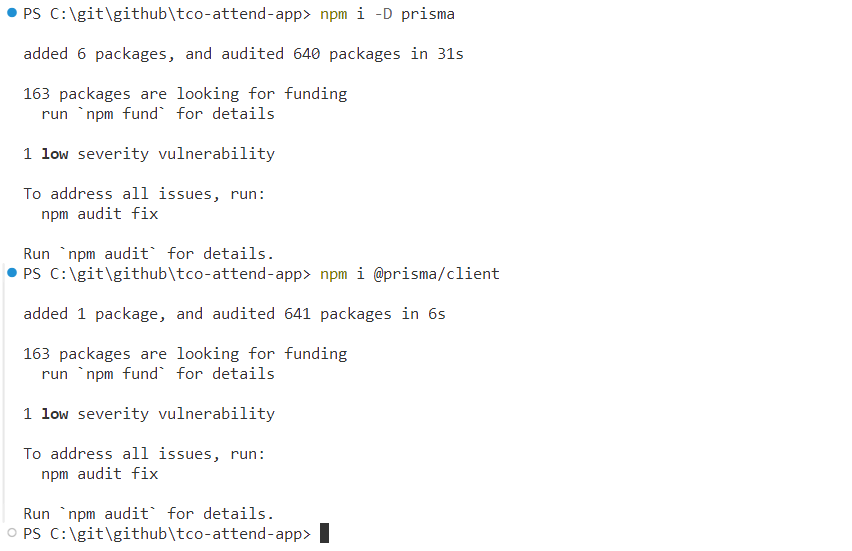
Gambar 2. 14 Instalasi *library* *postgres*

* 1. Berikutnya menginstal library Axios. Library ini berfungsi untuk menjembatani permintaan dari website ke *database* untuk melakukan reading ataupun writing. Instalasi library Axios dapat dilihat pada gambar 2.15.



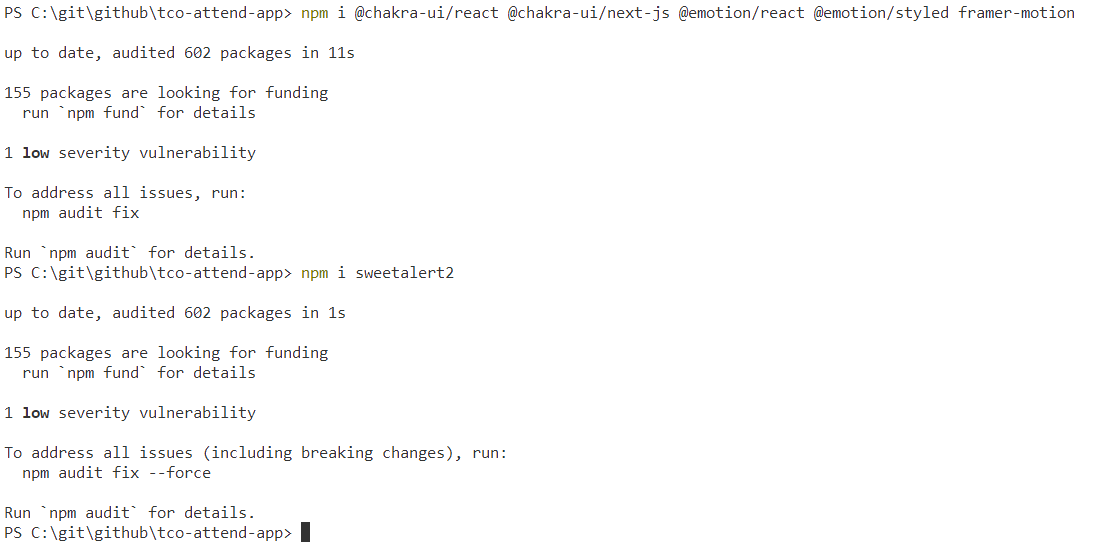
Gambar 2. 15 Instalasi library axios

* 1. Setelah instalasi *library Axios* selesai, selanjutnya melakukan instalasi *prisma* dan *prisma client,* untuk melakukan *mapping* pada *database* *postgresql.* Juga digunakan sebagai alat migrasi *database*. Proses instalasi *prisma* dan *prisma* *client* terdapat pada gambar 2.16.



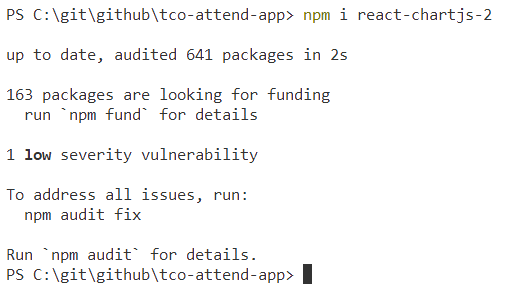
Gambar 2. 16 Instalasi *library* *prisma* dan *prisma* *client*

* 1. Selanjutnya, setelah *prisma* berhasil terinstal, dilanjutkan dengan menginstal *library* *Chakra* *UI* dan *Sweetalert2* sebagai penyedia komponen *user* *interface.* Proses instalasi bisa dilihat pada gambar 2.17 di bawah ini.



Gambar 2. 17 Instalasi *library* *Chakra* *UI* dan *Sweetalert2*

* 1. Setelah selesai menginstal *Chakra* *UI* dan *sweetalert*, dilanjutkan dengan menginstal *library React Chart JS 2* sebagai penyedia komponen diagram untuk mempresentasikan data di *website* ini. Proses instalasi bisa dilihat pada gambar 2.18 di bawah ini.

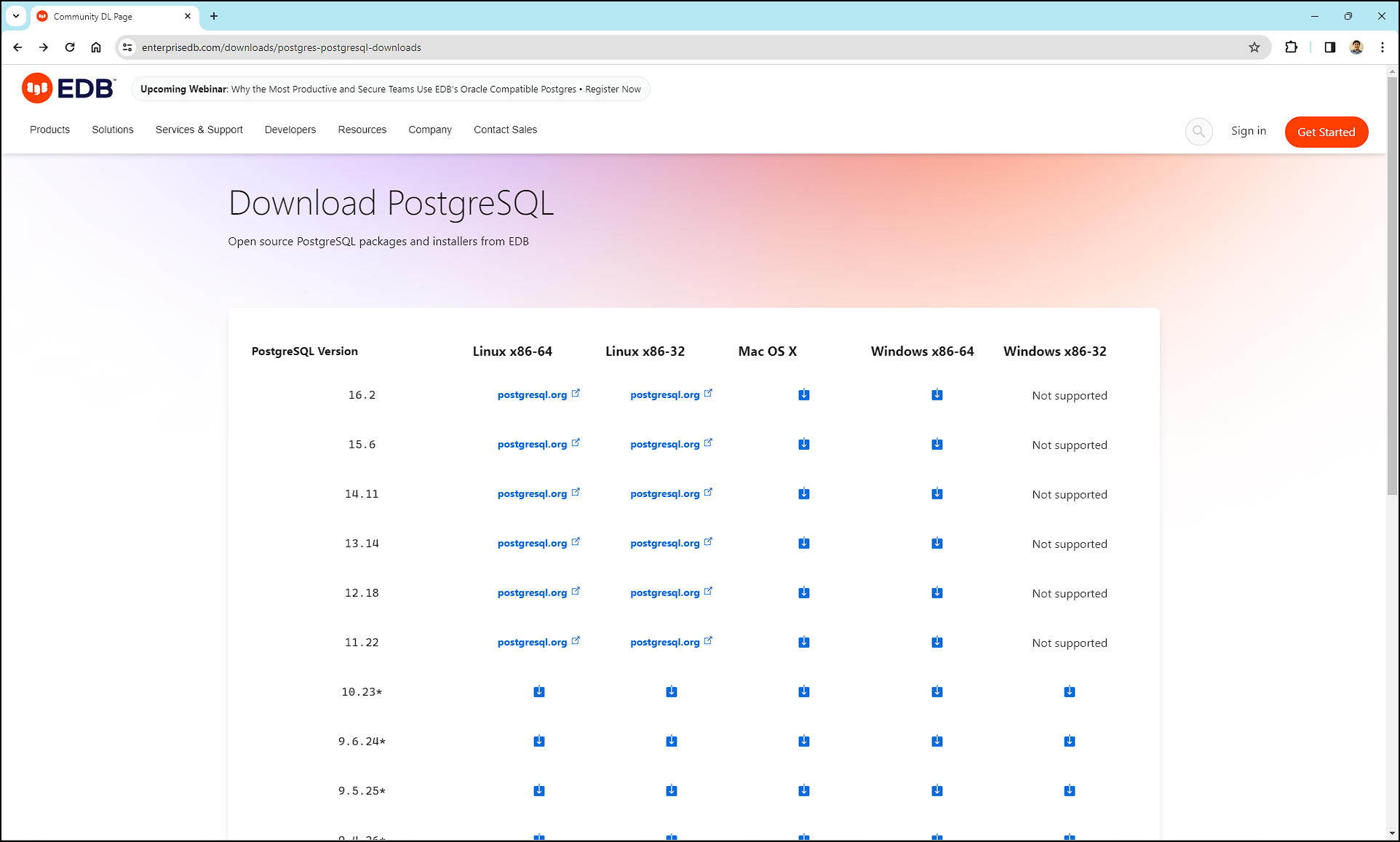


Gambar 2. 18 Instalasi *library* *React Chart JS 2*

* 1. Pembuatan Basis Data

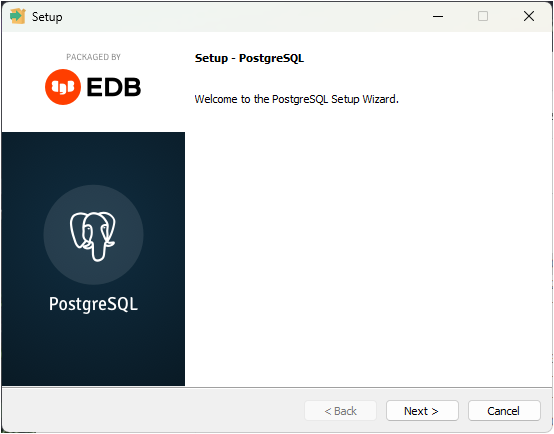
Membuat basis data dengan *PostgreSQL* dan *Prisma* sebagai *ORM* dengan *menginstal* *PostgreSQL* terlebih dahulu.

Hal yang pertama dilakukan adalah mengunduh aplikasi *PostgreSQL* di *website* resmi, pada proyek kali ini penulis mengunduh *PostgreSQL* versi 15.6. Berikut pada gambar 2.19 merupakan proses mengunduh *PostgreSQL* di *website* resmi.

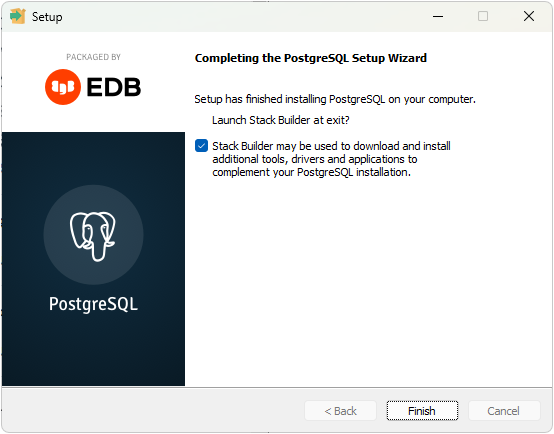


Gambar 2. 19 Mengunduh *PostgreSQL*

Instalasi dilakukan dengan menjalankan aplikasi *setup* *PostgreSQL* yang sudah di unduh sebelumnya. *Setup PostgreSQL* dapat dilihat pada gambar 2.20 dan gambar 2.21.

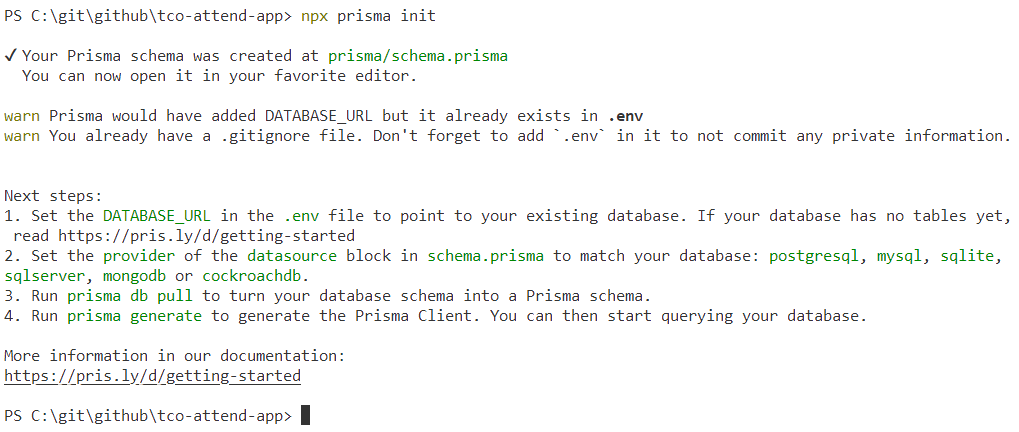


Gambar 2. 20 Instalasi *PostgreSQL*



Gambar 2. 21 Instalasi *PostgreSQL*

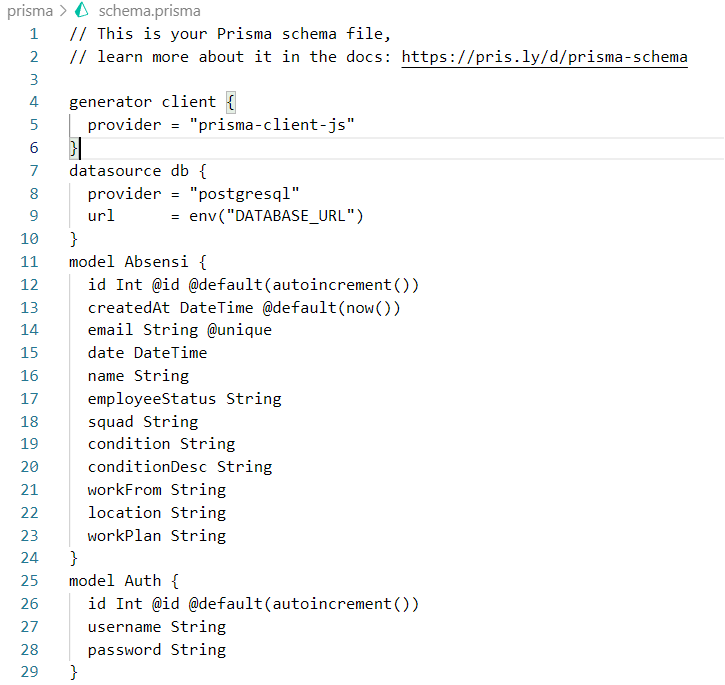
Setelah selesai melakukan instalasi *PostgreSQL*, dilanjutkan dengan melakukan inisialisasi pada *prisma*. Inisialisasi dilakukan dengan mengetikan perintah “*npx* *prisma* *init*” yang ditampilkan pada gambar 2.22.



Gambar 2. 22 Inisialisasi *prisma*

Perintah tersebut akan melakukan inisialisasi yang prosesnya menciptakan *file* baru yaitu *schema*.*prisma* dan .*env*.

Pada *file* *schema*.*prisma* berisikan konfigurasi utama dari *prisma*, di dalamnya terdapat konfigurasi *client* dan *database*. Isi dari *file* *schema.prisma* ditampilkan pada gambar 2.23.



Gambar 2. 23 Isi *file* *schema*.*prisma*

Pada *file* .*env* berisikan *database* *key* untuk menghubungkan *website* dengan *database* *PostgreSQL*. Isi dari *file* .*env* dapat dilihat pada gambar 2.24.



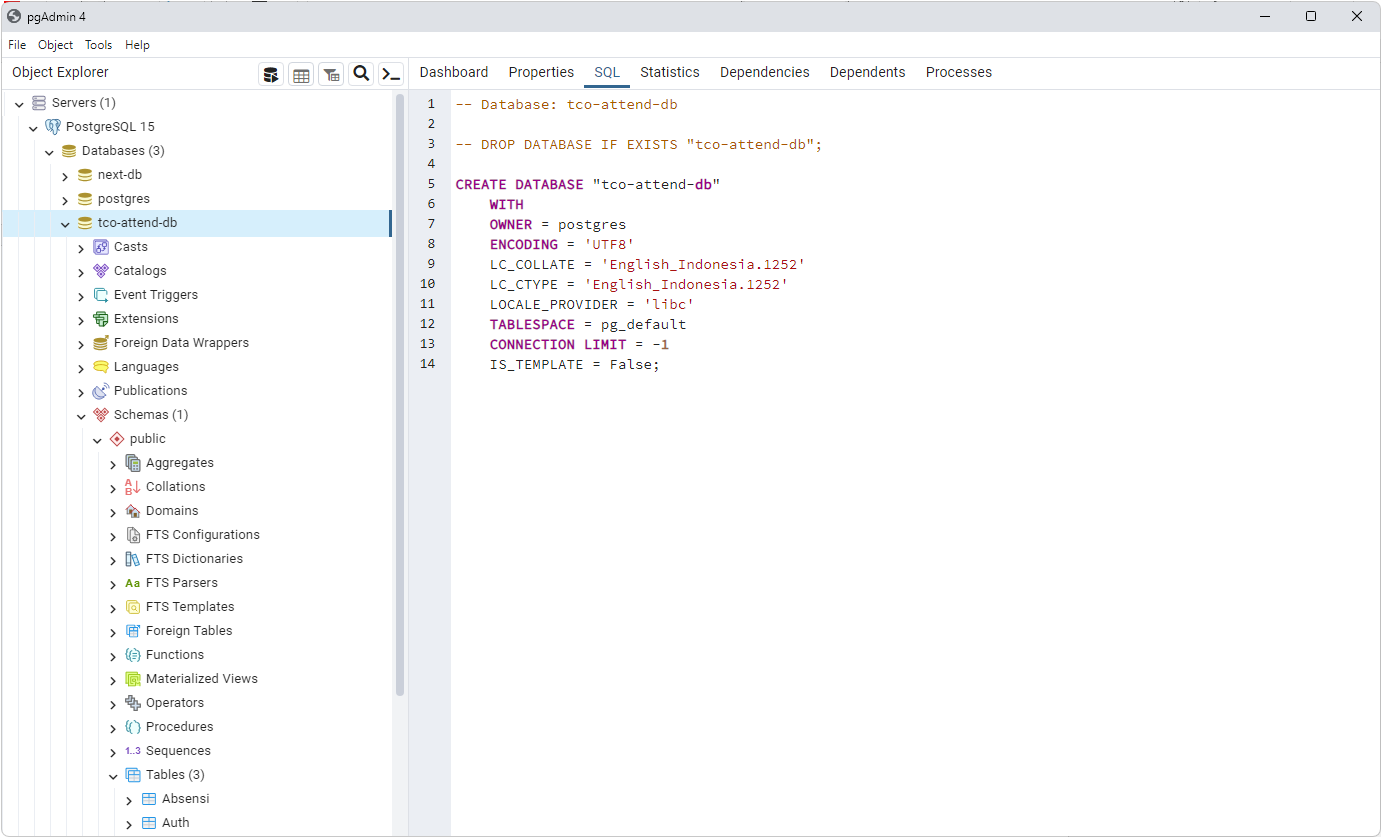
Gambar 2. 24 Isi *file* .*env*

Selanjutnya adalah melakukan migrasi *database* menggunakan *prisma*. Migrasi *prisma* ini bertujuan untuk mengeksekusi *file* *schema*.*prisma*, membuat *database* baru pada *PostgreSQL* yang berdasarkan konfigurasi dari *file* *schema*.*prisma* tersebut. Untuk melakukan migrasi menggunakan *prisma* adalah dengan mengetikan “*npx* *prisma* *migrate* *dev*” pada *terminal*. Proses migrasi bisa dilihat pada gambar 2.25.



Gambar 2. 25 Proses migrasi *database* menggunakan *prisma*

Pada saat melakukan migrasi *Prisma* akan membuat *database* baru pada *PostgreSQL* dengan nama *database* *tco-attend-db* yang berjalan pada *localhost:5232.* Berikut pada gambar 2.26 menampilkan *database* pada aplikasi *PostgreSQL.*



Gambar 2. 26 *Database* pada *PostgreSQL*

1. Pembuatan Antarmuka *Website*

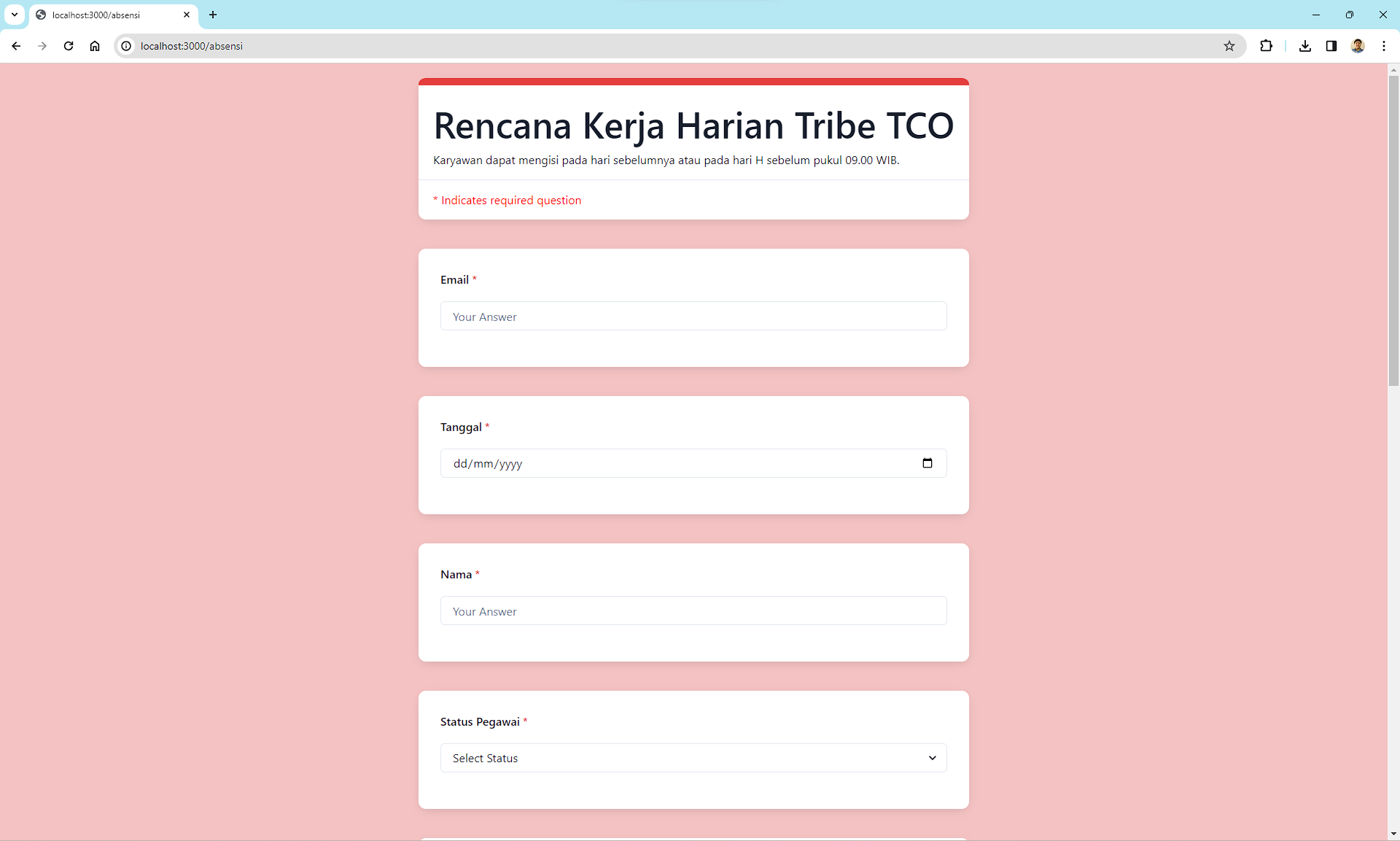
Implementasi antarmuka menggambarkan tampilan dari aplikasi yang dibangun sistem absensi karyawan berbasis *website*. Berikut merupakan implementasi antarmuka pada *web* absensi karyawan.

* + 1. Halaman Absensi

Halaman absensi merupakan halaman yang menampilkan *form* pengisian absensi. Pada halaman absensi terdapat dua halaman berbeda, yaitu halaman absensi dan halaman sukses. Halaman absensi merupakan halaman yang bertujuan untuk karyawan mengisikan form-form yang diperlukan. Terdapat beberapa form diantaranya :

1. *Form* *email* adalah *form* yang nantinya akan diisi oleh pengguna berupa teks *email*. *Form* *email* ini membutuhkan tanda "@" atau "at" saat diisi.
2. *Form* tanggal adalah *form* yang mengharuskan pengguna memasukkan tanggal sebagai *input*.
3. *Form* nama adalah *form* yang mengharuskan pengguna memasukkan nama lengkap.
4. *Form* status pegawai merupakan *form* *dropdown* yang mengharuskan pengguna memilih beberapa opsi, antara lain Karyawan Tetap, *Digital Talent* atau *Pro Hire*, dan Tenaga Kerja Penunjang.
5. *Form* *squad* adalah *form* *dropdown* yang mengharuskan pengguna memilih beberapa opsi, antara lain *Tribe Leader, Business Collaboration, Product Onboard Business Operational, dan Project & Platform*.
6. *Form* kondisi badan adalah *form dropdown* yang mengharuskan pengguna memilih beberapa opsi, di antaranya Sehat, Kurang Sehat (masih bisa bekerja), dan Sakit (perlu istirahat).
7. *Form* keterangan kondisi merupakan *form* lanjutan dari *form* kondisi yang mengharuskan pengguna untuk menjelaskan kondisinya saat mengisi formulir tersebut. Jika pengguna dalam keadaan sehat, maka pengguna hanya perlu memilih opsi "sehat" yang tertera pada *input* *radio*.
8. *Form* informasi kerja adalah *form* yang mengharuskan pengguna memilih beberapa opsi, di antaranya *Work From Office* (WFO) atau *Customer* *Visit*, *Work From Anywhere* (WFA), dan Cuti.
9. *Form* lokasi dan kota adalah *form* yang mengharuskan pengguna mengisi lokasi saat mengakses *website* tersebut.
10. *Form* rencana kerja atau aktivitas adalah *form* yang mengharuskan pengguna mengisi rencana kerja atau aktivitas pada hari itu.

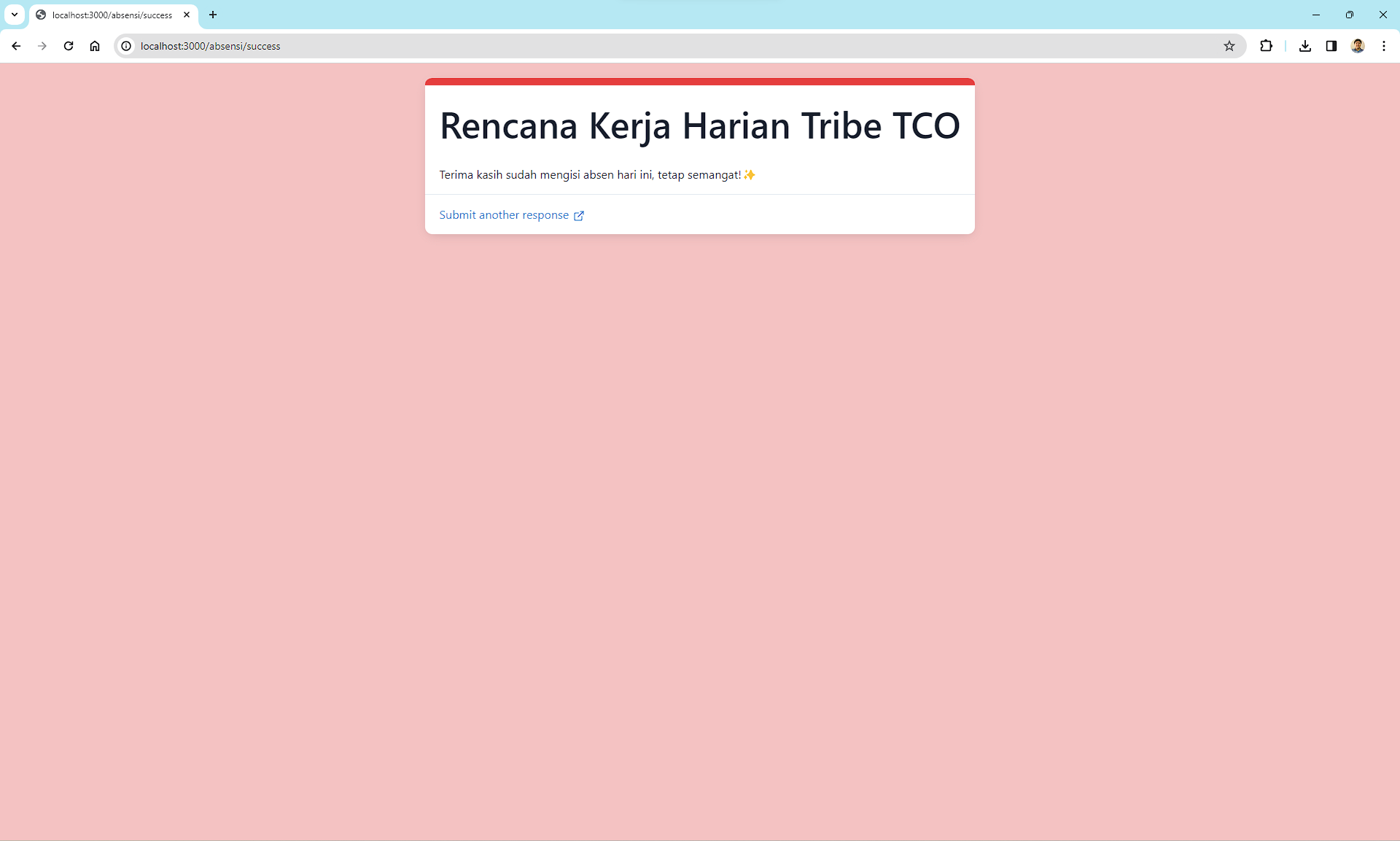
Berikut pada gambar 2.27 merupakan penampilan dari halaman *form* absensi.



Gambar 2. 27 Halaman *form* absensi

Pada halaman sukses merupakan halaman yang menampilkan pesan bahwa absensi terlah berhasil di rekam. Halaman ini muncul setelah pengguna selesai memasukan semua *form* yang ada pada halaman absen dan setelah pengguna menekan tombol *submit*.

Berikut pada gambar 2.28 adalah tampilan halaman sukes jika karyawan berhasil melakukan *submit* pada halaman absen.

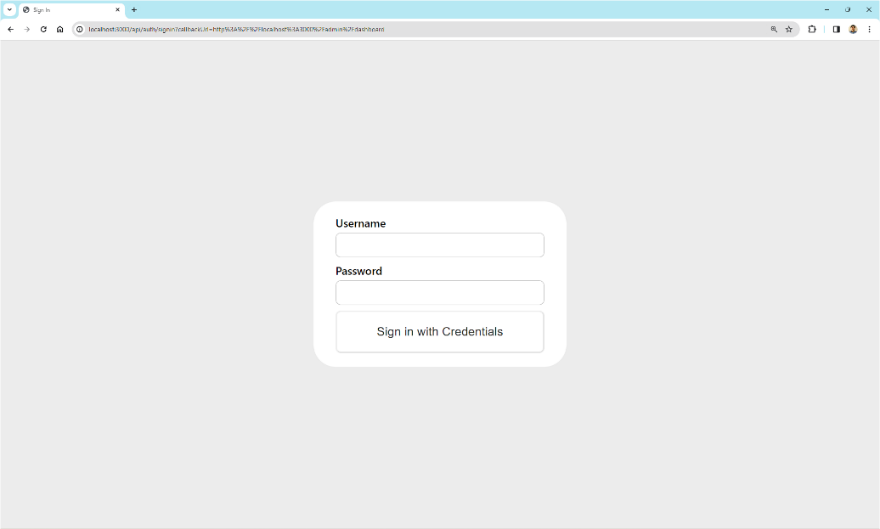


Gambar 2. 28 Halaman sukses ketika berhasil absen

* + 1. Halaman *Login*

Halaman *login* adalah halaman yang muncul ketika pengguna mencoba untuk masuk sebagai *admin*. Di halaman ini, pengguna diminta untuk memasukkan *username* dan *password*. Jika pengguna berhasil melakukan *login*, mereka akan dialihkan ke halaman selanjutnya, yaitu halaman *dashboard*.

Berikut pada gambar 2.29 adalah tampilan dari halaman *login*.

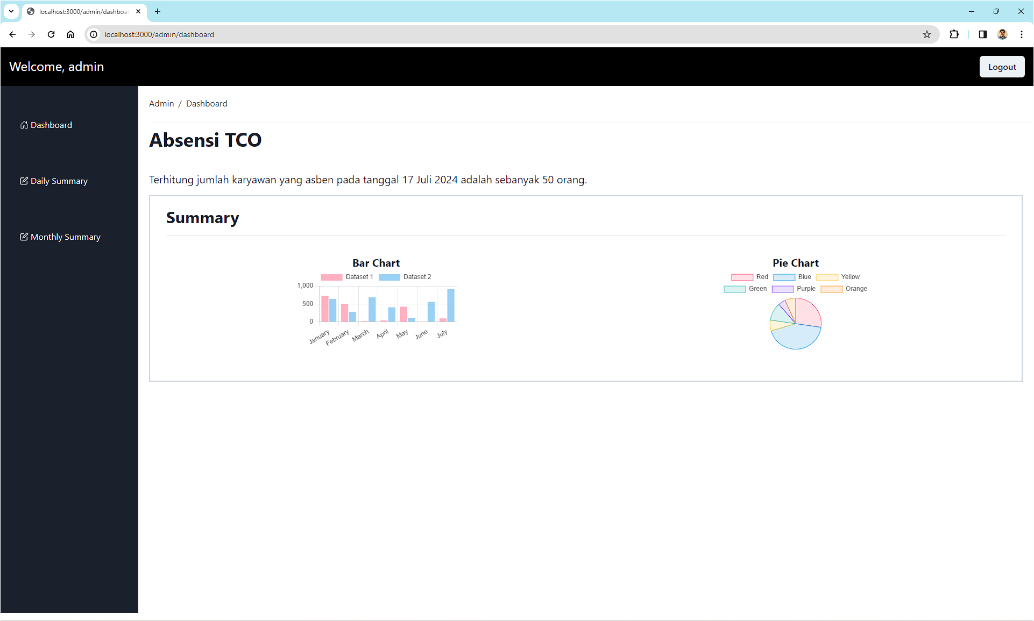


Gambar 2. 29 Halaman *login*

* + 1. Halaman Dashboard

Halaman *Dashboard* merupakan halaman yang ditampilkan pertama kali jika pengguna berhasil *login* sebagai *admin*. Pada halaman *dashboard* terdapat *navbar* yang berada paling atas tampilan *website* dan *sidebar* pada bagian kiri tampilan *website* serta bagian utama yang menampilkan judul “Absensi TCO”.

Berikut pada gambar 2.30 merupakan tampilan dari halaman dashboard.



Gambar 2. 30 Tampilan halaman *dashboard*

Pada bagian kiri *navbar* menampilkan pesan “*Welcome, admin*” dan pada bagian kanan *navbar* terdapat sebuah tombol *logout* untuk *admin* melakukan *logout*.

Pada *sidebar* terdapat tiga navigasi, yaitu diantaranya *dashboard* untuk mengarah ke halaman *dashboard*, *daily* *summary* untuk mengarah ke halaman *daily* *summary*, dan tombol *monthly* *summary* untuk mengarah ke halaman *monthly* *summary*.

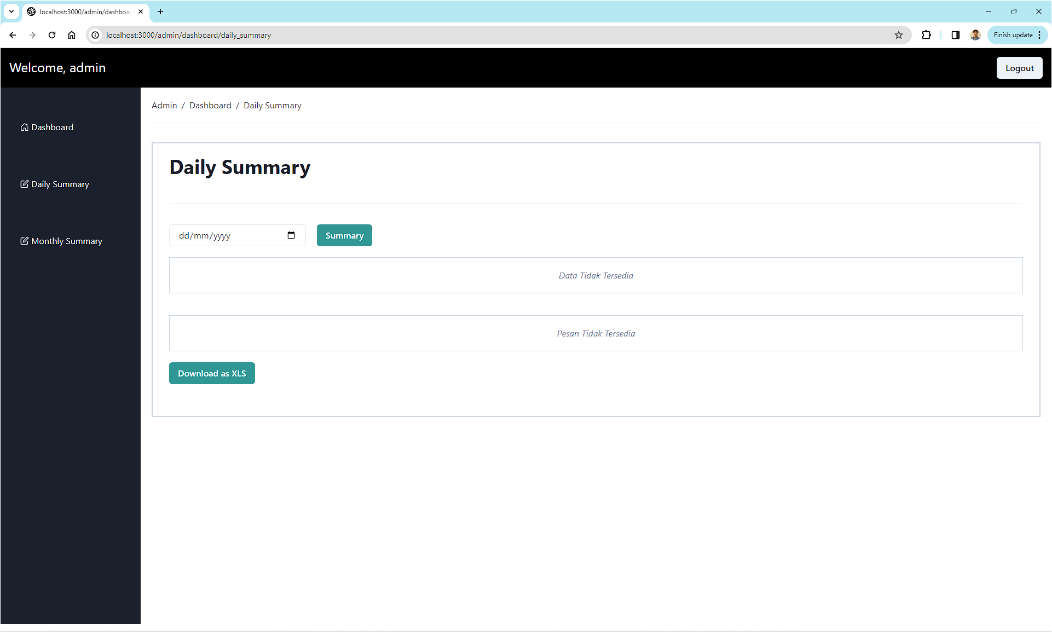
Pada bagian utama menampilkan sebuah *breadcrumb*. *Breadcrumb* adalah elemen *web* yang memiliki fungsi sebagai navigasi halaman di dalam *web*. Selain *breadcrumb*, terdapat judul halaman yang bertuliskan "Absensi TCO", yang diikuti oleh keterangan mengenai jumlah karyawan yang telah mengisi *form* absensi pada tanggal sekian. Di samping itu, terdapat juga sebuah summary yang menampilkan dua diagram data, yaitu diagram batang dan diagram *pie*.

Diagram batang menunjukkan total jumlah karyawan yang telah mengisi *form* absensi berdasarkan hari atau tanggal. Sementara itu, diagram *pie* menampilkan jumlah karyawan yang mengisi *form* absensi, yang diurutkan berdasarkan *squad*-nya.

* + 1. Halaman Daily Summary

Halaman *diary* *summary* merupakan halaman bagian dari *dashboard*. Pada halaman *daily* *summary* ini terdapat sebuah *navbar* dan *sidebar* yang sama dengan yang ada pada halaman *dashboard*.

Berikut pada gambar 2.31 merupakan tampilan dari halaman *daily* *summary*.



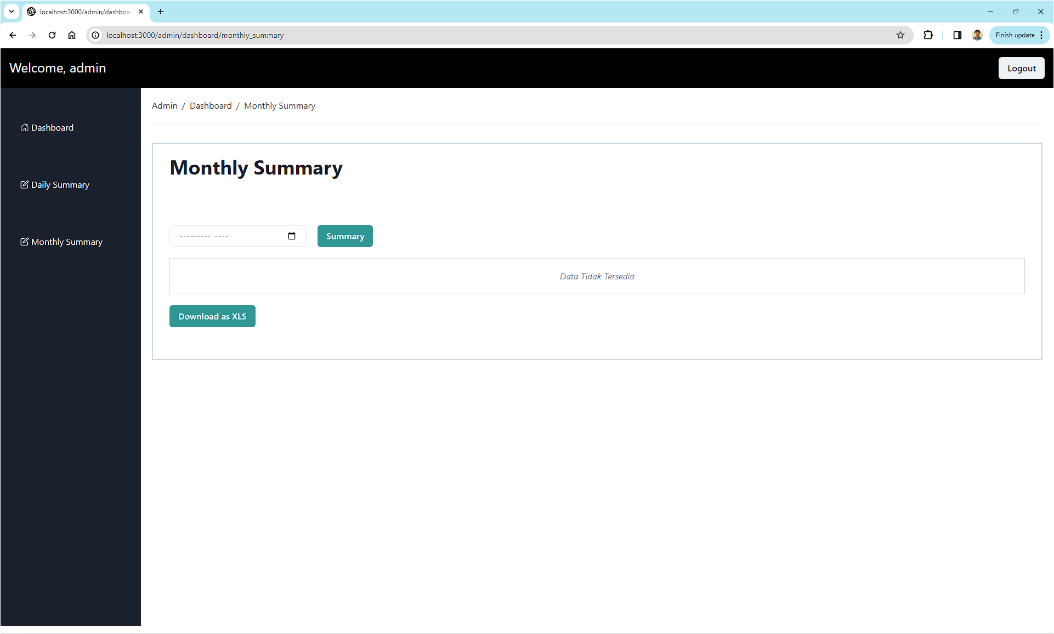
Gambar 2. 31 Tampilan halaman daily summary

Pada halaman utama dari *Daily* *Summary* ini menampilkan *breadcrumb* yang sama dengan halaman *dashboard*. Selain itu, pada halaman utama juga terdapat fitur *Daily* *Summary* itu sendiri. Fitur tersebut diawali dengan judul halaman "*Daily* *Summary*", kemudian terdapat sebuah *input* *form* tanggal dan tombol "*summary*". Input ini berfungsi sebagai filter tanggal yang akan direkap, dan tombol berfungsi menampilkan hasil rekap di bagian bawah. Saat *form* tanggal sudah terisi dan tombol *summary* ditekan, akan menampilkan absensi karyawan pada tanggal tersebut dalam bentuk tabel. Selain menampilkan absensi karyawan dalam bentuk tabel, juga akan menampilkan sebuah *template* *message* yang nantinya dilaporkan oleh pihak manajemen bertujuan sebagai laporan untuk pihak manajemen. Semua *output* yang sudah dijelaskan dapat diunduh dalam format *file* .*xls*.

* + 1. Halaman Monthly Summary

Halaman *monthly* *summary* merupakan halaman bagian dari *dashboard*. Pada halaman *monthly* *summary* ini terdapat sebuah *navbar* dan *sidebar* yang sama dengan yang ada pada halaman *dashboard*.

Berikut pada gambar 2.32 merupakan tampilan dari halaman *monthly* *summary*.



Gambar 2. 32 tampilan halaman *monthly* *summary*

Pada halaman utama dari *Monthly* *Summary* ini menampilkan *breadcrumb* yang sama dengan halaman *dashboard*. Selain itu, pada halaman utama juga terdapat fitur *Monthly* *Summary* itu sendiri. Fitur tersebut diawali dengan judul halaman "*Monthly* *Summary*", kemudian terdapat sebuah *input* *form* bulan dan tombol "*summary*". Input ini berfungsi sebagai filter tanggal yang akan direkap, dan tombol berfungsi menampilkan hasil rekap di bagian bawah. Saat *form* sudah terisi dan tombol *summary* ditekan, akan menampilkan absensi karyawan pada bulan tersebut dalam bentuk tabel. Semua output yang sudah dijelaskan dapat diunduh dalam format file .xls.

BAB III PENGUJIAN

Pengujian Website Absensi Karyawan

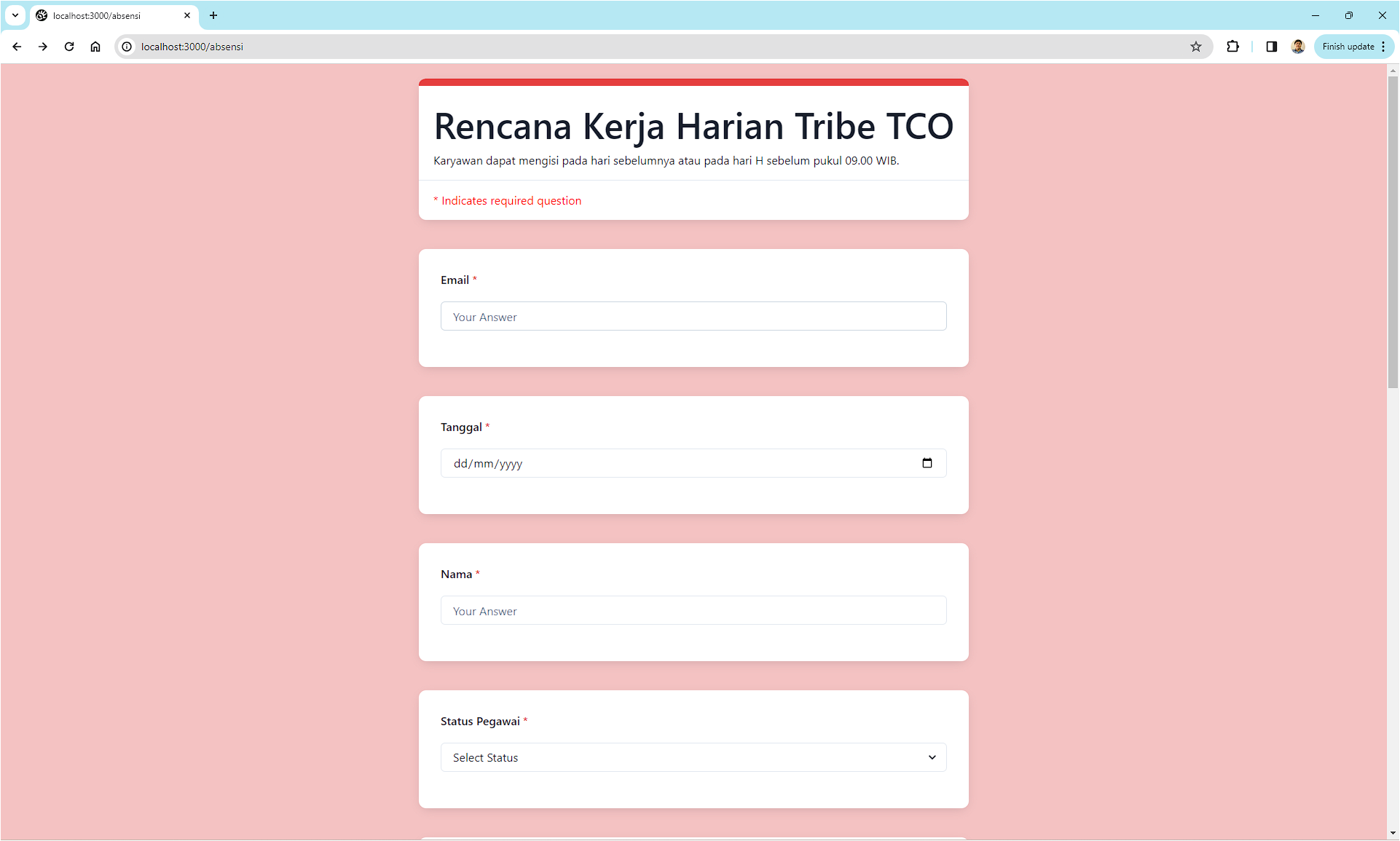
Pengujian sistem dilakukan untuk memastikan sistem berjalan dengan semestinya. Pengujian yang dilakukan merupakan uji coba kualitas sistem dengan menggunakan metode *blackbox* *testing*. *Black box* *testing* merupakan salah satu metode pengujian sistem yang berfokus kepada sisi fungsionalitas khususnya pada input aplikasi apakah sesuai dengan apa yang diharapkan atau tidak. Tahapan pengujian merupakan salah satu tahap yang harus ada pada siklus pengembangan sistem sebelum *website* di publikasi. Pada laporan ini menggunakan metode pengujian *black* *box* karena ingin mengetahui fungsionalitas sistem ini.

Hasil pengujian sistem ini dideskripsikan dengan menganalisis *output* yang dihasilkan oleh sistem berdasarkan *input* yang diberikan pengguna oleh sistem ini dari setiap menu dan fitur yang terdapat dalam sistem absensi karyawan ini. Berikut menu-menu atau fitur-fitur yang diujikan dalam tahap pengujian.

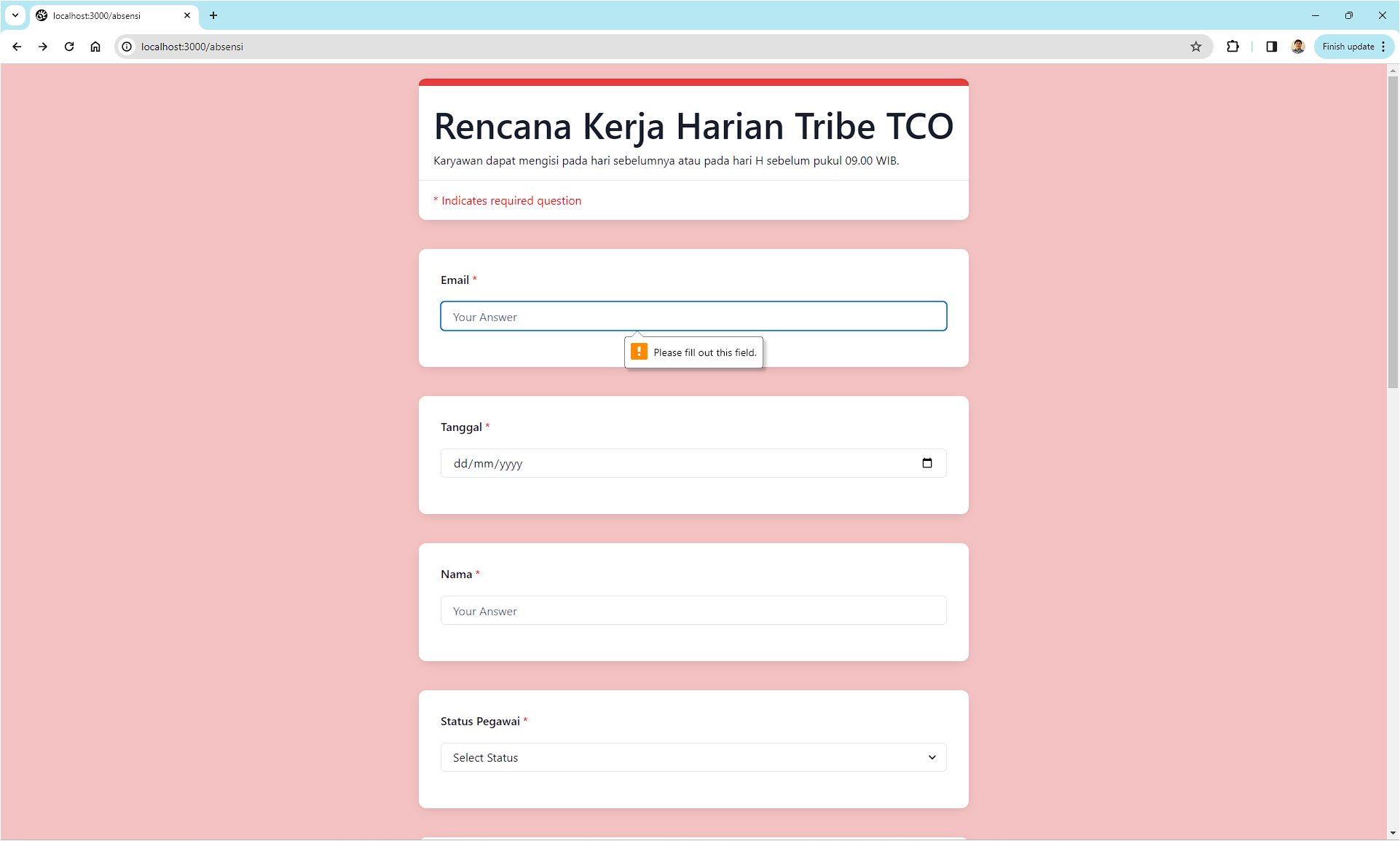
1. Pengujian sistem absensi pada *website* akan ditunjukan pada tabel 3.4.

Tabel 3. 4 Pengujian *black* *box* pada halaman absensi

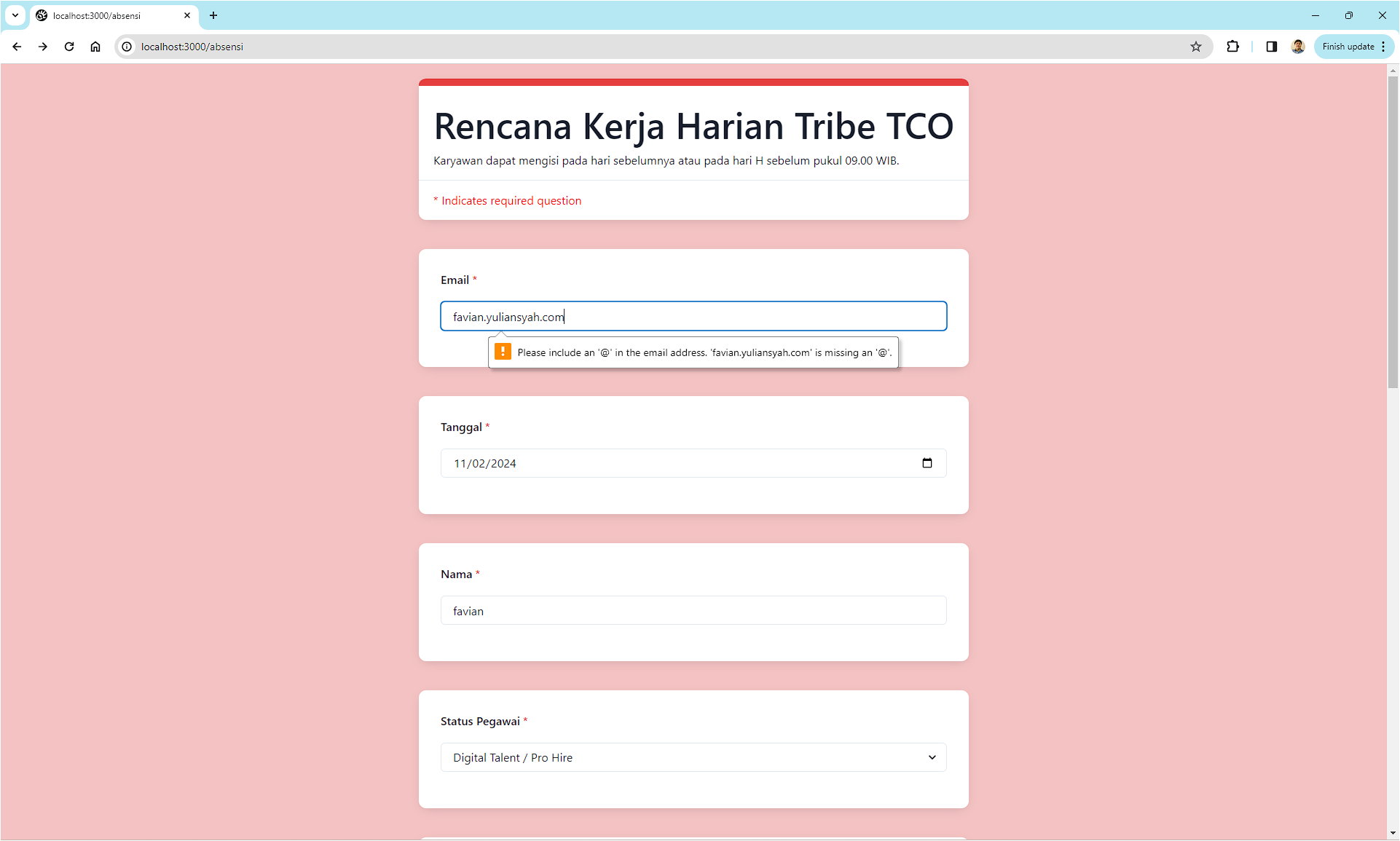
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skenario dan *Test* *Case*** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** | **Keterangan** |
| Mengakses dan melihat halaman *web* dengan mengetikan *URL* *website* absensi | Dapat diakses dan tampil semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Sistem dapat diakses dan menampilkan semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Berhasil | Merujuk pada gambar 3.33 |
| *Form* absensi tidak diisi | Pengguna tetap pada halaman absensi dan muncul *alert* untuk mengisi *form* yang masih kosong. | Muncul *alert* ketika pengguna belum melengkapi semua *form* setelah pengguna menekan tombol *submit* | Berhasil | Merujuk pada gambar 3.34 |
| *Form* *input* *email* pada halaman absensi tidak sesuai | Pengguna tetap pada halaman absensi dan muncul *alert* untuk mengecek ulang *form* *email* yang tidak sesuai | Muncul *alert* ketika semua *form* dilengkapi tetapi *form* *email* tidak mengandung tanda “@” pada *inputnya* | Berhasil | Merujuk pada gambar 3.35 |
| Mengisi semua *form* dan melakukan *submit*, dan tercatat pada *database* | Pengguna berhasil submit lalu dialihkan ke halaman sukses dan pada *database* tercatat semua *form* yang diisi | Pengguna dapat melakukan *submit* lalu dialhikan ke halaman sukses dan *submit* berhasil terekam di *database* | Berhasil | Merujuk pada gambar 3.36 dan gambar 3.37 |



Gambar 3. 33 Pengujian halaman absensi 1



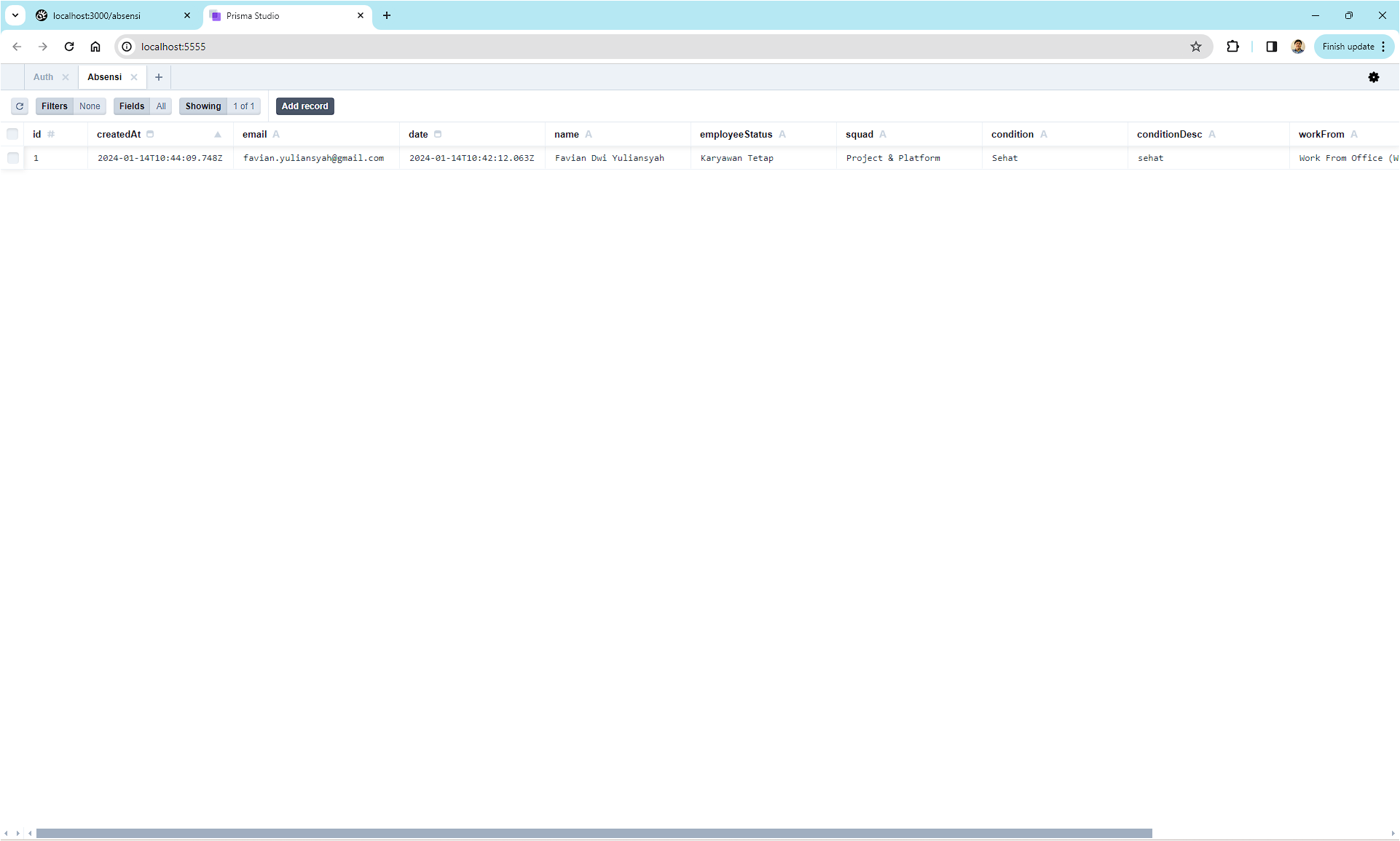
Gambar 3. 34 Pengujian halaman absensi 2



Gambar 3. 35 Pengujian halaman absensi 3



Gambar 3. 36 Pengujian halaman absensi 4

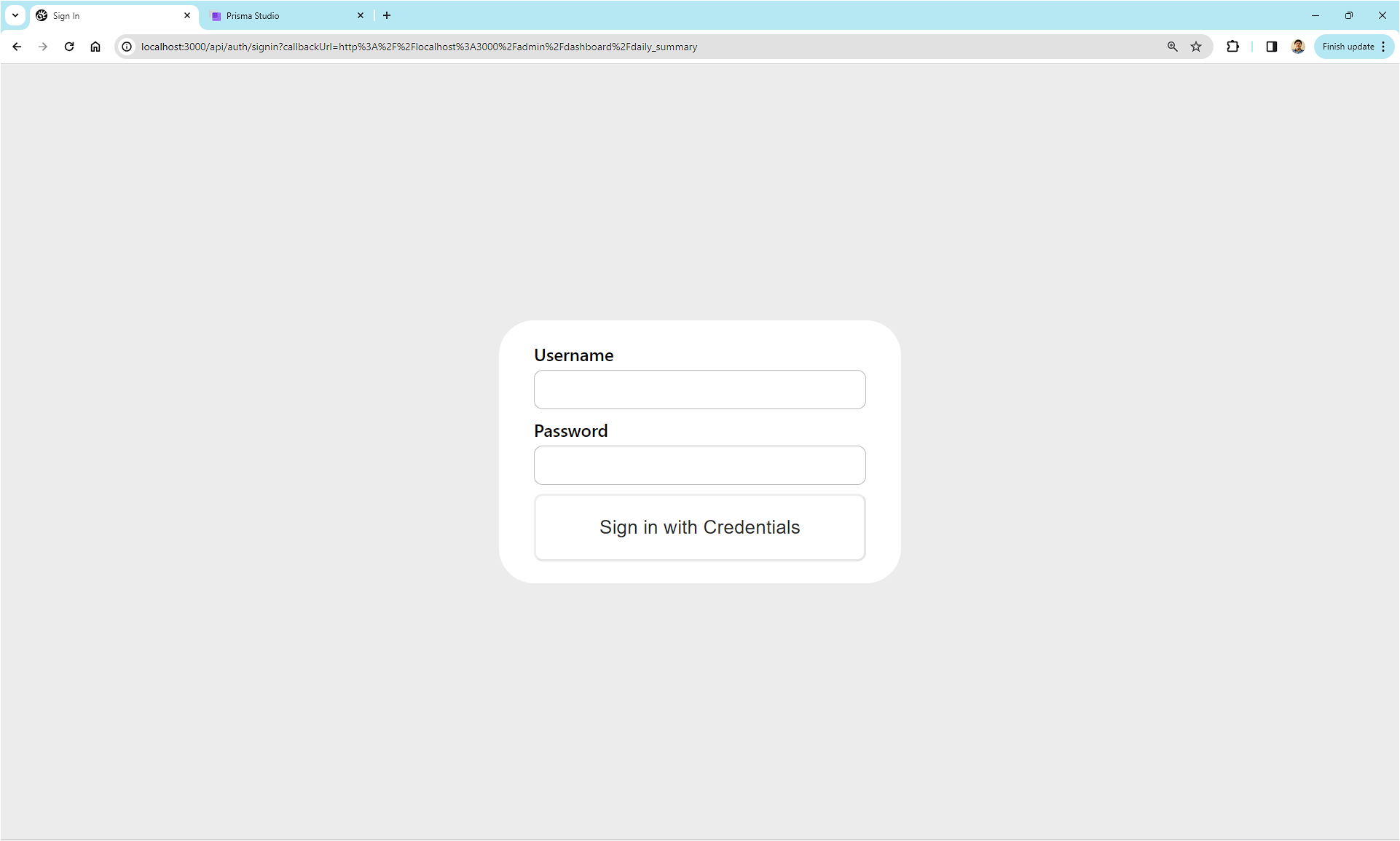


Gambar 3. 37 Pengujian halaman absensi 5

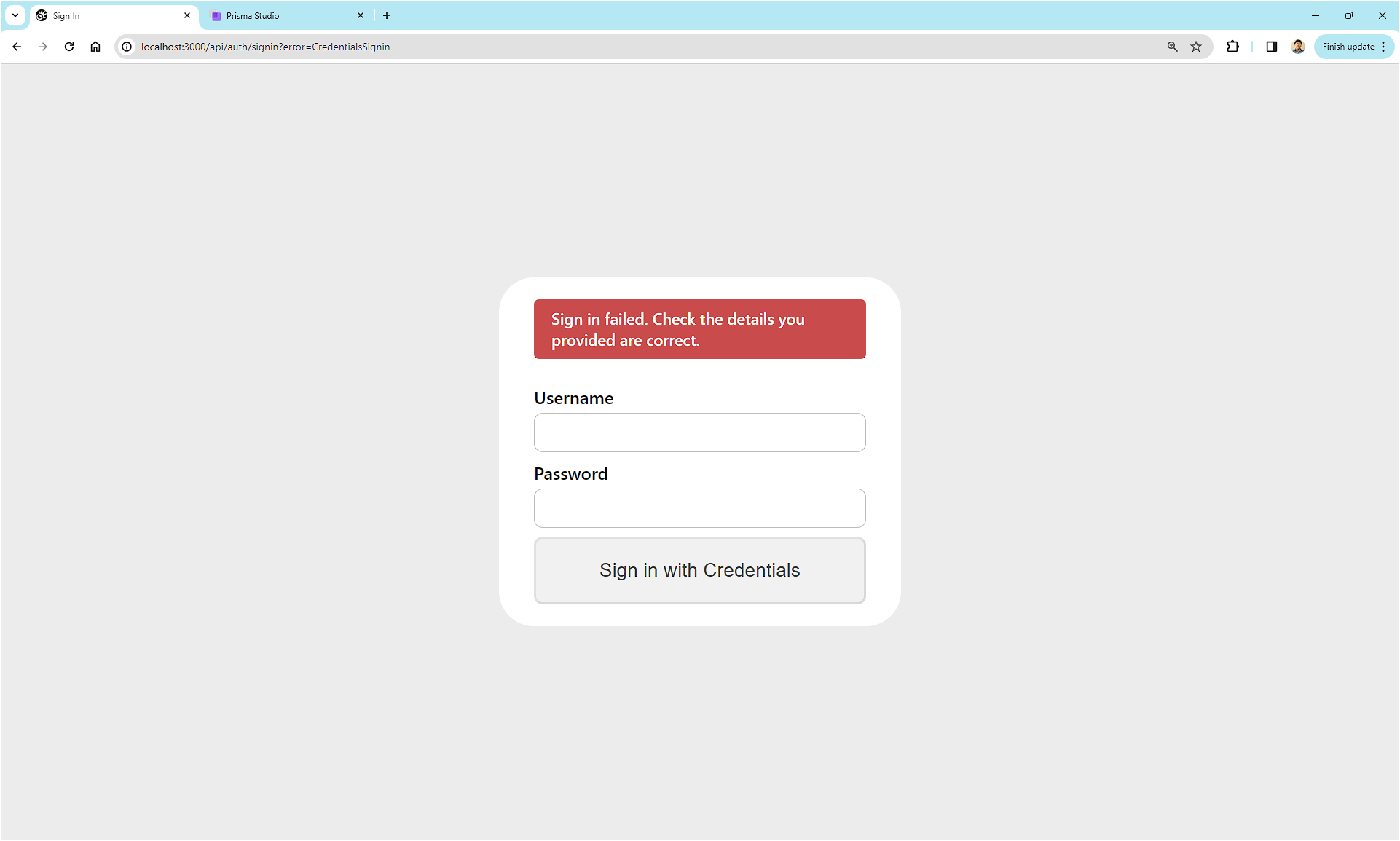
1. Pengujian sistem *login* pada *website* akan ditunjukan pada tabel 3.5.

Tabel 3. 5 Pengujian *black* *box* pada *login*

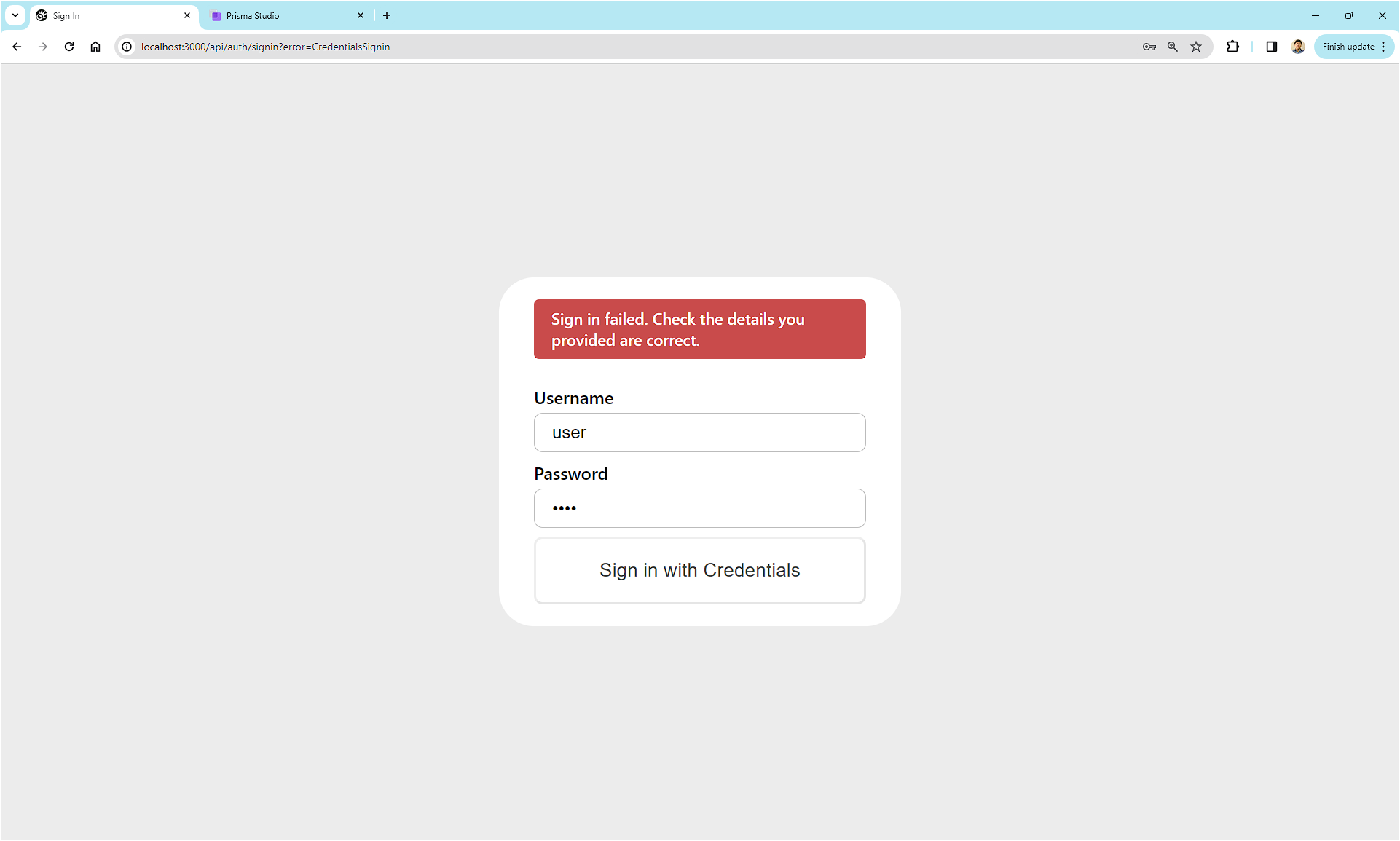
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skenario dan *Test* *Case*** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** | **Keterangan** |
| Mengakses dan melihat halaman *web* dengan mengetikan *URL* *website* absensi | Dapat diakses dan tampil semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Sistem dapat diakses dan menampilkan semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Berhasil | Merujuk pada gambar 3.38 |
| *Input username* dan *password* tidak diisi | Pengguna tetap pada halaman login dan muncul *alert* untuk mengisi *form* kosong | Pengguna tetap pada halaman *login* dan muncul *alert* untuk mengisi *form* kosong | Berhasil | Merujuk pada gambar 3.39 |
| *Input username* dan *password* diisi dengan data yang salah atau tidak sesuai | Pengguna tetap pada halaman *login* dan muncul *alert* bahwa *username* dan *password* yang *diinputkan* salah. | Pengguna tetap pada halaman *login* dan muncul *alert* bahwa *username* dan *password* yang diinputkan salah | Berhasil | Merujuk pada gambar 3.40 |
| Input *username* dan *password* diisi dengan data yang benar | Pengguna dialihkan ke halaman *dashboard* | Pengguna dialihkan ke halaman *dashboard* | Berhasil | Merujuk pada gambar 3.41 |



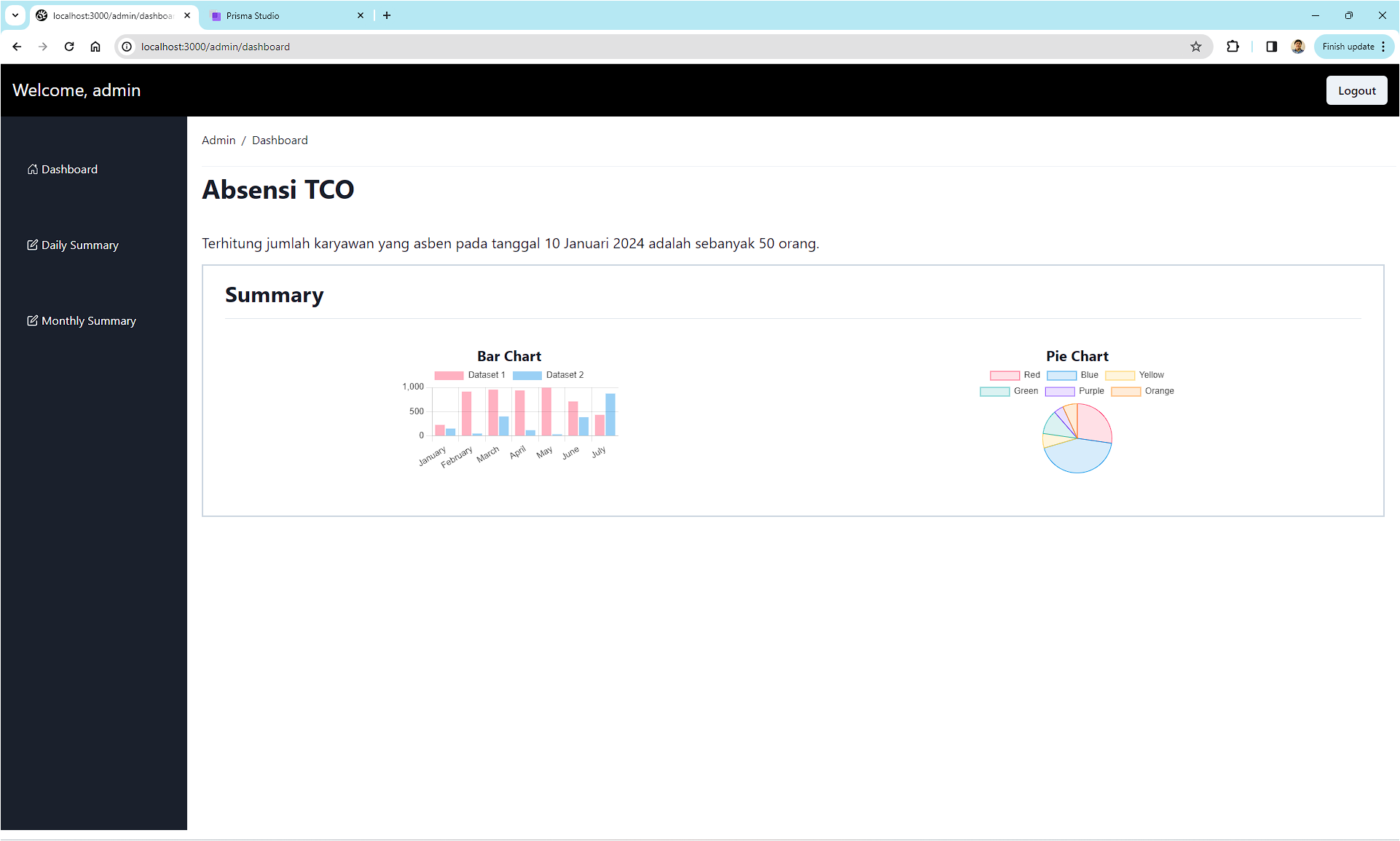
Gambar 3. 38 Pengujian halaman *login* 1



Gambar 3. 39 Pengujian halaman *login* 2



Gambar 3. 40 Pengujian halaman *login* 3

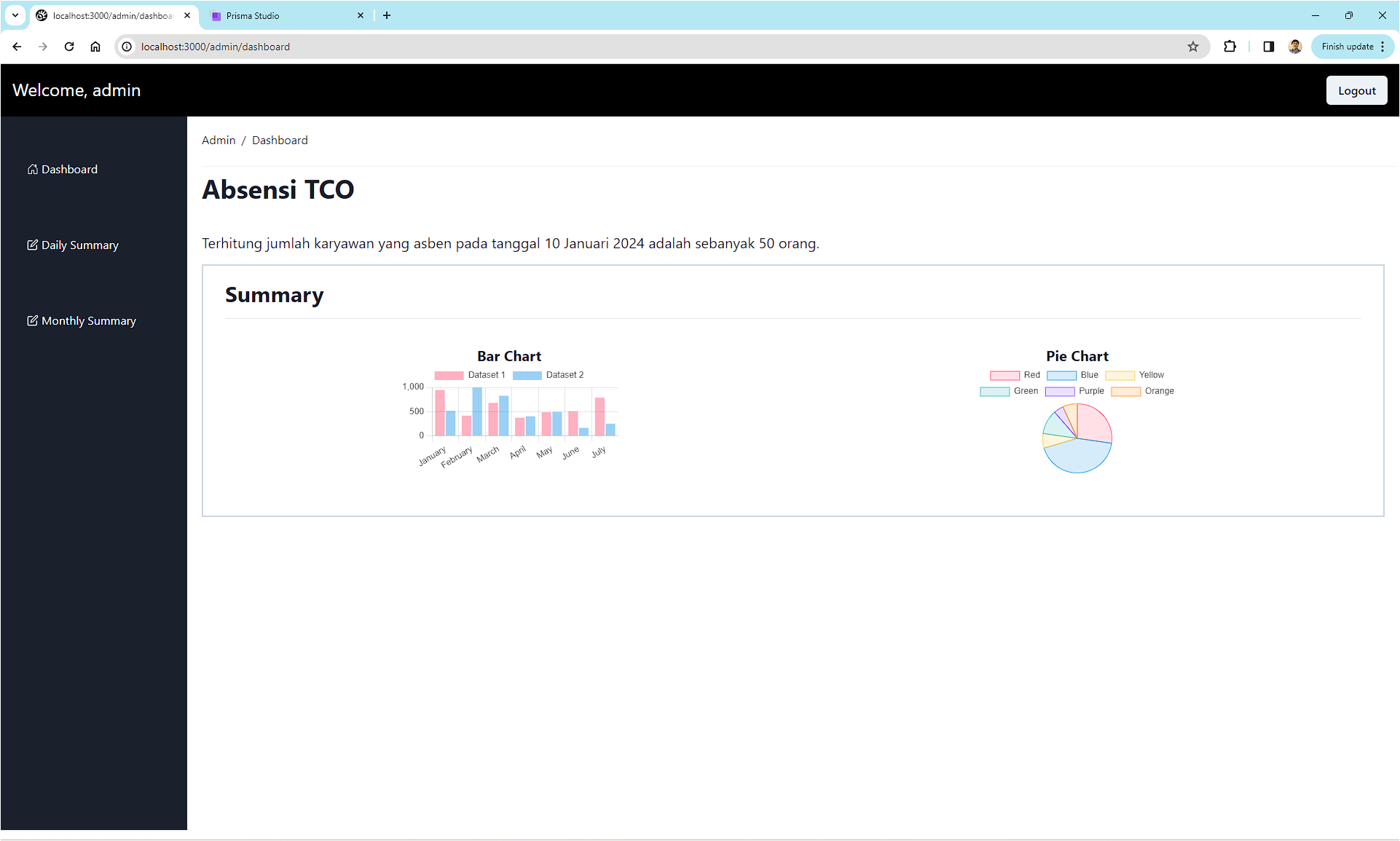


Gambar 3. 41 Pengujian halaman *login* 4

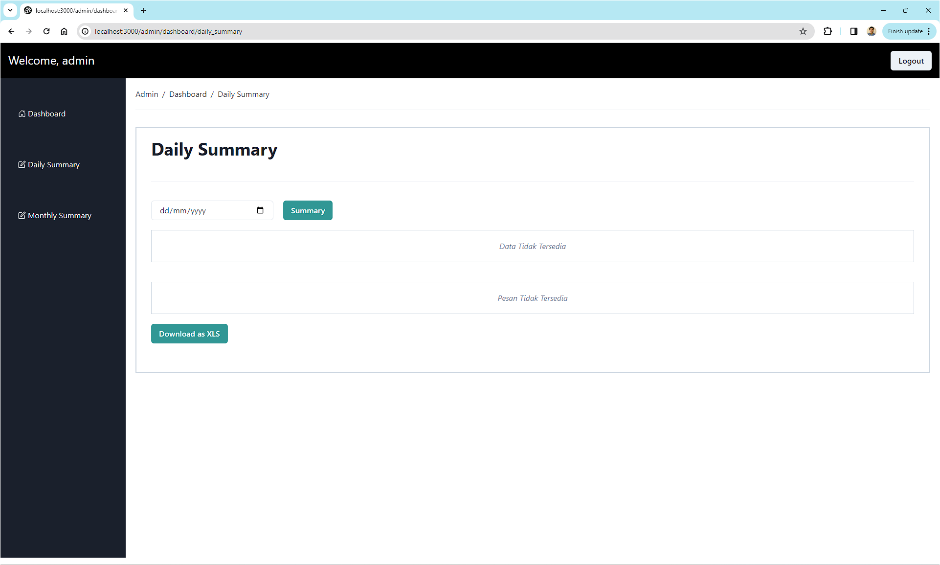
1. Pengujian sistem pada halaman *dashboard* akan ditunjukan pada tabel 3.6.

Tabel 3. 6 Pengujian *black* *box* sistem pada halaman *dashboard*

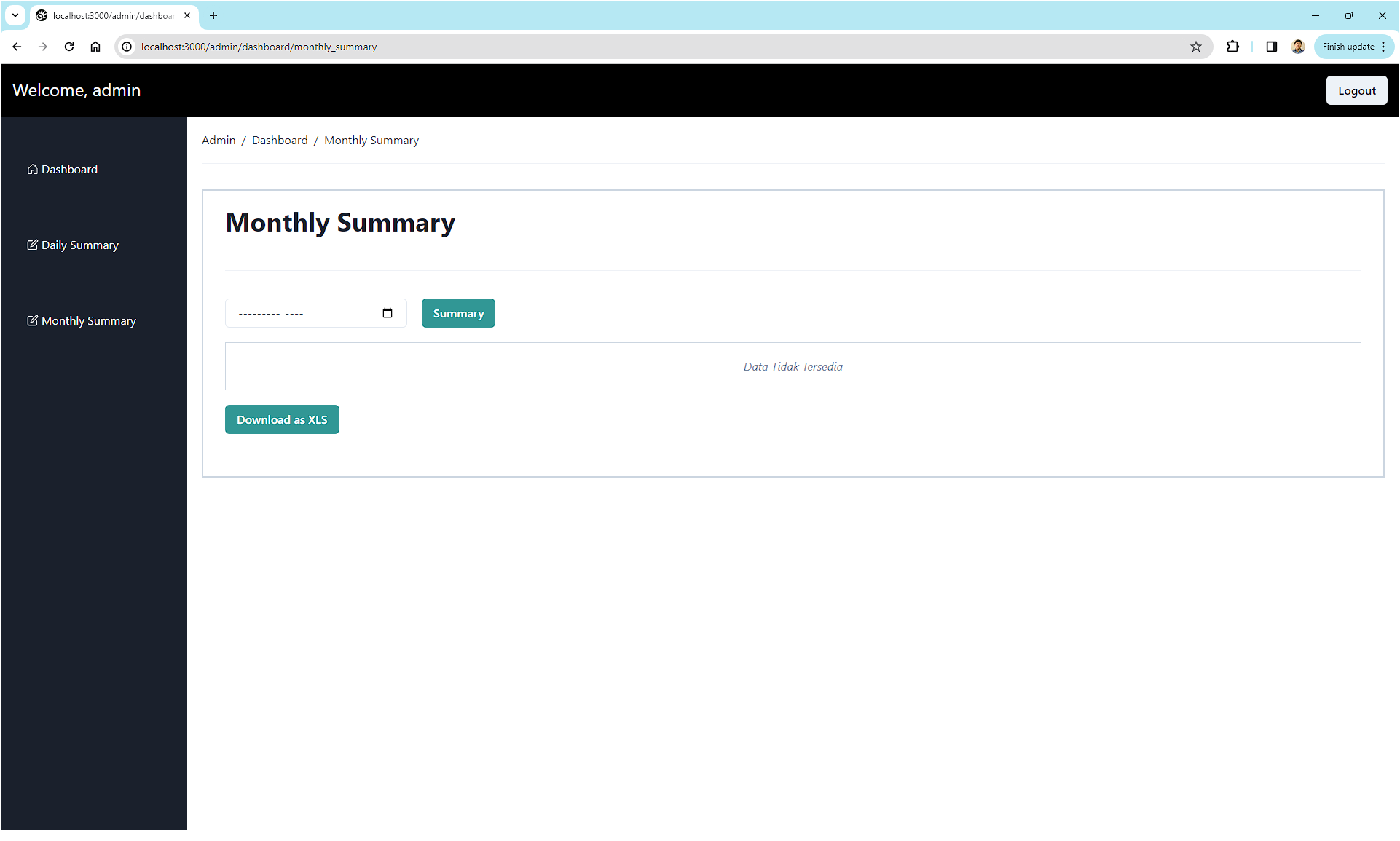
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skenario dan *Test* *Case*** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** | **Keterangan** |
| Mengakses halaman *dashboard* dalam keadaan belum *login* | Pengguna dialihkan ke halaman *login* | Pengguna dialihkan ke halaman *login* | Berhasil |  |
| Mengakses dan melihat halaman *web* dengan mengetikan *URL* halaman *dashboard* | Dapat diakses dan tampil semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Sistem dapat diakses dan menampilkan semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Berhasil | Merujuk pada gambar 3.42 |
| Menekan tombol logout pada *navbar* untuk mengeluarkan akun admin | Pengguna dialihkan ke halaman *login* dan akun *admin* berhasil dikeluarkan | Pengguna dialihkan ke halaman *login* dan akun *admin* berhasil dikeluarkan | Berhasil |  |
| Menekan *link* *dialy* *summary* pada *sidebar* | Pengguna dialihkan ke halaman *daily* *summary* | Pengguna dialihkan ke halaman *daily* *summary* | Berhasil | Merujuk pada gambar 3.43 |
| Menekan *link* *monthly* *summary* pada *sidebar* | Pengguna dialihkan ke halaman *monthly* *summary* | Pengguna dialihkan ke halaman *monthly* *summary* | Berhasil | Merujuk pada gambar 3.44 |



Gambar 3. 42 Pengujian halaman *dashboard* 1



Gambar 3. 43 Pengujian halaman *dashboard* 2

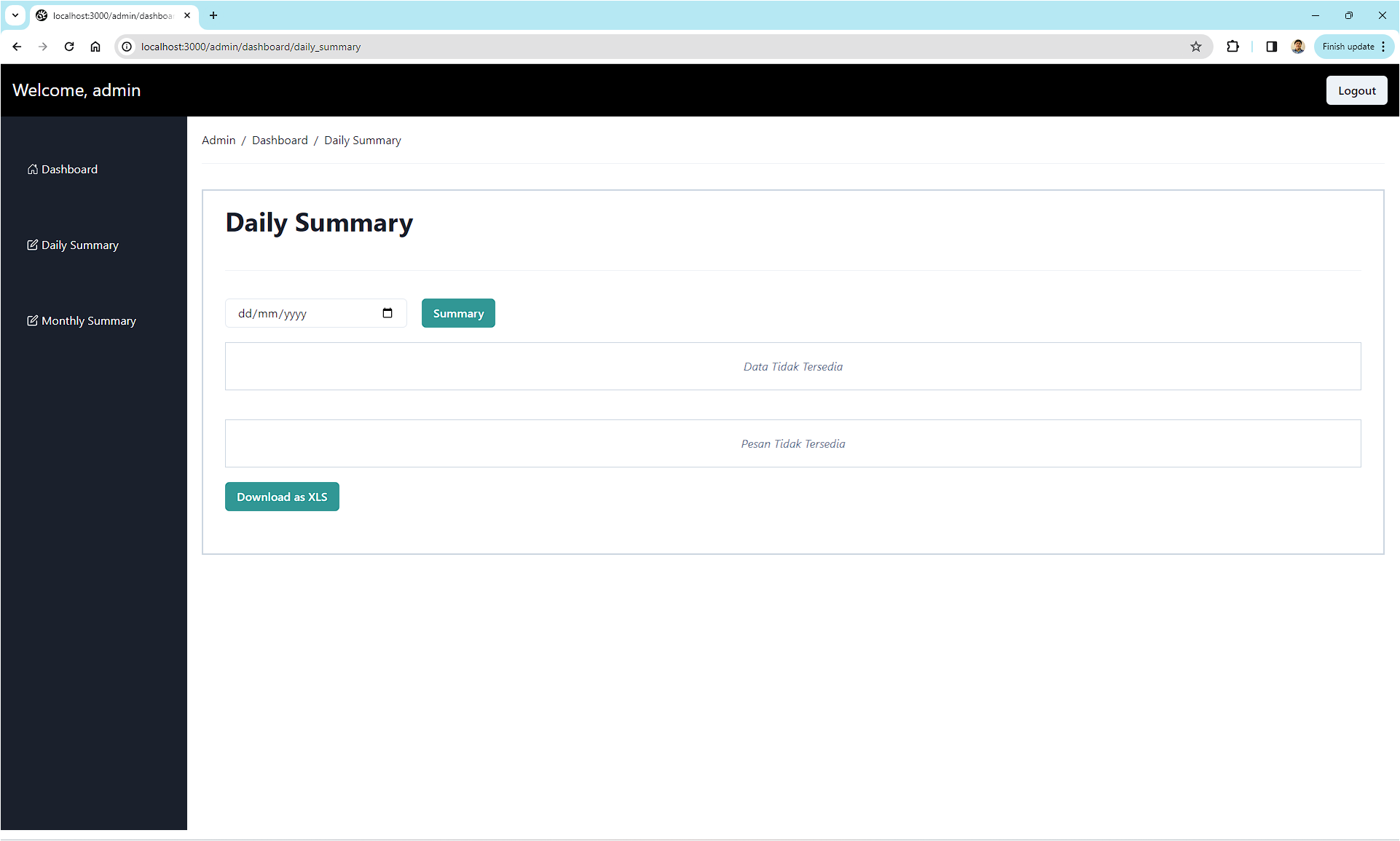


Gambar 3. 44 Pengujian halaman *dashboard* 3

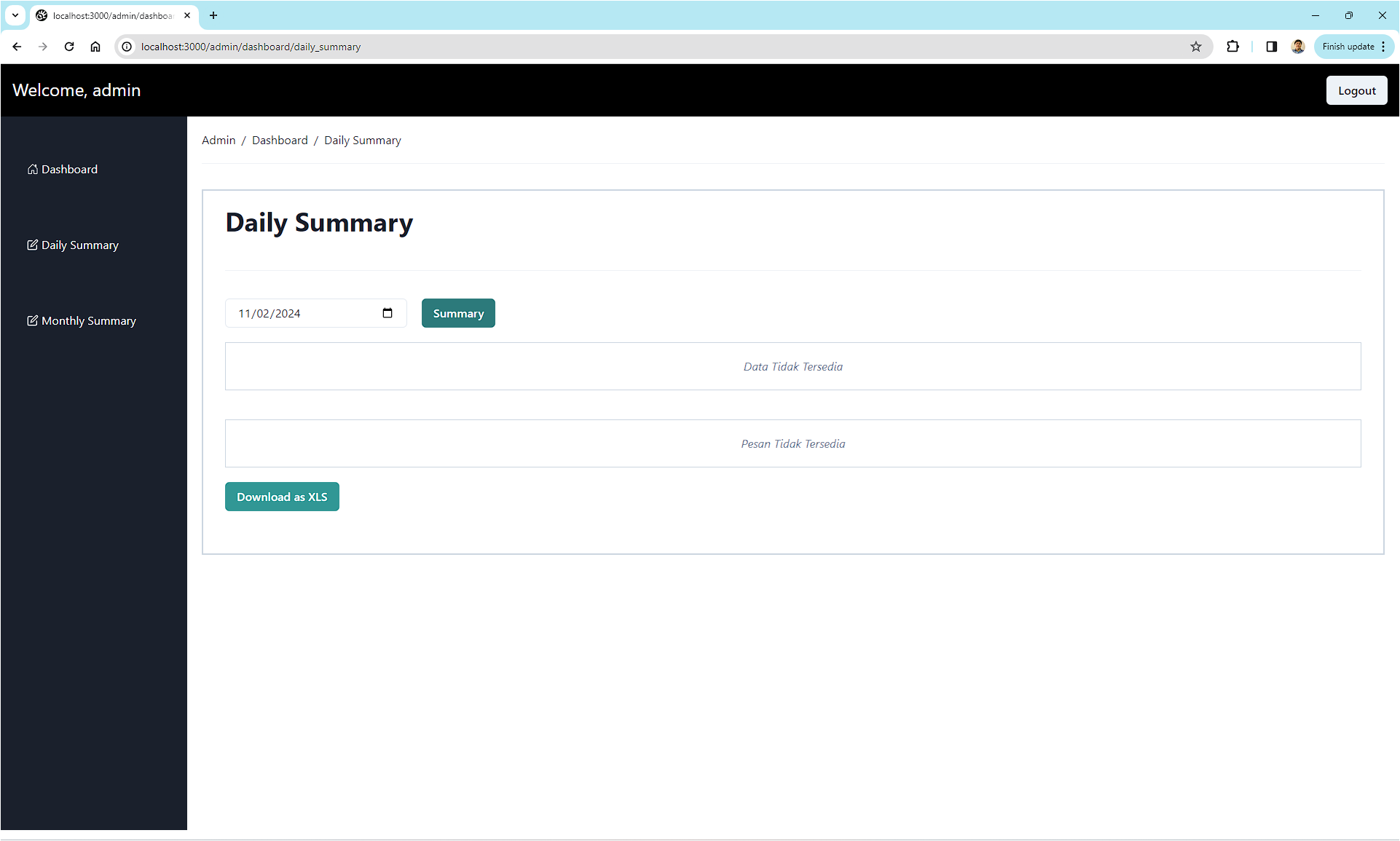
1. Pengujian fitur *daily* *summary* pada *website* akan ditunjukan pada tabel 3.7.

Tabel 3. 7 Pengujian *black* *box* pada fitur *daily* *summary*

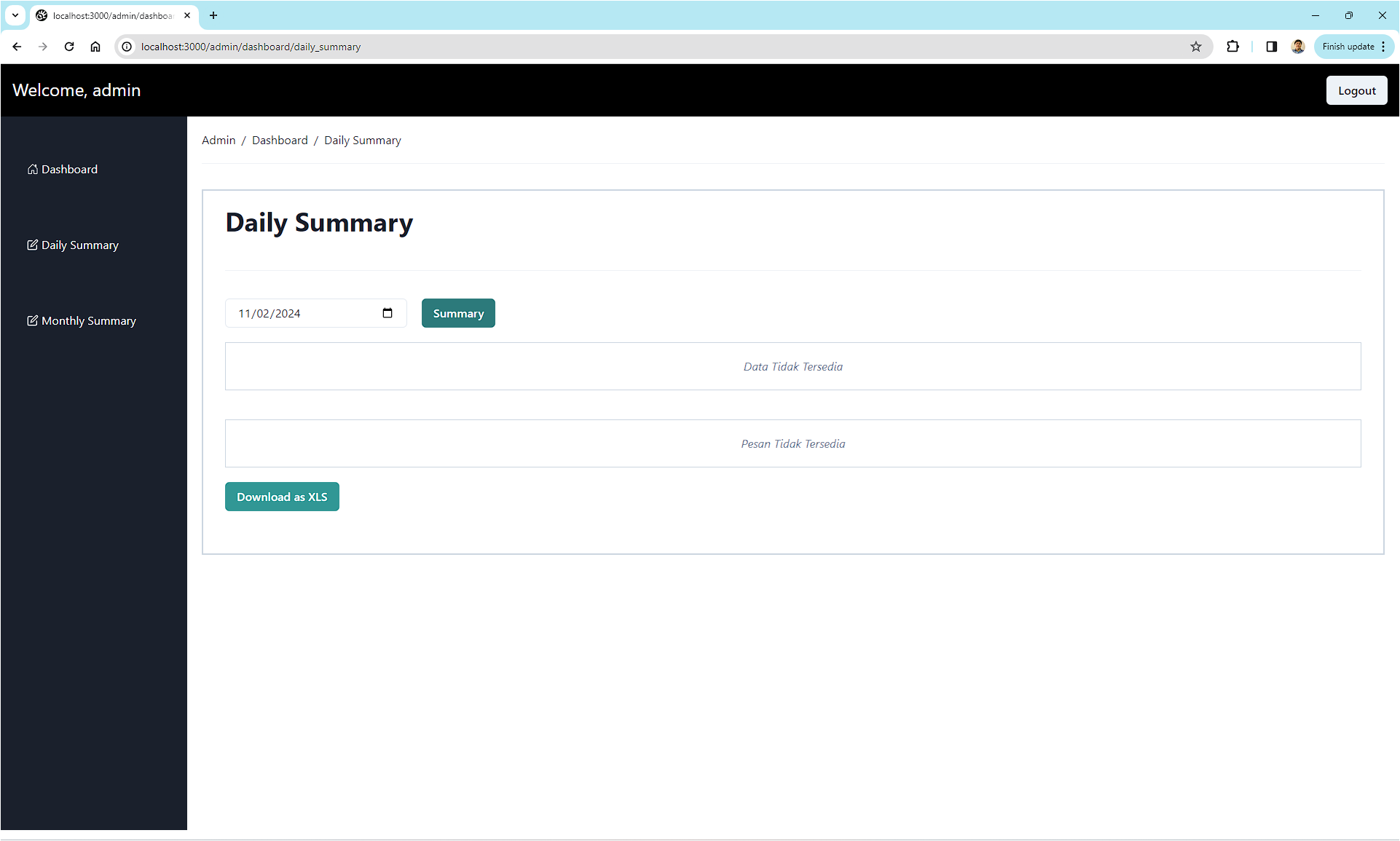
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skenario dan *Test* *Case*** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** | **Keterangan** |
| Mengakses halaman *daily* *summary* dalam keadaan belum *login* | Pengguna dialihkan ke halaman *login* | Pengguna dialihkan ke halaman *login* | Berhasil |  |
| Mengakses dan melihat halaman *web* dengan mengetikan *URL* halaman | Dapat diakses dan tampil semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Sistem dapat diakses dan menampilkan semua komponen seperti *header*, *navbar*, *content*, | Berhasil | Merujuk pada gambar 3.45 |
| *daily summary* |  | dan *footer* |  |  |
| Mengisi *form* tanggal dan menekan tombol *summary* | Pengguna dapat mengisi tanggal pada *form* dan menekan tombol *summary* | Pengguna dapat mengisi tanggal pada *form* dan menekan tombol *summary* | Berhasil |  |
| Menampilkan hasil *summary* berupa data tabel dan *template* pesan | Sistem dapat menampilkan hasil summary berupa data tabel dan data *template* pesan | Sistem dapat menampilkan hasil *summary* berupa data tabel dan data *template* pesan | Berhasil | Merujuk pada gambar 3.46 |
| Menekan tombol *Download* *as* *XLS* dan dapat mengunduh *file* .*xls* | Sistem dapat mengeksekusi perintah pada tombol D*ownload as XLS* dan pengguna dapat mengunduh *file* .*xls* | Sistem dapat mengeksekusi perintah pada tombol D*ownload as XLS* dan pengguna dapat mengunduh *file* .*xls* | Berhasil | Merujuk pada gambar 3.47 |



Gambar 3. 45 Pengujian halaman *daily* *summary* 1



Gambar 3. 46 Pengujian halaman *daily* *summary* 2

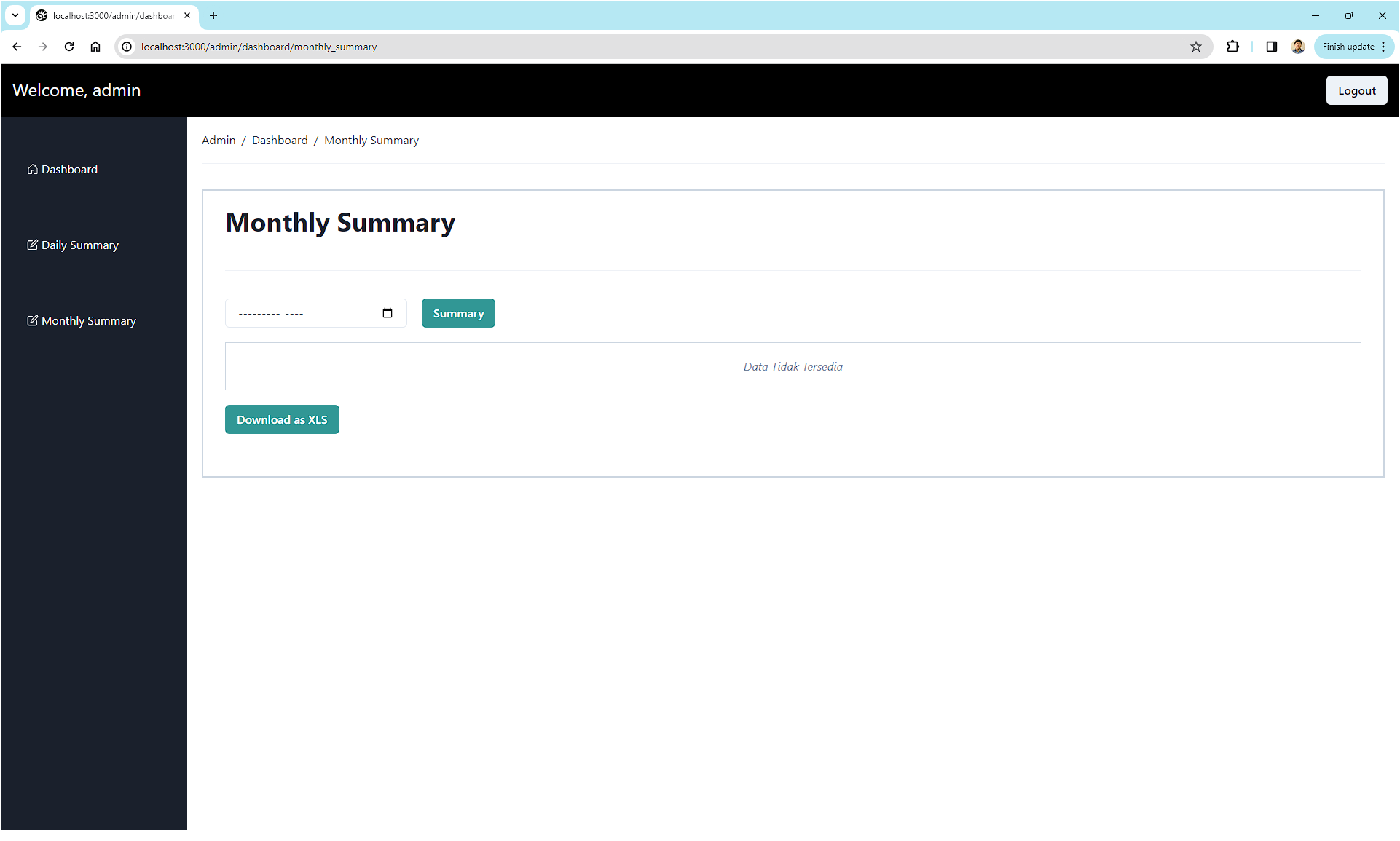


Gambar 3. 47 Pengujian halaman *daily* *summary* 3

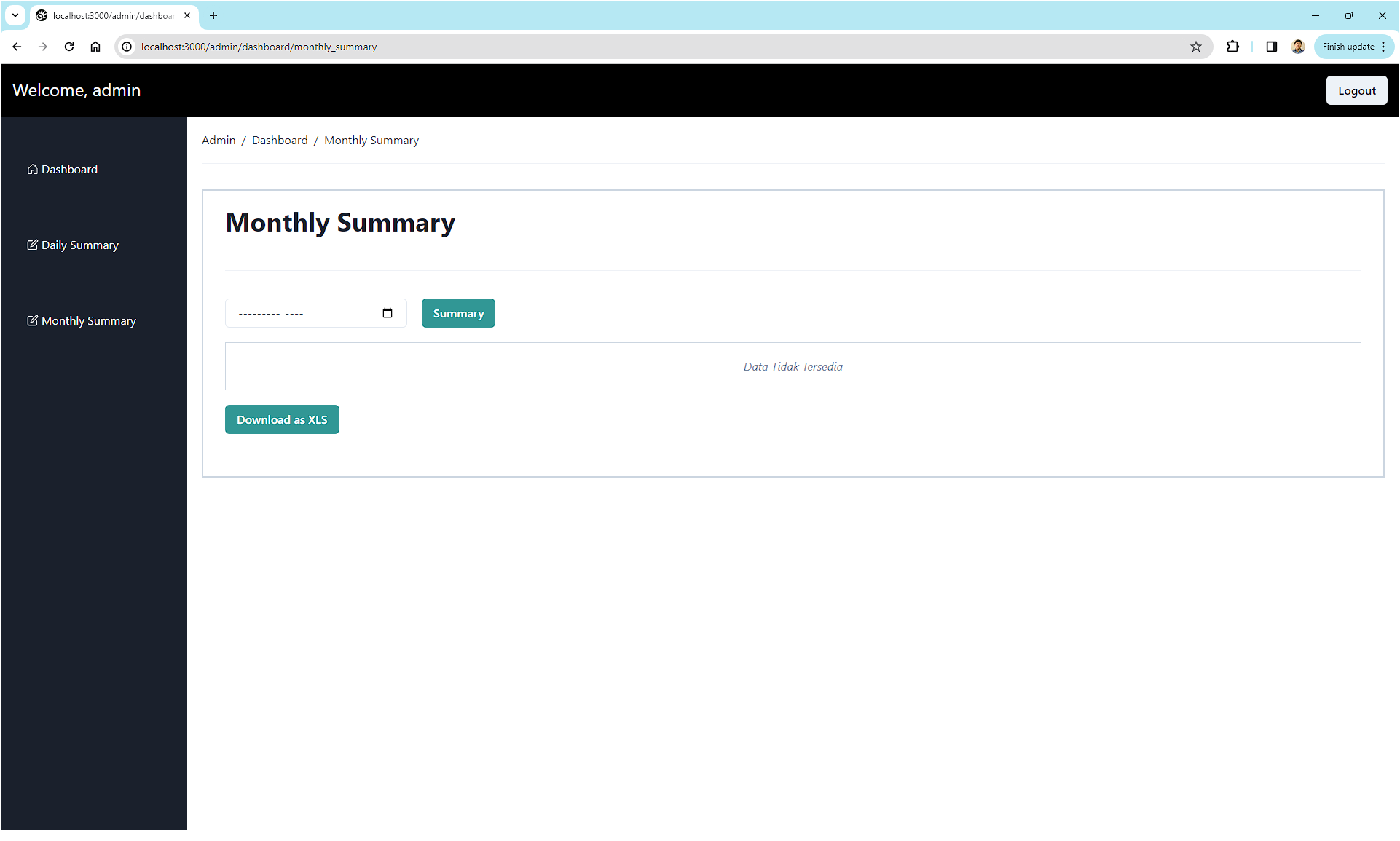
1. Pengujian *fitur* *monthly* *summary* pada *website* ditunjukan pada tabel 3.8.

Tabel 3. 8 Pengujian *black* *box* pada *monthly* *summary*

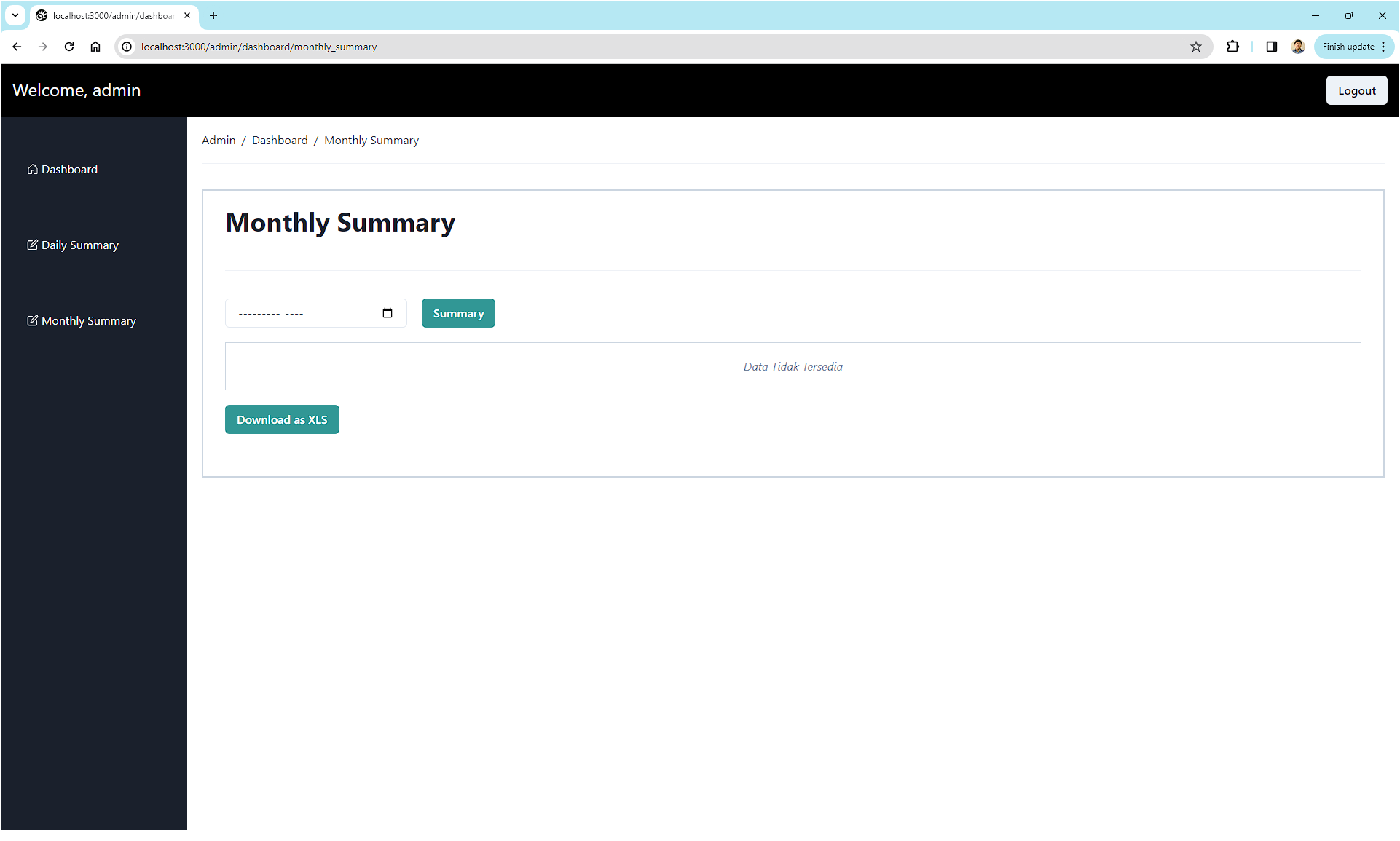
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skenario dan *Test* *Case*** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** | **Keterangan** |
| Mengakses halaman *monthly* *summary* dalam keadaan belum *login* | Pengguna dialihkan ke halaman *login* | Pengguna dialihkan ke halaman *login* | Berhasil |  |
| Mengakses dan melihat halaman *web* dengan mengetikan *URL* halaman *monthly summary* | Dapat diakses dan tampil semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Sistem dapat diakses dan menampilkan semua komponen seperti *header*, *navbar*, *content*, dan *footer* | Berhasil | Merujuk pada gambar 3.47 |
| Mengisi *form* tanggal dan menekan tombol *summary* | Pengguna dapat mengisi tanggal pada *form* dan menekan tombol *summary* | Pengguna dapat mengisi tanggal pada *form* dan menekan tombol *summary* | Berhasil |  |
| Menampilkan hasil *summary* berupa data tabel | Sistem dapat menampilkan hasil summary berupa data tabel | Sistem dapat menampilkan hasil *summary* berupa data tabel | Berhasil | Merujuk pada gambar 3.48 |
| **Skenario dan *Test* *Case*** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** | **Keterangan** |
| Menekan tombol *Download* *as* *XLS* dan dapat mengunduh *file* .*xls* | Sistem dapat mengeksekusi perintah pada tombol D*ownload as XLS* dan pengguna dapat mengunduh *file* .*xls* | Sistem dapat mengeksekusi perintah pada tombol D*ownload as XLS* dan pengguna dapat mengunduh *file* .*xls* | Berhasil | Merujuk pada gambar 3.49 |



Gambar 3. 48 Pengujian halaman *monthly* *summary* 1



Gambar 3. 49 Pengujian halaman *monthly* *summary* 2



Gambar 3. 50 Pengujian halaman monthly summary 3

BAB IV PENUTUP

* 1. Kesimpulan

Berdasarkan uraian dari bab-bab sebelumnya yang membahas mengenai *website* absensi ini, serta setelah penulis menguji sistem tersebut dapat disimpulkan bahwa:

1. Sistem absensi berbasis *website* menggunakan *framework* *javascript*, *next*.*js* juga menggunakan *prisma* *ORM* dan *PostgreSQL* sebagai *database* telah berhasil dibuat dan berjalan dengan baik dan stabil.
2. Sistem absensi berbasis website yang terintegrasi dengan database sehingga dapat dikelola dimana saja dan kapan saja, selain itu juga dapat mempercepat proses rekap data absensi.
3. Dengan adanya sistem absensi karyawan ini sudah dapat membantu pihak manajemen untuk merekap absensi karyawan dengan cara yang lebih efisien dan telah berhasil mengurangi adanya kesalahan dalam rekapitulasi data.
   1. Saran

Pengimplementasian sistem absensi berbasis *website* ini masih dapat dikembangkan lagi oleh pihak manapun yang ingin melakukan pengembangan lebih lanjut mengenai sistem ini, sehingga kedepannya proyek ini menjadi lebih baik karena adanya tambahan beberapa fitur dengan mempertimbangkan beberapa hal yang disarankan untuk diaplikasikan, diantaranya yaitu:

1. Akun *admin* pada sistem absensi berbasis website ini dapat diintegrasikan pada *database* untuk nantinya di kembangkan data akun *admin* yang lebih dinamis, yaitu *admin* bisa merubah *username* ataupun *password* kedepannya jika diperlukan perubahan.
2. Selain diintegrasikan pada *database* akun *admin* pada sistem absensi berbasis *website* ini dapat diintegrasikan pada *gmail*, ataupun nomor telepon (*whatsapp*), dan lain-lain untuk kebutuhan, seperti *2-Step Verification* pada proses *login* ataupun pada proses manajemen akun *admin*.
3. Rekapitulasi data absensi pada sistem absensi berbasis *website* ini dapat dikembangkan dengan menambah sebuah diagram data dengan tujuan agar pihak manajemen atau *admin* dapat dengan mudah membaca hasil rekapitulasi.
4. Adanya pengembangan pada sistem absensi berbasis *website* ini seperti sistem *login* untuk karyawan dengan tujuan agar tidak sembarang orang dapat mengakses *website* ini, dengan begitu pihak manajemen karyawan bisa mendapat data yang lebih akurat.
5. Adanya pengamanan lebih dari sistem aplikasi berbasis *website* ini seperti *captcha*, dikarenakan berisi informasi dan data penting baik itu data absensi karyawan ataupun data akun *admin*. Karena seperti yang diketahui bahwa aplikasi berbasis *website* memiliki peluang kerentanan yang lebih besar sehingga sangat berpotensi adanya kebocoran data kepada pihak yang tidak bertanggung jawab.



|  |
| --- |
| [Company name] |
| Laporan Akhir |
| Sistem Informasi Jaringan dan Aplikasi |

|  |
| --- |
| Favian Dwi Yuliansyah  [Date] |