**PERANCANGAN APLIKASI ABSENSI FINGERPRINT MENGGUNAKAN ALGORITMA MINUTIAE UNTUK MENINGKATKAN AKURASI DAN EFISIENSI ABSENSI KARYAWAN PADA PT. VISI KARYA PRAKARSA**

**SKRIPSI**

**Karya Tulis sebagai syarat untuk memperoleh**

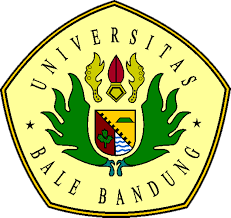
**Gelar Sarjana Komputer dari Fakultas Teknologi Informasi**

**Universitas Bale Bandung**

Disusun Oleh :

TIANA RIZKI ANANDA

NPM. 301190003

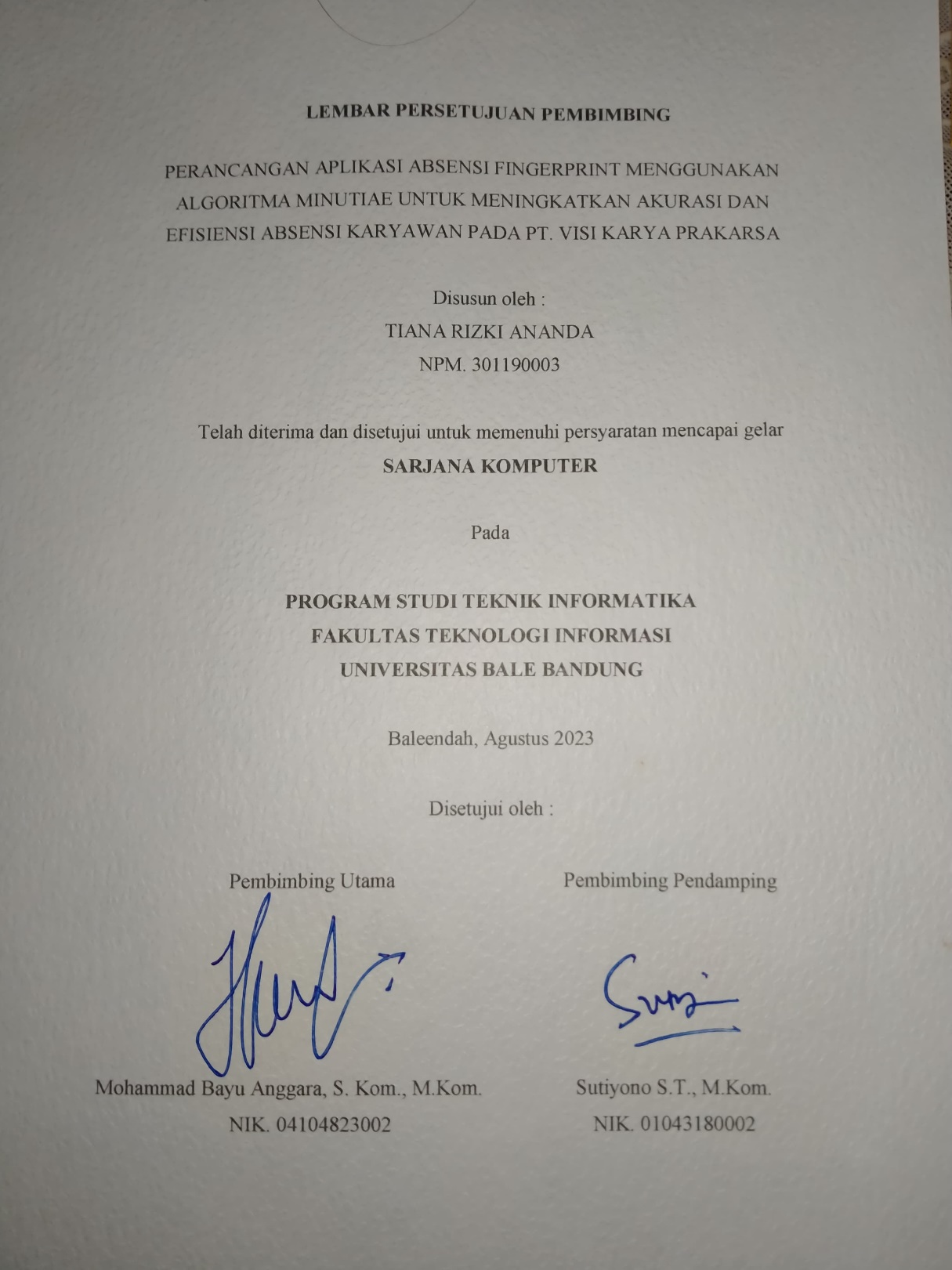


PROGRAM STRATA 1

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG

BANDUNG

2023



# 



**HALAMAN PERNYATAAN**

Yang bertanda tangan dibawah ini, saya:

|  |  |
| --- | --- |
| Nama | : Tiana Rizki Ananda |
| NPM | : 301190003 |
| Jurusan | : Teknik Informatika |
| Fakultas | : Fakultas Teknologi Informasi |
| Judul | :Perancangan Aplikasi Absensi Fingerprint Menggunakan Algoritma Minutiae Untuk Meningkatkan Akurasi Dan Efisiensi Absensi Karyawan Pada Pt. Visi Karya Prakarsa |

Dengan ini penulis menyatakan sebenar-benarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran, dan pemaparan asli dari penulis sendiri, baik untuk naskah laporan maupun kegiatan programming yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, penulis mencantumkan sumber yang jelas mengikuti tata penulisan karya ilmiah yang seharusnya.

Bandung, Agustus 2023

Tiana Rizki Ananda

NPM 301190003

# ABSTRAK

Sistem absensi karyawan yang efisien dan akurat memiliki peranan penting dalam pengelolaan sumber daya manusia dalam suatu organisasi untuk mengatur atau memantau kinerja karyawan pada suatu perusahaan. Pada PT. Visi Karya Prakarsa memiliki beberapa karyawan disetiap divisi yang dimana diperlukan suatu pemantauan kinerja kerja pada setiap karyawan. Namun nyatanya pada perusahaan ini memiliki permasalahan yaitu kehilangan data absensi adapun di PT. Visi Karya Prakarsa sering terjadi kecurangan absensi yang disebabkan oleh kekurangan pengawasan terhadap setiap karyawan. Dengan menggunakan absensi *fingerprint* untuk absensi karyawan dapat menjadi solusi yang lebih baik. Aplikasi ini digunakan untuk mengelola absensi karyawan dan dibangun agar dapat bekerja secara dinamis yang diharapkan dapat digunakan untuk jangka panjang pada perusahaan. Oleh karena itu untuk mengoptimalkan sistem absensi yang lebih efisien dan akurat di PT. Visi Karya Prakarsa, dirancang suatu sistem absensi *Fingerprint* yang dibantu suatu alat dengan *web* *server* untuk mengelola *report* absensi, agar dapat mengoptimalkan absensi *fingerprint* dan meningkatkan efisiensi absensi karyawan. Pada penelitian ini bertujuan merancang aplikasi absensi karyawan yang langsung mencetak hasil laporan absensi bulanan. Selain itu aplikasi ini dirancang untuk mengurangi kecurangan dalam penitipan absensi antar karyawan dengan menggunakan sistem yang terkomputerisasi, dan mengimplementasikan aplikasi absensi pada perusahaan tersebut. Sistem absensi ini menggunakan SDLC *waterfall* dan menggunakan algoritma *Minutiae* dan metode *Minutiae* *Crossing* *Number* untuk pencocokan sidik jarinya. Hasil penelitian ini memberikan saran mengenai absensi dengan memanfaatkan teknologi komputer kepada PT. Visi Karya Prakarsa yaitu sebagai bahan pertimbangan dalam melakukan perbaikan sistem absensi karyawan.

Kata Kunci : Absensi Karyawan,Algoritma *Minutiae, Fingerprint, Waterfall*

# *ABSTRACT*

*An efficient and accurate employee attendance system has an important role in managing human resources in an organization to manage or monitor employee performance in a company. At PT. Visi Karya Prakarsa has several employees in each division where it is necessary to monitor the work performance of each employee. But in fact, this company has a problem, namely losing attendance data, while at PT. Visi Karya Prakarsa there is often attendance fraud caused by lack of supervision of each employee. Using fingerprint attendance for employee attendance can be a better solution. This application is used to manage employee attendance and is built to work dynamically which is expected to be used for the long term at the company. Therefore, to optimize a more efficient and accurate attendance system at PT Visi Karya Prakarsa, a Fingerprint attendance system is designed which is assisted by a tool with a web server to manage attendance reports, in order to optimize fingerprint attendance and increase employee attendance efficiency. This research aims to design an employee attendance application that directly prints the results of the monthly attendance report. In addition, this application is designed to reduce fraud in attendance between employees by using a computerized system, and implementing an attendance application at the company. This attendance system uses SDLC waterfall and uses the Minutiae algorithm and Minutiae Crossing Number method for fingerprint matching. The results of this study provide suggestions regarding attendance by utilizing computer technology to PT. Visi Karya Prakarsa, namely as a consideration in making improvements to the employee attendance system.*

*Keywords: Employee Attendance, Minutiae Algorithm, Fingerprint, Waterfall*

# KATA PENGANTAR

Puji dan Syukur kami panjatkan ke Hadirat Allah SWT, karena berkat limpahan Rahmat dan Karunia-Nya sehingga penulis dapat menyelesaikan laporan skripsi ini dengan baik dan tepat pada waktunya. Skripsi ini merupakan salah satu syarat untuk menyelesaikan program studi Teknik Informatika jenjang Strata-1 Universitas Bale Bandung. Dalam skripsi ini membahas mengenai Perancangan Aplikasi Absensi Fingerprint Menggunakan Algoritma Minutiae Untuk Meningkatkan Akurasi Dan Efisiensi Absensi Karyawan Pada PT. Visi Karya Prakarsa.

Laporan skripsi ini dapat dibuat dengan bantuan dari berbagai pihak uuntuk menyelesaikan tantangan dan hambatan selama pengerjaannya. Oleh karena itu, pada kesempatan kali ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan karunia-Nya selama proses pengerjaan laporan skripsi ini.
2. Kedua Orang Tua yaitu Bapak Dadang Rusmana dan Ibu Rita Nurhayati yang telah memberikan doa dan dukungan selama proses pembuatan laporan skripsi ini.
3. Bapak Yudi Herdiana, S.T., M.T selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
4. Bapak Yusuf Muharam, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Teknolgi Informasi Universitas Bale Bandung.
5. Bapak Mohammad Bayu Anggara, S.Kom., M.Kom. selaku dosen pembimbing utama.
6. Bapak Sutiyono S.T., M.Kom. selaku dosen pembimbing pendamping.
7. Ibu Rosmalina, S.T., M.Kom., selaku dosen penguji 1.
8. Bapak Sukiman, S.Tr.Kom., S.Pd., M.Kom., selaku dosen penguji 2.
9. Seluruh jajaran Dosen beserta Staff Fakultas Teknologi Informasi Universitas Bale Bandung.
10. Seluruh karyawan dan Staff PT. Visi Karya Prakarsa.
11. Rekan-reka FTI angkatan 2019 yang senantiasa saling membantu dan memberikan semangat dalam proses penelitian maupun penulisan laporan
12. Semua pihak yang memberikan dukungan dan bantuannya kepada penulis untuk menyelesaikan skripsi ini namun tidak bisa disebutkan satu persatu.
13. Iksan Zaenudin, yang senantiasa mendengarkan keluh kesah peneliti, memberi dukungan, serta motivasi, pengingat, yang selalu turut membantu peneliti dan menemani peneliti sehingga skripsi ini dapat terselesaikan dengan baik.
14. Siffa Rahmawati dan Herlin Martin Marcella selaku sahabat seperjuangan yang turut berkontribusi dalam penulisan penelitian ini.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini, namun penulis telah melakukan semua yang diperlukan dengan yang terbaik. Oleh karena itu, segala kritik dan saran yang membangun akan penulis terima dengan senang hati. Akhir kata, semoga laporan skripsi ini dapat diterima dan bermanfaat bagi kita semua.

Bandung, Agustus 2023

Penulis

# DAFTAR ISI

[ABSTRAK vi](#_Toc145695408)

[*ABSTRACT* vii](#_Toc145695409)

[KATA PENGANTAR viii](#_Toc145695410)

[DAFTAR ISI x](#_Toc145695411)

[DAFTAR GAMBAR xii](#_Toc145695412)

[DAFTAR TABEL xiv](#_Toc145695413)

[DAFTAR LAMPIRAN xv](#_Toc145695414)

[BAB I PENDAHULUAN 1](#_Toc145695415)

[1.1 Latar Belakang 1](#_Toc145695416)

[1.2 Rumusan Masalah 3](#_Toc145695417)

[1.3 Batasan Masalah 4](#_Toc145695418)

[1.4 Tujuan 4](#_Toc145695419)

[1.5 Metodologi Penelitian 4](#_Toc145695420)

[1.6 Sistematika Penulisan 5](#_Toc145695421)

[BAB II TINJAUAN PUSTAKA 7](#_Toc145695422)

[2.1 Landasan Teori 7](#_Toc145695423)

[2.2 Dasar Teori 11](#_Toc145695424)

[2.2.1 Sistem 11](#_Toc145695425)

[2.2.2 Absensi 11](#_Toc145695426)

[2.2.3 Presensi 11](#_Toc145695427)

[2.2.4 Absen 11](#_Toc145695428)

[2.2.5 *Fingerprint* 12](#_Toc145695429)

[*2.2.6* *Minutiae* 14](#_Toc145695430)

[2.2.7 Metode SDLC Waterfall 18](#_Toc145695431)

[2.2.8 *Unified Modelling Language* (UML) 20](#_Toc145695432)

[2.2.9 Balsamiq Mockups 4.7.3.0 23](#_Toc145695433)

[2.2.10 *Bootstrap* 23](#_Toc145695434)

[2.2.11 *SQL lite* 23](#_Toc145695435)

[2.2.12 *Web Server* 24](#_Toc145695436)

[2.2.13 Metode pengujian 24](#_Toc145695437)

[BAB IIIMETODOLOGI PENELITIAN 26](#_Toc145695438)

[3.1 Kerangka Pikir 26](#_Toc145695439)

[3.2 Deskripsi 27](#_Toc145695440)

[BAB IV PERANCANGAN DAN HASIL 31](#_Toc145695441)

[4.1 Analisis 31](#_Toc145695442)

[4.1.1 Analisis Sistem 31](#_Toc145695443)

[4.1.2 Analisis Masalah 31](#_Toc145695444)

[4.1.3 Analisis *Software* 32](#_Toc145695445)

[4.1.4 Analisis Pengguna 32](#_Toc145695446)

[4.1.5 *User Interface* 33](#_Toc145695447)

[4.1.6 Fitur-Fitur 33](#_Toc145695448)

[4.1.7 Analisis Data 34](#_Toc145695449)

[4.1.8 Analisis Biaya 35](#_Toc145695450)

[4.2 Perancangan 35](#_Toc145695451)

[BAB V IMPLEMENTASI DAN PENGUJIAN 63](#_Toc145695452)

[5.1 Implementasi 63](#_Toc145695453)

[5.2 Pengujian 99](#_Toc145695454)

[BAB VI KESIMPULAN DAN SARAN 103](#_Toc145695455)

[6.1 Kesimpulan 103](#_Toc145695456)

[6.2 Saran 104](#_Toc145695457)

[DAFTAR PUSTAKA 105](#_Toc145695462)

[LAMPIRAN 106](#_Toc145695463)

[RIWAYAT HIDUP PENULIS 110](#_Toc145695464)

# DAFTAR GAMBAR

[Gambar 2.1 Metode Waterfall 18](#_Toc146482131)

[Gambar 3.1 Kerangka Pikir 26](#_Toc135738045)

[Gambar 4.1 Use Case Diagram Aplikasi Absensi 35](#_Toc143188479)

[Gambar 4.2 *Activity* Diagram *Login* 45](#_Toc143188480)

[Gambar 4.3 *Activity* Diagram Karyawan 46](#_Toc143188481)

[Gambar 4.4 *Activity* Diagram Departemen 47](#_Toc143188482)

[Gambar 4.5 *Activity* Diagram Absensi 49](#_Toc143188483)

[Gambar 4.6 *Activity* Diagram Setting 50](#_Toc143188484)

[Gambar 4.7 *Activity* Diagram Laporan 51](#_Toc143188485)

[Gambar 4.8 *Class* Diagram Aplikasi Absensi 51](#_Toc143188486)

[Gambar 4.9 *User Interface login* 54](#_Toc143188487)

[Gambar 4.10 *User Interface Dashboard* 54](#_Toc143188488)

[Gambar 4.11 *User Interface* Data Karyawan 55](#_Toc143188489)

[Gambar 4.12 *User Interface* Edit Karyawan 55](#_Toc143188490)

[Gambar 4.13 *User* *Interface* *Delete* Karyawan 56](#_Toc143188491)

[Gambar 4.14 *User* *Interface* Tambah Karyawan 56](#_Toc143188492)

[Gambar 4.15 *User* *Interface* Departemen 57](#_Toc143188493)

[Gambar 4.16 *User* *Interface* Edit Departemen 57](#_Toc143188494)

[Gambar 4.17 *User* *Interface* *Delete* Departemen 58](#_Toc143188495)

[Gambar 4.18 *User* *Interface* Tambah Departemen 58](#_Toc143188496)

[Gambar 4.19 *User* *Interface* Data Absen 59](#_Toc143188497)

[Gambar 4.20 *User* *Interface* Edit Absen 59](#_Toc143188498)

[Gambar 4.21 *User* *Interface* *Delete* Absen 60](#_Toc143188499)

[Gambar 4.22 *User* *Interface* Tambah Absen 60](#_Toc143188500)

[Gambar 4.23 *User* *Interface* *Setting* 61](#_Toc143188501)

[Gambar 4.24 *User* *Interface* Edit *Setting* 61](#_Toc143188502)

[Gambar 4.25 *User* *Interface* Laporan 62](#_Toc143188503)

[Gambar 5.2 Halaman *login* 91](#_Toc143188402)

[Gambar 5.3 Halaman *Dashboard* 92](#_Toc143188403)

[Gambar 5.4 Halaman Karyawan 92](#_Toc143188404)

[Gambar 5.5 Halaman Edit Karyawan 93](#_Toc143188405)

[Gambar 5.6 Halaman *Delete* Karyawan 93](#_Toc143188406)

[Gambar 5.7 Halaman Tambah Karyawan 94](#_Toc143188407)

[Gambar 5.8 Halaman Departemen 94](#_Toc143188408)

[Gambar 5.9 Halaman Edit Departemen 95](#_Toc143188409)

[Gambar 5.10 Halaman *Delete* Departemen 95](#_Toc143188410)

[Gambar 5.11 Halaman Absensi 96](#_Toc143188411)

[Gambar 5.12 Halaman Edit Absen 96](#_Toc143188412)

[Gambar 5.13 Halaman *Delete* Absen 96](#_Toc143188413)

[Gambar 5.14 Halaman Tambah Absen 97](#_Toc143188414)

[Gambar 5.15 Halaman *Setting* 97](#_Toc143188415)

[Gambar 5.16 Halaman Edit *Setting* 98](#_Toc143188416)

[Gambar 5.17 Halaman Laporan 98](#_Toc143188417)

[Gambar 5.18 Halaman Hasil Print Laporan 98](#_Toc143188418)

# DAFTAR TABEL

[Tabel 2.1 Acuan Penelitian 8](#_Toc142855583)

[Tabel 2.2 *Pseudocode* *ridge* *bifurcation* dan *ridge* *ending* 14](#_Toc142855584)

[Tabel 2.3 *Use* *Case* Diagram 20](#_Toc142855585)

[Tabel 2.4 *Activity* Diagram 21](#_Toc142855586)

[Tabel 2.5 *Class* Diagram 22](#_Toc142855587)

[Tabel 4.1 Analisis Biaya 35](#_Toc143188537)

[Tabel 4.2 Deskripsi Aktor 36](#_Toc143188538)

[Tabel 4.3 Deskripsi *Use Case Login* 37](#_Toc143188539)

[Tabel 4.4 *Use* *Case* *Dashboard* 37](#_Toc143188540)

[Tabel 4.5 *Use Case* Karyawan 38](#_Toc143188541)

[Tabel 4.6 *Use Case* *Form* Edit Karyawan 39](#_Toc143188542)

[Tabel 4.7 *Use Case Form Delete* Karyawan 39](#_Toc143188543)

[Tabel 4.8 *Use Case* Departemen 40](#_Toc143188544)

[Tabel 4.9 *Use Case Form* Edit Departemen 40](#_Toc143188545)

[Tabel 4.10 *Use Case Form Delete* Departemen 41](#_Toc143188546)

[Tabel 4.11 *Use Case* Absensi 41](#_Toc143188547)

[Tabel 4.12 *Use Case Form* Edit Absen 42](#_Toc143188548)

[Tabel 4.13 *Use Case Form Delete* Absen 42](#_Toc143188549)

[Tabel 4.14 *Use Case Setting* 43](#_Toc143188550)

[Tabel 4.15 *Use Case Form* Edit *Setting* 43](#_Toc143188551)

[Tabel 4.16 Skenario *Use* *Case* Laporan 44](#_Toc143188552)

[Tabel 4.17 Tabel Karyawan 52](#_Toc143188553)

[Tabel 4.18 Tabel Departemen 52](#_Toc143188554)

[Tabel 4.19 Tabel Laporan 53](#_Toc143188555)

[Tabel 5.1 Spesifikasi perangkat keras 89](#_Toc143188560)

[Tabel 5.2 Spesifikasi perangkat lunak 89](#_Toc143188561)

[Tabel 5.3 Tabel pengujian aplikasi absensi karyawan 99](#_Toc143188562)

# DAFTAR LAMPIRAN

[Lampiran 1 Hasil Wawancara 106](#_Toc146130334)

[Lampiran 2 Hasil Dokumentasi 109](#_Toc146130335)

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

PT. Visi Karya Prakarsa adalah salah satu dari perusahaan di sektor teknologi informasi. Ini dimulai pada tahun 2020 dengan tujuan untuk menciptakan produk Antrique. Produk antrique adalah aplikasi antrian berbasis web dan mobile. PT. Visi Karya Prakarsa yang beralamat di Sekejati, Kec. Buah Batu, Kota Bandung, Jawa Barat, di Jl. Jupiter Barat 36 No.37. Karena perusahaan ini termasuk perusahaan yang masih muda, ada beberapa proses bisnisnya yang belum terkomputerisasi terutama kegiatan dalam presensi atau absensi karyawan. Absensi karyawan berperan penting untuk setiap proses aktivitas kerja. Absensi karyawan juga merupakan sarana informasi agar manajer yang relevan untuk bidang kedisiplinan setiap karyawan secara langsung. Melalui informasi karyawan dapat tumbuh memotivasi dan meningkatkan kepercayaan atasan terhadap karyawan.

Permasalahan yang ditemukan ketika melakukan observasi yaitu sistem absensi karyawan di PT. Visi Karya Prakarsa masih dilakukan secara manual dengan melakukan kegiatan *daily report* setiap pagi hari mengenai hasil pekerjaan karyawan sebagai patokan absensi karyawan di setiap harinya. Dalam absensi setiap harinya kemungkinan menimbulkan kecurangan yang tidak terduga, ketika melakukan *report* hasil pekerjaan karyawan tidak hadir atau hanya menitipkan absensi yang dilakukan antar karyawan. Seringkali sistem absensi yang dilakukan secara manual ini mengalami kehilangan pada data absensi, sehingga *leader* atau *team* yang melakukan report data bulanan mengalami kesulitan dalam melaporkan *report* bulanan karyawan dan tidak tersusun. Pengembangan suatu sistem absensi *modern* dengan penekanan pada integritas harus dijaga sebagai alternatif absensi manual, dapat menjadi sistem standar baru yaitu sistem absensi karyawan di PT.Visi Karya Prakarsa.

Penggunaan *fingerprint* untuk absensi karyawan dapat menjadi solusi yang lebih baik, begitu juga pada sistem absensi ini memiliki keterbatasan masalah diantaranya yaitu aplikasi absensi hanya dapat diaplikasikan dalam absensi karyawan dan memantau *report* harian, mingguan dan bulanan pada *website*. Aplikasi ini digunakan untuk mengelola absensi karyawan dan dibangun agar dapat bekerja secara dinamis yang diharapkan dapat digunakan untuk jangka panjang pada perusahaan.

Oleh karena itu untuk mengoptimalkan sistem absensi yang lebih efisien dan akurat di PT. Visi Karya Prakarsa, dirancang suatu sistem informasi absensi menggunakan *Fingerprint* yang dibantu suatu alat dengan *web* *server* mengelola *report* absensi, agar dapat mengoptimalkan absensi *fingerprint* dan meningkatkan efisiensi absensi karyawan.

Dalam perancangan aplikasi sistem absensi ini menggunakan metode Ekstraksi sidik jari *Minutiae-Based* yaitu proses pengambilan informasi yang relevan dari citra sidik jari untuk mengidentifikasi dan mengekstraksi fitur-fitur penting yang disebut *Minutiae*. *Minutiae* adalah titik-titik khusus pada sidik jari yang mencakup titik persimpangan *(bifurcation)* dan titik ujung *(ridge ending)*. Implementasi algoritma Ekstraksi sidik jari *Minutiae-based* pada absensi *fingerprint* di PT. Visi Karya Prakarsa alat tersebut dapat berguna untuk kegiatan absensi sekaligus membantu proses absensi agar lebih mengefisienkan waktu absensi dan meminimalisir kecurangan terhadap karyawan.

Pada beberapa penelitian terdahulu yang ditulis oleh Faiza Alif Fakhrina, Rahmadwati. Wijono dengan judul “*Thinning* *Zhang*-*Suen* dan *Stentiford* Untuk Menentukan Ekstraksi Ciri *(Minutiae)* Sebagai Identifikasi Pola Sidik Jari *Whorl* dan *Loop*” mengenai pengenalan pola sidik jari dan mengekstraksi tingkat keakurasian sistem dalam proses pengenalan sidik jari. Adapun jurnal terdahulu lainnya yang ditulis oleh Julius Santony dengan judul “*Minutiae Object Extraction in Fingerprint Image Using Morphological Methods Gabor Filters* (Ekstraksi Objek *Minutiae* Pada Citra Sidik Jari Dengan Metode Morfologi dan Gabor Filters” peneletian tersebut mengenai ekstraksi objek garis pada sidik jari agar dapat dilihat dengan jelas. Pada penelitian selanjutnya yang ditulis oleh Boldson Herdianto Situmorang, Gustian Rama Putra, Syaruf Hidayatullah dengan judul “Identifikasi Biometrika Menggunakan Ekstraksi *Minutiae* Pada Citra Sidik Jari” penelitian tersebut mengenai pengembangan sistem identifikasi biometrik berbasis citra sidik jari dengan memanfaatkan kamera ponsel untuk akuisisi citra sidik jari.

Pada penelitian ini bertujuan untuk merancang aplikasi absensi karyawan yang langsung mencetak hasil laporan absensi harian, mingguan dan bulanan. Selain itu aplikasi ini dirancang untuk mengurangi kecurangan dalam penitipan absensi antar karyawan dengan menggunakan sistem yang terkomputerisasi, dan mengimplementasikan aplikasi absensi pada PT.Visi Karya Prakarsa.

Pada pengujian perancangan aplikasi absensi ini, akan dilakukan pengujian sistem yaitu dengan menggunakan pengujian *Black Box* yaitu dengan metode pengujian identifikasi *fingerprint*, pengujian rekam data absensi, dan pengujian integrasi*.* Selain itu pengujian pada akurasi menggunakan algoritma ekstraksi *minutiae*.

Berdasarkan uraian diatas, identitas seseorang dapat ditentukan berdasarkan gambar pada sidik jarinya. Penelitian ini mengimplementasikan metode *Minutiae* dan Metode *Crossing Number* untuk pencocokan sidik jari yang lebih akurat, dan metode pengembangan sitem menggunakan SDLC *Waterfall*. Hasil penelitian ini memberikan saran mengenai absensi dengan memanfaatkan teknologi kepada PT. Visi Karya Prakarsa yaitu sebagai bahan pertimbangan dalam melakukan perbaikan sistem absensi karyawan. Dan untuk memudahkan manajer atau atasan dalam mengelola *report* harian, mingguan dan bulanan agar lebih cepat dan akurat. Oleh karena itu penulis mempertimbangkan untuk membuat aplikasi karyawan dengan judul “PERANCANGAN APLIKASI ABSENSI FINGERPRINT MENGGUNAKAN ALGORITMA *MINUTIAE* UNTUK MENINGKATKAN AKURASI DAN EFISIENSI ABSENSI KARYAWAN PADA PT. VISI KARYA PRAKARSA” dimana aplikasi ini diharapkan dapat mengoptimalkan kinerja karyawan dan dapat berguna untuk kegiatan absensi sekaligus membantu proses absensi agar lebih mengefisienkan waktu absensi dan meminimalisir kecurangan terhadap karyawan.

**1.2** Rumusan Masalah

Dari latar belakang yang telah dikemukakan di atas, maka perumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana merancang aplikasi absensi karyawan yang dapat langsung mencetak hasil laporan absensi bulanan dengan waktu yang cepat tetapi mendapatkan hasil yang akurat ?
2. Bagaimana mengurangi kecurangan absensi karyawan dengan menggunakan sistem yang terkomputerisasi ?
3. Bagaimana mengimplementasikan aplikasi absensi karyawan di PT. Visi Karya Prakarsa ?

## 1.3 Batasan Masalah

Adapun batasan masalah yang ada agar permasalahan sesuai dengan tujuan maka perlu ditetapkan batasan masalah sebagai berikut :

1. Aplikasi absensi hanya dapat diaplikasikan dalam absensi karyawan dan memantau report pada website.
2. Aplikasi ini digunakan untuk mengelola absensi karyawan.
3. Aplikasi ini dibangun untuk bekerja secara dinamis dan untuk penggunaan jangka panjang pada perusahaan.

## 1.4 Tujuan

Berdasarkan rumusan masalah dan batasan masalah diatas, maka tujuan penelitian yang ingin dicapai yaitu :

1. Untuk merancang aplikasi absensi karyawan yang langsung mencetak hasil laporan absensi harian, mingguan dan bulanan.
2. Untuk mengurangi kecurangan dalam penitipan absensi antar karyawan dengan menggunakan sistem yang terkomputerisasi.
3. Untuk mengimplementasikan aplikasi absensi karyawan di PT. Visi Karya Prakarsa.

## Metodologi Penelitian

* + 1. Pengumpulan data

Metode pengumpulan data yang penulis ambil yaitu melakukan observasi secara langsung dan mengamati bagaimana proses sistem absensi pada PT. Visi Karya Prakarsa. Wawancara, dengan metode wawancara penulis menanyakan beberapa pertanyaan mengenai absensi karyawan kepada manajer di perusahaan tersebut. Studi pustaka pengumpulan data pada penelitian yang berkaitan dengan analisis dari berbagai tulisan jurnal dan buku yang telah terbit.

* + 1. Perancangan

Waterfall adalah salah satu metodologi pengembangan perangkat lunak. Dalam pendekatan *waterfall* (air terjun), masing-masing tahapan harus diselesaikan sebelum melanjutkan ke langkah berikutnya. Langkah-langkah dalam model *waterfall* termasuk analisis sistem, desain, implementasi, pengujian, penyebaran dan penegakan hukum. Setiap langkah memiliki keluaran yang sesuai yang berfungsi sebagai petunjuk untuk langkah berikutnya, dan perubahan signifikan pada kebutuhan atau pola perilaku sering tidak dapat dilakukan sampai langkah sebelumnya selesai.

## 1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini akan membantu untuk mendapatkan informasi tentang tugas akhir yang akan dilakukan dan agar penulisan laporan ini tidak menyimpang dari batas-batas masalah yang ada, oleh karena itu struktur laporan ini seperti yang diharapkan.

Sistematis penulisan laporan akhir ini adalah sebagai berikut :

1. BAB I Pendahuluan

Pada bab ini memberikan latar belakang penelitian dan penjelasan umum tentang masalah yang ditemukan peneliti, rumusan masalah, batasan masalah, tujuan penelitian tugas akhir, metodologi penelitian tugas akhir, dan sistematika penulisan.

1. BAB II Tinjauan Pustaka

Tinjauan literatur ini memberikan teori dan konsep dasar terkait dengan topik yang dibahas, seperti pengertian dari absensi, fingerprint, pengenalan fingerprint berbasis detail, laporan penelitian sebelumnya, beberapa modul pada metode RAD dan perangkat pendukung penyebaran perangkat keras lainnya. Penulis sebagai dasar untuk bekerja dengan sistem.

1. BAB III Metodologi Penelitian

Metode penelitian ini meliputi langkah-langkah pembuatan sistem berupa bahan, metode dan langkah kerja yang penulis lakukan untuk menyelesaikan pekerjaan ini, dijelaskan dengan bantuan sistem dan alat.

1. BAB IV Analisis, Perancangan dan Hasil

Bab ini membahas tentang metode yang telah ditempuh oleh penulis untuk membuat sistem yang direncanakan berupa rangkaian alat dan fungsi yang tergabung dalam sistem dan rangkaian hasil pengujian sistem.

1. BAB V Implementasi dan Pengujian

Bab ini berisi mengenai proses penggabungan pada implementasi sistem, spesifikasi sistem, instalasi sistem dan proses menjalankan sistem yang sudah di rancangan dan hasil pengujian pada sistem yang telah dirangkai.

1. BAB VI Kesimpulan dan Saran

Bab ini berisi kesimpulan yang diambil dari hasil analisis data pada bab – bab sebelumnya. Pihak yang memberikan saran dianggap berguna dan dapat membangun serta mengembangkan isi laporan sesuai dengan tujuan penelitian skripsi ini.

# BAB II TINJAUAN PUSTAKA

## 2.1 Landasan Teori

Landasan teori meliputi abstrak yang disusun berdasarkan jurnal yang berkaitan dengan penelitian yang dilakukan. Dalam penelitian ini, penulis mereview penelitian terdahulu sebagai referensi penelitian, berikut jurnal yang digunakan dalam studi :

Pada beberapa penelitian terdahulu oleh Faiza Alif Fakhrina, Rahmadwati,

Wijono pada tahun 2016 dengan judul “*Thinning Zhang-Suen* dan *Stentiford* Untuk Menentukan Ekstraksi Ciri *(Minutiae)* Sebagai Identifikasi Pola Sidik Jari *Whorl* dan *Loop*”. Peneliti tersebut menguji dan menganalisis pengenalan pola sidik jari dan mengekstraksi tingkat keakurasian sistem dalam proses pengenalan sidik jari. Dengan menggunakan beberapa metode untuk melakukan pengujian tersebut seperti metode *thinning* yang dapat mengurangi waktu proses sistem, pada metode *thinning* yang digunakan yaitu algoritma *thinning* paralel *Zhang*-*Suen* dan *Stentiford*, hasil dari metode *Stentiford* cenderung mengikuti bentuk objek, maka citra *thinning* yang dihasilkan sesuai dengan citra asli. Adapun metode LDA *(Linear Discriminant Analysis)* yaitu metode *class-specific linear* dimana dapat melakukan transformasi pereduksian dimensi dari elemen-elemen yang merupakan anggota sebuah kelas dikelompokkan bersama didalam ruang dimensi rendah. Metode ekstraksi sidik jari *Minutiae-based* digunakan untuk mengetahui lokasi titik *minutiae* secara efektif dan akurat. Pada kenyataannya piksel yang sesuai dengan *minutiae* berdasarkan *crossing number*.

Pada penelitian lainnya dilakukan oleh Julius Santony pada tahun 2020 dengan judul “*Minutea Object Extraction in Fingerprint Image Using Morphological Methods and Gabor Filters* (Ekstraksi Objel Minutea Pada Citra Sidik Jari Dengan Metode Morfologi dan Gabor Filter)”. Peneliti menguji ekstraksi objek garis pada sidik jari agar dapat dilihat dengan jelas, Penelitian yang dilakukan di citra sidik jari melibatkan deteksi objek dan target menggunakan metode morfologis, ekstraksi objek *minutiae* dengan menggunakan proses filter *gabor*, dan ekstraksi dengan tahapan penggabungan, binerisasi, *thinning* dan *Minutiae Extraction* dapat memperjelas garis sidik jari yang ditunjukkan pada 10 citra yang diuji, menghasilkan lebih sedikit tanda dan percabangan daripada citra aslinya, sehingga sidik jari lebih jelas dan mudah dikenali.

Adapun peneliti lainnya dilakukan oleh Boldson Herdianto Situmorang, Gustian Rama Putra, Syarif Hidayatullah pada tahun 2023 dengan judul “Identifikasi Biometrika Menggunakan Ekstraksi *Minutiae* Pada Citra Sidik Jari”. Peneliti tersebut menguji hasil keakuratan ekstraksi *minutiae* menggunakan konsep *crossing* *number* dan untuk pengukuran kemiripan menggunakan *minutiae* *based* *matching*. Peneliti menyimpulkan bahwa kamera ponsel dengan spesifikasi 5 dan 8 megapiksel dapat digunakan untuk Proses pengambilan gambar sistem biometrik berbasis sidik jari. Proses ekstraksi fitur hasil gambar menggunakan metode *Minutiae Crossing Number* dan proses *match* pengukuran menggunakan metode *Minutiae Based Matching* menghasilkan nilai akurasi sebesar 92,8% untuk kamera 5 megapiksel dan 95,3% untuk kamera 8 megapiksel.

Tabel 2.1 Acuan Penelitian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Judul Penelitian | Masalah | Metode | Solusi |
| 1 | *Thinning Zhang-Suen* dan *Stentiford* Untuk Menentukan Ekstraksi Ciri *(Minutiae)* Sebagai Identifikasi Pola Sidik Jari *Whorl* dan *Loop* | pengenalan pola sidik jari dan mengekstraksi tingkat keakurasian sistem dalam proses pengenalan sidik jari | metode *crossing number*, *core* dan *delta*, serta *Center Point Location* dan metode *thinning* | Memanfaatkan metode *crossing number*, analisis sidik jari dilakukan dengan memeriksa ciri-ciri yang dimiliki oleh sidik jari. *Thinning* merupakan teknik yang sangat penting karena dapat mengurangi waktu startup sistem. Selain itu, metode *thinning* tidak dapat mengubah struktur sidik jari dan memperkuat letak punggung (*ridge*) sidik jari dan ciri *falley* atau bukit. |
| 2 | *Minutea Object Extraction in Fingerprint Image Using Morphological Methods and Gabor Filters* (Ekstraksi Objel Minutea Pada Citra Sidik Jari Dengan Metode Morfologi dan Gabor Filter) | Untuk ekstraksi objek garis pada sidik jari agar dapat dilihat dengan jelas, dan untuk memudahkan proses identifikasi pada sidik jari. | Ekstraksi *minutiae,* metode *morfology,* ektrasksi objek *minutiae* dengan proses *gabor filter* dan *minutiae extraction* | Dengan menggunakan algoritma ekstraksi objek minutiae dapat memperjelas objek pada sidik jari agar dapat menampilkan garis sidik jari dengan jelas. |
| 3 | Identifikasi Biometrika Menggunakan Ekstraksi *Minutiae* Pada Citra Sidik Jari | Pada penelitian ini adalah untuk mengembangkan sistem identifikasi biometrik berdasarkan sidik jari dengan memanfaatkan kamera ponsel pintar untuk menangkap gambar sidik jari. | *minutiae crossing number, minutiae based matching.* | Metode yang digunakan dalam pengolahan citra terdiri dari pencocokan berbasis *minutiae* dan *minutiae crossing number* untuk proses pengeluaran ciri dan pengeluaran kemiripan. Proses ekstraksi ciri hasil citra menggunakan metode *minutiae crossing number* dan proses pengukuran kecocokan menggunakan metode *minutiae based matching* sehingga dapat menghasilkan nilai akurasi, dan nilai akurasi tergantung dari hasil tahap akuisisi citra, *pre*-*processing*, nilai *threshold* pada proses identifikasi dan jumlah citra yang digunakan pada data *training* di *database*. |

## 2.2 Dasar Teori

Dalam perancangan Aplikasi Absensi Fingerprint Menggunakan Algoritma Minutiae untuk Meningkatkan Akurasi dan Efisiensi pada PT. Visi Karya Prakarsa, pastinya memiliki metode atau teori dasar yang dapat digunakan dalam aplikasi absensi tersebut antara lain yaitu :

### 2.2.1 Sistem

Menurut jurnal T. Henny Febriana Harumy, Juaha Sitorus, Meliza Lubis mengenai sistem, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan sasaran yang tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan di dalam sistem. (Harumy, T.H.F., Julham Sitorus, 2018)

### 2.2.2 Absensi

Pengertian absensi menurut Rizaldi, Fazrin Muhammad absensi merupakan suatu model untuk mencatat jam kedatangan dan jam kepulangan seseorang. Absensi pada umumnya digunakan pada lingkungan yang formal seperti sekolah, maupun lingkungan pekerjaan. Tujuan dari penggunaan absensi adalah untuk memantau data kehadiran serta sebagai sarana untuk mengukur tingkat kedisiplinan seseorang. (Rizaldi, 2022)

### 2.2.3 Presensi

Pengertian menurut jurnal Mohammad Arya Rosyid Sikumbang, Roni Habibi, Syafrial Fachri Pane mengenai presensi, presensi pegawai adalah suatu kegiatan pencatatan terhadap setiap kehadiran pegawai dengan tujuan untuk mengetahui data yang berkaitan dengan kehadiran masuk dan keluar pegawai secara periodik baik harian maupun bulanan. Data presensi pegawai didapatkan dari penerapan teknologi *fingerprint* yang sudah diterapkan oleh perusahaan Badan Pusat Statistik Kota Bandung. (Sikumbang et al., 2020)

### 2.2.4 Absen

Pengertian Absen menurut jurnal Mohammad Arya Rosyid Sikumbang, Roni Habibi, Syafrial Fachri Pane mengenai absen, absen adalah pegawai tidak dapat masuk dikarenakan beberapa macam hal diantaranya karena sakit, izin, alpa atau cuti yang merupakan menjadi parameter dalam menentukan kedisiplinan pegawai di perusahaan Badan Pusat Statistik Kota Bandung, akan tetapi, berdasarkan parameter dalam menentukan kedisiplinan pegawai tersebut kerja dinas kerap dijadikan menjadikan alasan untuk melakukan absen pada perusahaan. Sehingga berpengaruh dalam penentuan kedisiplinan pegawai (Sikumbang et al., 2020).

### 2.2.5 *Fingerprint*

Pengertian menurut *Fingerprint* atau sidik jari adalah guratan-guratan yang terdapat pada kulit ujung jari. Sidik jari manusia terdiri dari dua buah lapisan: lapisan luar (epidermis) dan lapisan dalam (dermis). Lapisan dermis bertanggung jawab untuk menumbuhkan lapisan epidermis yang akan menumbuhkan sel-sel ke permukaan jari. *Ridge* adalah tekstur yang menonjol (puncak) dan *valley* adalah tekstur yang tidak menonjol (lembah). (Meyliana, 2020)

* + - 1. Pengenalan Sidik Jari *(Fingerprint Recognition)*

Pengenalan sidik jari menggambarkan proses memperoleh representasi digital sidik jari dan membandingkan dengan versi digital sidik jari yang tersimpan. Sistem pengenalan sidik jari menjadi tiga tahap. Pertama adalah tahap *Preprocessing,* untuk peningkatan kualitas gambar. Tahap kedua adalah *Descriptors Extraction,* untuk mengekstraksi fitur dan gambar hasil *pre-processing.* Tahap ketiga adalah *Mathcing,* untuk mencocokkan citra input dengan citra *template* pada *dataset*. (Dewanti et al., 2020)

1. *Image Acquisition*

Tahap akuisisi adalah proses untuk mendapatkan citra sidik jari awal dengan cara online maupun offline (PRASASTI et al., 2020).

1. *Preprocessing*

*Pre-processing* adalah proses menghilangkan data yang tidak diinginkan dalam gambar sidik jari seperti *noise* atau refleksi. Pada tahap ini, citra sidik jari diproses untuk mendapatkan hasil citra dengan kualitas yang lebih baik. *Preprocessing* memiliki dua langkah proses yaitu mentransformasikan gambar asli sidik jari menjadi gambar yang diberi filter abu-abu *(Grayscale)* dan menghilangkan *noise*  pada gambar asli (Dewanti et al., 2020).

Konversi *Grayscale* adalah proses konversi citra dengan orientasi warna RGB menjadi citra dengan orientasi warna abu-abu atau *grayscale*. Proses ini dilakukan dengan menggunakan fungsi ‘rgb2gray’ pada aplikasi matlab atau dengan cara melakukan operasi perhitungan menggunakan persamaan 13.

*Grayscale* = 0.2989 \* R + 0.5870 \* G + 0.1140 \* B (13)

Nilai R, G dan B didapat dari data citra dengan orientasi warna RGB. (Putra et al., 2023)

1. *Ekstraksi* Fitur

Setelah tahap prapemprosesan selesai, tahap selanjutnya adalah tahap ekstraksi fitur atau ekstraksi ciri. Fitur yang akan diambil dari citra sidik jari adalah berupa *minutiae*, yaitu titik-titik yang terbentuk oleh guratan-guratan kulit pada sidik jari. Titik-titik yang terbentuk oleh guratan kulit tersebut memiliki banyak jenisnya. Namun pada penelitian ini, jenis titik yang akan dijadikan sebagai ciri dari sidik jari adalah *ridge ending* dan *ridge bifurcation. Ridge ending* menggambarkan berakhirnya guratan sidik jari dan *ridge bifurcation* menandakan percabangan dari guratan kulit pada sidik jari (Bansal). Fitur ini adalah salah satu yang membuat sidik jari memiliki sifat universal, unik, permanen dan kuantitatif. Oleh karena itu, proses ekstraksi fitur dilakukan dengan menggunakan metode *ekstraksi minutiae* dengan konsep *crossing number.* (Putra et al., 2023)

Pola *ridge* yang biasanya digunakan dalam representasi sidik jari adalah *ridge* *end* dan *ridge* *bifurcations* seperti pada Tabel 2.2. Secara sederhana, *ridge* *bifurcation* merupakan percabangan dari pola *ridge* dan *ridge* *ending* adalah akhir dari pola *ridge*. *Psudocode* identifikasi *ridge bifurcation* dan *ridge ending* dapat dijelaskan sebagai berikut (Hendri Himawan Triharminto, 2019)

Tabel 2.2 Pseudocode ridge bifurcation dan ridge ending

|  |
| --- |
| Input -> citra thinning  For i = 1 : jumlah baris  For j = 1 : jumlah kolom  %% RIDGE BIDURCATION  Jika i = j  Periksa piksel p2, p3, p4, p5, p6  Beri bobot 1 apabila salah satu piksel p1, p2, p3, p4, p5, p6 = 255  Jumlah bobot total piksel terdekat  Jika bobot = 3 *maka ridge bifurcation*  %%RIDGE ENDING  Jika i = j  Periksa piksel p2, p3, p4, p5, p6  Beri bobot 1 apabila salah satu piksel p1, p2, p3, p4, p5, p6 = 255  Jumlah bobot total piksel terdekat  Jika bobot = 1 *maka ridge end*  End  End |

1. *Matching*

Proses identifikasi citra sidik jari dengan menggunakan metode *minutiae* proses ini menghitung skor kecocokan antara data hasil ekstraksi *minutiae* pada citra uji dengar *ekstraksi minutiae* yang tersimpan pada *database* (Putra et al., 2023)*.*

### *Minutiae*

Menurut jurnal Afzalul Patriot Nasution, Andrew Brian Osmond, dan Randy Efra Saputra *Minutiae* adalah karakteristik dari pola pada suatu percabangan dari pola *(ridge bifurcation)* atau suatu pemberhentian pola sidik jari *(ridge ending).* Ada dua jenis utama *minutiae* pada sidik jari yaitu sebagai berikut :

1. *Ridge* *Ending*

Titik ujung adalah titik di mana garis bergelombang pada sidik jari berakhir. Mereka merupakan titik paling ujung pada garis-garis yang membentuk pola sidik jari. Titik ujung bisa berbentuk seperti titik tunggal atau sedikit melengkung.

1. *Ridge* *Bifurcation*

Persimpangan adalah titik di mana garis-garis pada sidik jari bercabang. Mereka merupakan titik di mana satu garis utama terbagi menjadi dua atau lebih cabang. Persimpangan biasanya memiliki bentuk seperti garpu atau bercabang.

Semua point pada minutiae dapat didefinisikan dengan rumus berikut :

𝐶𝑁 = 0, 5 ∑ | 8𝑖 = 1 𝑃𝑖 − 𝑃𝑖 + 1| 𝑃9 = 𝑃1

Dimana :

P1 = Dimana P9. Ini didefinisikan sebagai jumlah perbedaan terbesar kedua antara murid yang identik dan murid di kelas tersebut. Dengan menggunakan data properti CN seperti yang ditunjukkan pada tabel, titik-titik punggungan dapat diklasifikasikan sebagai ujung punggungan, percabangan, atau non-minutiae. (Nasution et al., 2018)

Karakter sidik jari pada manusia antara lain :

1. *Ridge* merupakan garis yang menonjol pada permukaan jari.
2. *Valley* merupakan daerah lembah di permukaan jari.
3. *Minutiae* atau titik perhentian *(Ending)* atau titik percabangan *(Bifurcation)* untuk setiap *Ridge*. (Iwan Sugiarto, 2015)
   * + 1. *Ekstraksi* sidik jari *Minutiae*

Pada penelitian oleh Faiza Alif Fakhrina, Rahmadwati, metode ekstraksi *minutiae* pada sidik jari biasanya menggunakan citra skala abu-abu kemudian dikonversi ke citra biner, lalu dilakukan proses *thinning* untuk mendapatkan garis *ridge* pada sidik jari menjadi skeleton. Diantara semua ciri sidik jari, *minutiae* *point* memiliki posisi orientasi yang unik untuk membedakan sidik jari, representasi ciri *minutiae* dapat mengurangi masalah pencocokan pola sidik jari. Untuk mendapatkan tingkat akurasi yang baik pada *minutiae* dengan menggunakan bermacam kualitas citra sidik jari. Maka algoritma deteksi *minutiae* digunakan untuk mengetahui lokasi titik *minutiae* secara efektif dan akurat. Pada kenyataannya piksel yang sesuai dengan *minutiae* berdasarkan beda *crossing number* dari 2 (Faiza Alif Fakhrina, Rahmadwati, Rahmadwati, 2016). *Crossing number* (cn(p)) dari piksel p pada citra biner adalah setengah penjumlahan dari beda antara bagian piksel pada 8-tetangga dari p :

dimana :

: piksel yang mendefinisikan 8-tetangga dari *p*

Keterangan :

CN = Crossing Number

Pi = Banyaknya Piksel

Dengan menggunakan metode *crossing number*, analisis *minutiae* akan menghasilkan data detail menit yang terdiri dari dua jenis detail menit, yaitu penghentian punggungan dan percabangan. Hasilnya kemudian digunakan untuk merumuskan rencana untuk membasmi titik-titik *minutiae* yang dianggap salah dari beberapa aturan yaitu :

1. Jika sebuah percabangan memiliki lebih dari dua kolom dan panjangnya di luar ambang batas yang dipertimbangkan, maka piksel yang bersangkutan tidak dianggap sebagai percabangan.
2. Jika terdapat patahan pendek (jarak antara patahan dengan patahan tetangganya tidak lebih dari 1 piksel), maka piksel tersebut tidak dikenali sebagai percabangan.
3. Jika terdapat beberapa patahan (bifurkasi atau *ridge* *ending*) dalam satu klaster yang simetris, maka semua patahan tersebut harus simetris, tetapi hanya satu patahan yang simetris dengan titik pusat klaster yang diberikan.
4. Apabila detail *minutiae* terlihat dan tidak memiliki guratan (tonjolan) untuk mendukungnya, maka detail *minutiae* tersebut dikatakan hapus. (Putra et al., 2023)
   * + 1. Algoritma *Crossing Number*

Algoritma *Crossing* *Number* (CN) adalah satu-satunya algoritma yang digunakan dalam pemrosesan sidik jari untuk secara sistematis memeriksa detail-detail kecil kecocokan. Asumsi yang mendasari algoritma ini adalah setiap titik ujung pada sidik jari akan memiliki nomor persilangan kutub yang konsisten, tergantung pada arah garis-garis yang mendekat.

Banyak orang menggunakan Konsep *Crossing* *Number* (CN) untuk ekstraksi *minutiae*. Biasanya, untuk setiap jendela 3x3, jika piksel pusat adalah 1 dan ada tiga piksel tetangga yang bernilai 1, maka piksel pusat adalah percabangan. Jika piksel pusat bernilai 1 dan hanya ada satu piksel tetangga yang bernilai 1, maka piksel pusat tersebut adalah sebuah titik akhir.

Pada bab ini, rata-rata kesejajaran garis alur (D) juga akan dibahas. Rasio antara dua garis alur yang dibelokkan atau ditambatkan adalah rasio antara garis alur. Nilai D terhitung dengan empat perbandingan berikut.

dengan M dan N, ada beberapa bar dan citranya yang telah dipoles untuk negara tersebut. Li dan Bi adalah panjang baris ke I dan jumlah gambar bernilai 1 pada baris ke I. Lj dan kj adalah panjang baris ke j dan jumlah piksel bernilai 1 pada baris ke j.

Setiap detail kecil dibedakan berdasarkan parameter koordinat x, koordinat y, dan orientasi. Penghitungan orientasi untuk percabangan dilakukan secara terfokus. Empat garis alur terpisah yang terhubung ke titik percabangan memiliki busur yang bergantian. Percabangan dibagi menjadi tiga titik akhir.

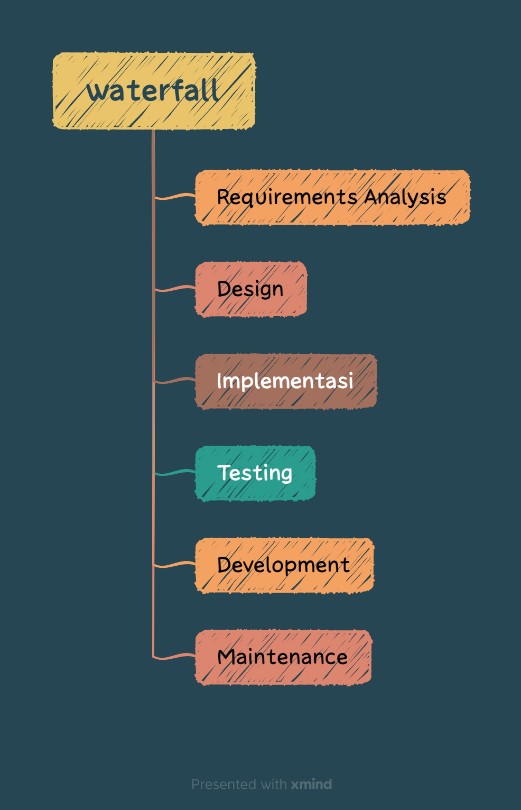
Tiga titik akhir yang baru terdiri dari tiga piksel tetangga dari titik percabangan dan tiga kantong dari garis alur yang terhubung ke percabangan sebelum dikirim.

* + - 1. *Minutiae Matching*

Menurut jurnal Gunarti Purba mengenai *minutiae* *matching* pada tahap ini yang tersisa pada gambar akan dibandingkan dengan *minutiae* yang ada pada *database* yang sebelumnya yang dimana memiliki atribut dan juga lokasi yang berbeda-beda. Untuk menunjukkan persamaan antara sidik jari pada jari yang sama. (Purba, 2021)

### 2.2.7 Metode SDLC Waterfall

SDLC (*System Development Life Cycle)* atau siklus hidup pengembangan sistem adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem – sistem tersebut. Metode pengembangan sistem metode SDLC atau sering disebut sebagai pendekatan air terjun (*waterfall*). Dalam model *Waterfall,* setiap tahap harus berurutan, dan tidak dapat melompat ke tahap berikutnya. *Waterfall* harus menyelesaikan tahap pertama baru lanjut ke tahap ke dua dan seterusnya. Setiap langkah memiliki keluaran yang bertindak sebagai pemegang tempat untuk langkah berikutnya, memungkinkan kemajuan pengembangan sistem yang tidak dapat diprediksi. Namun, karena perubahan harus dilakukan pada titik waktu sebelumnya, model air terjun agak tidak stabil jika ada perubahan yang diperlukan di tengah jalan.



Gambar 2.1 Metode Waterfall

Gambar diatas merupakan contoh metode *waterfall* yaitu salah satu metode yang digunakan dalam perancangan sistem absensi karyawan.

1. Analisis Kebutuhan (*Requirement Analyst)*

Analisis sistem adalah mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi dalam pelaksanaan. Tentunya juga mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan, yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan – perbaikannya.

1. Desain (*Design)*

Desain adalah perancangan bangun implementasi yang menggambarkan bagaimana suatu sistem dapat dibentuk yang berupa gambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Menyangkut di dalamnya konfigurasi komponen perangkat lunak dan perangkat keras dari suatu sistem.

1. Implementasi (*implementation)*

Tahap implementasi, yaitu desain program diterjemahkan ke dalam kode – kode instruksi yang akan dijalankan komputer. Menerjemahkan hasil proses perancangan menjadi sebuah bentuk program komputer yang dimengerti oleh mesin komputer. Penerjemahan ke bahasa komputer menggunakan bahasa pemrograman dalam menjalankan instruksi yang disusun.

1. Uji coba (*Testing)*

Uji coba program merupakan elemen yang kritis dari SQA *(Software Quality Assurance)* dan mempresentasikan tinjauan ulang yang menyeluruh terhadap spesifikasi, desain dan pengkodean. Uji coba mempresentasikan ketidak normalan yang terjadi pada pengembangan software.

1. Penyebaran (*Development)*

Pada tahap development yaitu mengoperasikan program di lingkungannya saat produk atau aplikasi dinyatakan fungsional dan bisa digunakan secara langsung oleh pengguna. Tahap development dilakukan dengan menyiapkan fasilitas fisik dan pengguna aplikasi.

1. Pemeliharaan Sistem (*Maintenance)*

Pada tahap pemeliharaan sistem ini setelah aplikasi berhasil dan siap pakai maka harus melakukan pembenaran dan penyempurnaan software.

### *Unified Modelling Language* (UML)

Menurut dalam perancangan pada penelitian digunakan UML *(Unified Modeling Language)* sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. UML *(Unified Modeling Language)* adalah sebuah bahasa pemodelan visual yang digunakan untuk mendokumentasikan, merancang, dan menggambarkan struktur, perilaku dan interaksi dari sebuah sistem perangkat lunak. UML adalah standar industri yang luas digunakan dalam pengembangan perangkat lunak untuk mengkomunikasikan dan memodelkan desain sistem secara visual.

1. *Use Case Diagram*

*Use case* *diagram* adalah jenis diagram yang paling sering digunakan dalam analisis dan perancangan sistem untuk mengungkapkan interaksi antara aktor (pengguna atau sistem eksternal) dan sistem yang sedang dianalisis. Diagram ini membantu dalam memvisualisasikan fungsionalitas yang diharapkan dari sebuah sistem yang kacau.

Tabel 2. 3 Use Case Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Symbol | Nama | Keterangan |
| 1 |  | *Actor* | Mewakili peran orang lain, sistem atau alat saat berkomunikasi dengan *use case* |
| 2 |  | *Use Case* | Abstraksi dan interaksi antara sistem dan aktor |
| 3 |  | *Association* | Abstraksi dan penghubung antara aktor dengan *use case* |
| 4 |  | *Extend* | Menunjukkan bahwa suatu *use case* sepenuhnya melengkapi fungsionalitas *use case* lainnya |
| 5 |  | *Generalization* | Menunjukan spesialisasi aktor dengan *use case* |
| 6 |  | *Include* | Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari use case lainnya. |

1. *Activity Diagram*

*Activity diagram* adalah jenis diagram pemodelan proses yang paling umum digunakan untuk menggambarkan logika atau aktivitas aktual di dalam proses atau sistem tertentu. Diagram ini memberikan representasi visual tentang bagaimana beberapa aktivitas berinteraksi dan saling terkait satu sama lain ketika dievaluasi.

Tabel 2.4 Activity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Keterangan |
| 1. |  | Status awal | Sebuah diagram aktivitas memiliki sebuah status awal |
| 2 |  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
| 3 |  | Percabangan | Percabangan dimana ada pilihan aktivitas yang lebih dari satu |
| 4 |  | Penggabungan | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu |
| 5 |  | Status akhir | Status akhir yang diimplementasikan oleh sistem, diagram aktivitas dengan status akhir |

3. *Class* *Diagram*

*Class Diagram* merupakan suatu diagram yang digunakan untuk menampilkan kelas-kelas berupa paket-paket untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya. *Class* *diagram* desain modelnya dibagi menjadi 2 bagian. *Class* *diagram* yang pertama merupakan jabaran dari domain model yang merupakan abstraksi dari basis data. *Class* diagram yang kedua merupakan bagian dari modul program MVC pattern *(Model View Controller)*, dimana terdapat *class* *boundary* sebagai *class* *interface, class control* sebagai tempat menemukannya algoritma, dan *class entitas* sebagai tabel dalam basis data dan *query* program.

Tabel 2.5 Class Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Keterangan |
| 1. | |  | | --- | | Nama\_kelas | | + atribut | | *+* operasi | | Kelas | Kelas pada struktur sistem |
| 2 |  | Antarmuka / *interface* | Sama dengan konsep *interface* dalam pemrograman berorientasi objek |
| 3 |  | Asosiasi | Relasi antar *class diagram* |
| 4 |  | Asosiasi berarah | Realisasi antar kelas menggunakan nilai makna kelas tunggal yang digunakan oleh kelas berikutnya. |

### Balsamiq Mockups 4.7.3.0

Balsamiq adalah salah satu dari beberapa alat yang digunakan oleh desainer UI dan UX untuk mempengaruhi desain proyek yang akan datang. Aplikasi ini memiliki tingkat adopsi yang sangat rendah karena tidak memerlukan kode untuk berfungsi. Menggunakan alat ini dalam aplikasi memungkinkan Anda untuk dengan cepat menarik dan melepaskan 32 elemen desain yang diperlukan, membuat proses desain menjadi lebih efisien. Balsamiq juga dapat digunakan untuk membuat prototipe interaktif untuk situs web atau aplikasi yang saat ini sedang dikembangkan. Fitur ini memungkinkan desainer untuk memahami kegunaan pekerjaan mereka sehingga revisi dapat disesuaikan di kemudian hari.

### 2.2.10 Bootstrap

Bootstrap adalah kerangka kerja HTML, CSS dan JavaScript yang sumber terbuka dan bebas untuk merancang aplikasi web. Kerangka kerja ini berisi *template* desain berbasis HTML dan CSS untuk tipografi, formulir, tombol, navigasi dan komponen antarmuka lainnya, serta juga ekstensi opsional JavaScript. Tidak seperti kebanyakan kerangka kerja web lainnya, kerangka kerja inu fokus pada pengembangan front-end saja. Kelebihan bootstrap :

1. Dapat mempercepat waktu proses pembuatan *front-end website.*
2. Tampilan bootstrap yang sudah cukup terlihat modern.
3. Tampilan bootstrap sudah responsive, sehingga mendukung segala jenis resolusi, baik itu pc, tablet dan juga smartphone.
4. Website menjadi sangat ringan ketika diakses, karena bootstrap dibuat dengan sangat terstruktur.

### SQL lite

SQLite adalah sebuah perpustakaan (library) yang menyediakan sistem manajemen basis data berbasis berkas (file-based database management system). Ini berarti bahwa SQLite tidak memerlukan server terpisah untuk mengelola basis data, melainkan mengandalkan sebuah berkas tunggal yang berisi semua informasi basis data. SQLite terkenal karena ukurannya yang kecil, efisiensinya, dan kemampuannya untuk menyimpan basis data dalam berkas tunggal yang mudah diintegrasikan ke dalam berbagai aplikasi.

### 2.2.12 Web Server

Web server adalah sebuah software yang berfungsi untuk menerima dan melayani permintaan yang dikirimkan user melalui browser kemudian ditampilkan kepada user sesuai dengan permintaan yang dikirimkan server. Bisa dikatakan web server adalah pusat kontrol yang berfungsi untuk memproses permintaan yang diterima dari browser.

### Metode pengujian

* + - 1. Metode Pengujian *Black Box Testing*

Pengujian *Black Box Testing* adalah teknik pengujian peraplikasi yang dilakukan tanpa pengetahuan atau akses ke struktur internal atau logika sistem yang sedang dievaluasi. Dalam percobaan kotak hitam, fokus utamanya adalah pada input yang diberikan ke sistem dan output yang diperoleh tanpa memahami bagaimana sistem mencapai hasil yang diinginkan.

* + - 1. Metode pengujian penelitian

Ada beberapa metode untuk menganalisis data yang dapat digunakan untuk menentukan hasil investigasi terhadap perancangan yang tidak dapat dilihat, di antaranya:

1. Metode pengujian identifikasi *fingerprint*

Metode yang hanya menggunakan sidik jari untuk memverifikasi kemampuan sistem dalam menganalisa dan menggunakan sidik jari dari setiap pengguna yang terdaftar di dalam sistem.

1. Pengujian rekam data absensi

Metode ini digunakan untuk memastikan bahwa sistem dapat mengumpulkan data yang akurat dan mengirimkannya dengan cara yang aman.

1. Pengujian integrasi

Metode untuk mengidentifikasi kemampuan sistem autentikasi sidik jari untuk menyinkronkan data yang dienkripsi dengan sistem manajemen kunci atau sistem lain yang terkait.

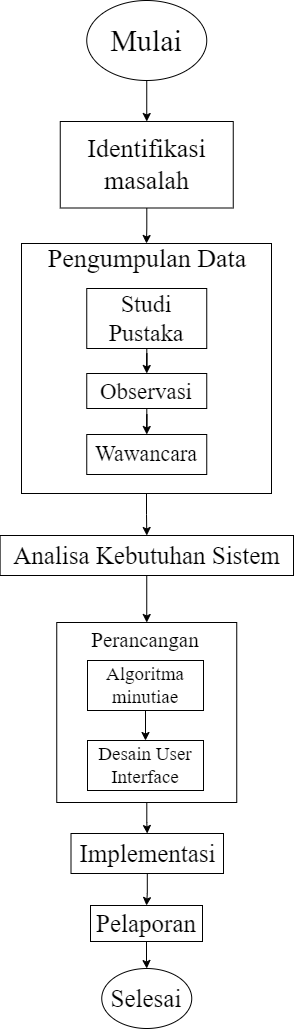
* + - 1. Metode pengujian alat

Ada beberapa teknik untuk memodifikasi objek yang dapat digunakan untuk memodifikasi sistem absensi objek. Beberapa teknik yang umum digunakan tercantum di bawah ini:

1. Pengujian Fungsionalitas: Metode ini digunakan untuk mengetahui apakah sebuah objek dapat berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan. Pengujian dilakukan sambil memastikan bahwa semua fitur dan fungsi yang diharapkan dapat bekerja dengan baik telah tersedia.
2. Pengujian Kinerja: Metode ini digunakan untuk menilai tingkat kinerja suatu organisasi dalam hal waktu respons, kecepatan pemrosesan data, dan tingkat staf. Untuk memastikan bahwa alat dapat berfungsi dengan baik dalam situasi yang sulit, pengujian dilakukan dengan menempatkan alat dalam kondisi beban kerja yang menantang.
3. Pengujian Keamanan: Metode ini digunakan untuk mengetahui tingkat keamanan alat saat melakukan enkripsi data karyawan dan data absensi. Pengujian dilakukan dengan memastikan bahwa alat tersebut memiliki fitur keamanan yang kuat, seperti enkripsi data, proteksi anti spyware, dan kontrol akses khusus untuk pengguna.

# BAB IIIMETODOLOGI PENELITIAN

## 3.1 Kerangka Pikir



Gambar 3.1 Kerangka Pikir

## 3.2 Deskripsi

Deskripsi merupakan sebuah kaidah yang mempunyai hubungan dengan adanya upaya pengolahan data menjadi sebuah hal dapat dikemukakan dan diutarakan dengan cara yang jelas serta tepat guna mencapai suatu tujuan tertentu sehingga nantinya dapat dimengerti dan dipahami oleh pembaca.

* + 1. Identifikasi Masalah

Tahap pertama dalam merancang aplikasi absensi dalam kerangka pikir penulis adalah menemukan masalah atau merumuskan masalah. Dalam hal ini penulis meminta izin kepada pihak perusahaan untuk melakukan penelitian kemudian melakukan observasi dan wawancara untuk mengetahui permasalahan pada PT. Visi Karya Prakarsa untuk penulis terapkan.

* + 1. Pengumpulan Data

1. Studi Pustaka

Pengumpulan data pada penelitian ini berkaitan dengan analisis dari berbagai tulisan dalam jurnal ilmiah dan buku yang sudah terbit dan berkaitan dengan topik penelitian yang penulis ambil. Teknik pada pengumpulan data dilakukan dengan membaca buku-buku, referensi dan jurnal-jurnal. Pada penelitian ini penulis menggunakan beberapa jurnal dengan menggunakan metode algoritma ekstraksi sidik jari *minutiae-based* untuk mengetahui lokasi titik *minutiae* secara efektif dan akurat. Pada kenyataannya piksel yang sesuai dengan *minutiae* berdasarkan beda *crossing number*.

1. Observasi

Metode pengambilan data dengan cara langsung mengamati dan mencatat pada objek yang dipelajari pada PT. Visi Karya Prakarsa. Metode observasi pada penelitian ini proses identifikasi dan mencatat data yang relevan secara langsung pemantauan organisasi yang terkait dengan objek untuk penelitian.

1. Wawancara

Metode pengumpulan data dengan melakukan wawancara yang relevan dengan topik penelitian. Metode wawancara dapat dilakukan dengan mengumpulkan informasi tentang absensi karyawan dari manajer dan pihak yang terkait lainnya.

* + 1. Analisa Kebutuhan Sistem

Setelah mendapatkan data kuesioner, tahap selanjutnya adalah merancang dan merencanakan aplikasi absensi yang dimulai dengan analisis kebutuhan perangkat lunak dan perangkat keras yang akan digunakan dalam perancangan aplikasi absensi. Pada tahap ini dilakukan pengumpulan kebutuhan elemen-elemen di tingkat perangkat lunak. Dengan desain harus dapat ditentukan data atau informasi, fungsi, proses atau prosedur yang diperlukan serta untuk sistem kerjanya.

* + 1. Perancangan

1. Perancangan Diagram

Tahap ini diselesaikan dengan membuat desain menggunakan bahasa pemodelan yang dikenal sebagai *Unified* *Modeling* *Language* (UML). UML digunakan dalam penelitian ini dan terdiri dari:

1. *Use* *case* diagram digunakan untuk menunjukkan hubungan aktor dan *use* *case* yang paling umum serta fungsi-fungsi apa saja yang mungkin ada di dalam sebuah sistem perancangan aplikasi absensi ini, diantaranya :

* Identifikasi aktor utama seperti karyawan dan admin
* Identifikasi *use* *case* utama seperti absensi karyawan, lihat laporan absensi, atur jadwal kerja, dan sebagainya.

1. *Activity diagram* digunakan untuk merepresentasikan aktivitas yang ada pada sebuah sistem dan mencakup gambaran setiap aktivitas yang dilakukan pengguna aplikasi. Prosedur identifikasi aktivasi utama yang akan penulis gambarkan dalam diagram di bawah ini:
2. Prosedur admin
3. Prosedur untuk identita
4. Prosedur untuk absensi
5. Prosedur untuk mencatat jadwal kerja.
6. *Class diagram* untuk menggambarkan struktur sistem dari strategi pendefinisian kelas sekolah yang akan dibangun untuk membangun sistem. *Class* *diagram* yang akan dibuat adalah sebagai berikut:

* *Class admin*
* *Class* identitas karyawan
* *Class* absensi
* *Class* jadwal kerja

1. Perancangan Antarmuka

Pada tahap ini dalam proses antarmuka, desain aplikasi dibuat menggunakan figma yang lebih canggih, tetapi masih dapat mengilustrasikan fungsionalitas aplikasi dan, yang lebih penting, menunjukkan bagaimana interaksi pengguna dan sistem akan terjadi di dalamnya.

* + 1. Implementasi

Sistem sidik jari dibuat melalui empat proses diantaranya yaitu :

1. Akuisisi, pengambilan sidik jari melalui alat pemindai atau *fingerprint,* setelah itu citra diproses ke dalam sistem ekstraksi. Pada tahap sistem, citra sidik jari tersebut diekstraksi dengan menggunakan metode *minutiae* untuk dapat dicocokan dengan *database*.
2. *Pre Processing,* citra yang diperoleh pada tahap akuisisi diproses untuk meningkatkan kualitas citra dan memudahkan identifikasi citra sidik jari. Kemudian masuk ke tahan Segmentasi, yaitu pada tahap ini citra menghasilkan citra biner dengan dua nilai *grayscale* yaitu hitam dan putih. Tahap ketiga adalah proses *Thinning* yaitu proses mengurangi jumlah piksel pada gambar.
3. Ekstraksi minutiae, mencari ciri yang optimal dari suatu objek yang membedakan dari objek lain.
4. Pencocokan sidik jari *(Matching)*, tujuan dari sistem pencocokan adalah untuk menggunakan template dan masukan dari dua sidik

jari untuk menentukan apakah sidik jari menunjukkan jari yang sama.

Setelah melakukan perancangan pada alat, tahap selanjutnya adalah mengimplementasikan rancangan yang telah dilakukan dengan menggunakan bahasa pemrograman phyton dan *database* menggunakan SQLlite.

* + 1. Pengujian

Pada tahap ini dibuat untuk memastikan hasilnya sesuai dengan desain yang diinginkan. Jika masih ada masalah, kembali ke tahap implementasi dan lakukan koreksi yang diperlukan hingga semuanya bekerja sebagaimana mestinya. Pengujian pada sistem menggunakan pengujian metode *Black Box Testing,* pada pengujian penelitian menggunakan pengujian identifikasi *fingerprint*, pengujian rekam data absensi, dan pengujian integrasi. Sedangkan pengujian alat menggunakan metode fungsionalitas, kinerja dan pengujian keamanan.

* + 1. Pelaporan

Langkah terakhir adalah pembuatan laporan skripsi sebagai satu-satunya elemen yang diterima sesuai dengan kriteria yang ditentukan dalam formulir pengajuan proposal disertasi untuk Fakultas Teknologi dan Informasi di Universitas Bale Bandung.

# BAB IV PERANCANGAN DAN HASIL

## Analisis

Analisis dilakukan sebagai tahapan pertama dalam penelitian untuk mengidentifikasi kebutuhan dan persyaratan aplikasi. Berikut ini adalah analisis program yang dimulai oleh penulis untuk membantu dalam pengembangan aplikasi berpikir abstrak. Analisis ini akan mencakup analisis masalah, analisis perangkat lunak, analisis pengguna, analisis kesesuaian fungsi, analisis data, dan analisis biaya.

### 4.1.1 Analisis Sistem

Analisis sistem dilakukan untuk memberikan solusi terhadap permasalahan yang ada di PT. Visi Karya Prakarsa. Analisis ini juga bertujuan untuk membandingkan sistem yang sedang berjalan saat ini. Sementara itu, perancangan desain sistem bertujuan memberikan gambaran atau pandangan yang jelas sesuai proses desain sistem dari awal hingga akhir penelitian. Pengguna pada aplikasi ini meliputi admin dan user hanya untuk absensi.

### 4.1.2 Analisis Masalah

Langkah pertama adalah menganalisis sistem yang sudah beroperasi dengan tujuan untuk memahami lebih jelas bagaimana sistem bekerja dan masalah apa saja yang perlu diselesaikan untuk memulai sistem yang baru.

Berdasarkan analisis langsung pada sistem kerja absensi yang sedang berjalan di PT. Visi Karya Prakarsa ditemukan beberapa kendala yaitu sistem absensi karyawan di PT. Visi Karya Prakarsa masih dilakukan secara manual dengan melakukan kegiatan *daily report* setiap pagi hari mengenai hasil pekerjaan karyawan sebagai patokan absensi karyawan di setiap harinya. Dalam absensi setiap harinya kemungkinan menimbulkan kecurangan yang tidak terduga, ketika melakukan *report* hasil pekerjaan karyawan tidak hadir atau hanya menitipkan absensi yang dilakukan antar karyawan. Seringkali sistem absensi yang dilakukan secara manual ini mengalami kehilangan pada data absensi, sehingga *leader* atau *team* yang melakukan report data bulanan mengalami kesulitan dalam melaporkan *report* bulanan karyawan dan tidak tersusun.

### 4.1.3 Analisis Software

Berdasarkan analisis kebutuhan maka untuk memenuhi kebutuhan dalam pengembangan aplikasi absensi karyawan memerlukan *software* sebagai alat penunjang pengembangan aplikasi absensi karyawan*.* Adapun *software* yang dibutuhkan sebagai alat penunjang pada penelitian ini adalah sebagai berikut:

1. Sistem operasi Windows 11
2. *Framework* bootstrapdigunakan untuk mengembangkan aplikasi situs web dengan cepat dan mudah*.*
3. Replit sebagai *text* editordigunakan untuk melakukan pengkodean dalam pembuatan aplikasi absensi, sekaligus replit digunakan untuk web server (django).
4. Web browser digunakan untuk melihat tampilan web aplikasi absensi yang sudah dibuat.
5. Draw.io digunakan untuk merancang diagram UML.
6. Balsamiq Mockups digunakan untuk membuat rancangan antarmuka aplikasi absensi.

### 4.1.4 Analisis Pengguna

Penganalisaan pengguna adalah yang berkaitan dengan yang akan memakai aplikasi absensi ini. Pengguna aplikasi absensi adalah salah satu karyawan yang ada khususnya *leader/team* yang berkaitan dengan sistem absensi, karena pengguna aplikasi absensi diharuskan menguasai hal-hal yang bersangkutan dengan data karyawan, data absensi dan pembuatan laporan absensi sehingga dapat menggunakan aplikasi dengan baik sebagaimana fungsinya. Agar dapat mewujudkan tujuan dari dibuatnya aplikasi absensi yang bisa memberikan kemudahan dan kenyamanan bagi penggunanya untuk mengelola laporan absensi yang akurat dan cepat. Maka strategi yang akan diterapkan dalam aplikasi absensi ini adalah dilihat dari sisi user interface dan fitur-fitur yang diterapkan dalam aplikasi absensi tersebut.

### 4.1.5 *User Interface*

Efektivitas dan keramahan antarmuka pengguna aplikasi bergantung pada pengalaman pengguna sewaktu menggunakannya. Sebagai akibatnya, alat yang dibuat dengan tujuan untuk memberikan informasi yang tepat waktu dan akurat tidak lagi efektif, sehingga antarmuka pengguna juga harus dirancang untuk memenuhi kebutuhan pengguna akan penjelasan yang jelas dan dapat dimengerti tentang konsep-konsep yang kompleks. Maksud dan tujuan dari penyediaan informasi ini adalah untuk memungkinkan aplikasi menjadi sumber daya yang dapat diandalkan yang dapat digunakan oleh pengguna untuk memahami upah yang berlaku di perusahaan.

Berikut adalah *user interface* aplikasi absensi menggunakan metode minutiae sebagai pengukur kinerja kerja di perusahaan PT. Visi Karya Prakarsa, meliputi :

1. *User interface* halaman *Login*
2. *User interface* halaman Karyawan
3. *User interface* halaman Bagian
4. *User interface* halaman Absen
5. *User interface* halaman *Setting*
6. *User interface* halaman *Log out*

### 4.1.6 Fitur-Fitur

Fitur-fitur yang digunakan dalam perancangan aplikasi absensi ini secara keseluruhan dirancang untuk memudahkan pengguna dalam mengelola absensi karyawan, menyajikan laporan absensi, dan merekap data absensi. Berikut adalah beberapa fitur yang disediakan oleh aplikasi absensi pada PT. Visi Karya Prakarsa.

1. Halaman dashboard yang berfungsi sebagai halaman utama

Fitur-fitur yang digunakan dalam perancangan aplikasi absensi karyawan ini dimaksudkan agar karyawan pada perusahaan tersebut dapat memudahkan karyawan dalam absensi, pihak manajemen juga dapat memantau jam masuk dan jam keluar setiap karyawan nya, serta memudahkan manager dalam merekap data absensi karyawan pada setiap bulannya.

1. Halaman karyawan yang menampilkan data karyawan yang dimana isi dari data karyawan tersebut terdiri dari NIP, nama ID sidik jari, no telepon, bagian dan action untuk edit dan hapus. Dan tambah karyawan yang berisi input data yang diperlukan.
2. Halaman departemen menampilkan data departemen setiap karyawan nya dan tambah bagian untuk menambah data bagian ketika diperlukan.
3. Halaman absensi meliputi data absen yang memuat nip, nama, tanggal absen, jam datang, jam pulang, status dan *action*.
4. Halaman *setting* ini meliputi jam datang dan jam pulang untuk mengedit ulang jam kerja pada perusahaan tersebut.
5. Halaman laporan ini meliputi pelaporan absensi karyawan didalamnya berisi jumlah kehadiran setiap karyawan.

### 4.1.7 Analisis Data

Untuk mengimplementasikan perancangan aplikasi absensi dengan baik, penulis membutuhkan beberapa data untuk perancangan aplikasi dan penelitian. Ada beberapa data yang digunakan dan didistribusikan oleh aplikasi absensi, berikut ini adalah data-data tersebut:

* 1. Data Admin

Analisis data yang akan di dapat oleh admin yaitu terstrukturnya sistem absensi sehingga admin dapat memantau data absensi karyawan, jam kerja menjadi lebih efektif sehingga semua karyawan dapat datang tepat waktu dan admin dapat memantau karyawan yang terlambat masuk pada jam kerja. Selain itu pada analisis pihak admin juga dapat membuat laporan absensi karyawan perbulannya.

* 1. Data karyawan

Analisis karyawan hanya untuk absensi karyawan yang dilakukan dengan menggunakan alat *fingerprint* yang dimana hasil dari citra sidik jari akan menghasilkan informasi jam masuk dan jam keluar pada laporan dalam aplikasi. Dengan itu admin akan mengetahui jam masuk dan jam keluar setiap karyawan nya.

### 4.1.8 Analisis Biaya

Analisis biaya untuk pengembangan sistem. Diantaranya, biaya yang dibutuhkan dalam proses perancangan aplikasi ini diantaranya adalah :

Tabel 4.1 Analisis Biaya

|  |  |  |
| --- | --- | --- |
| No. | Jenis Kebutuhan | Biaya |
| 1. | Internet | Rp. 500.000 |
| 2 | Alat *Fingerprint* | Rp. 120,000 |
| 3. | Analisis | Rp. 500.000 |
| 4. | *Programming* | Rp. 1.500.000 |
| 5. | Biaya ATK | Rp. 350.000 |
| Jumlah | | Rp. 2.970.000 |

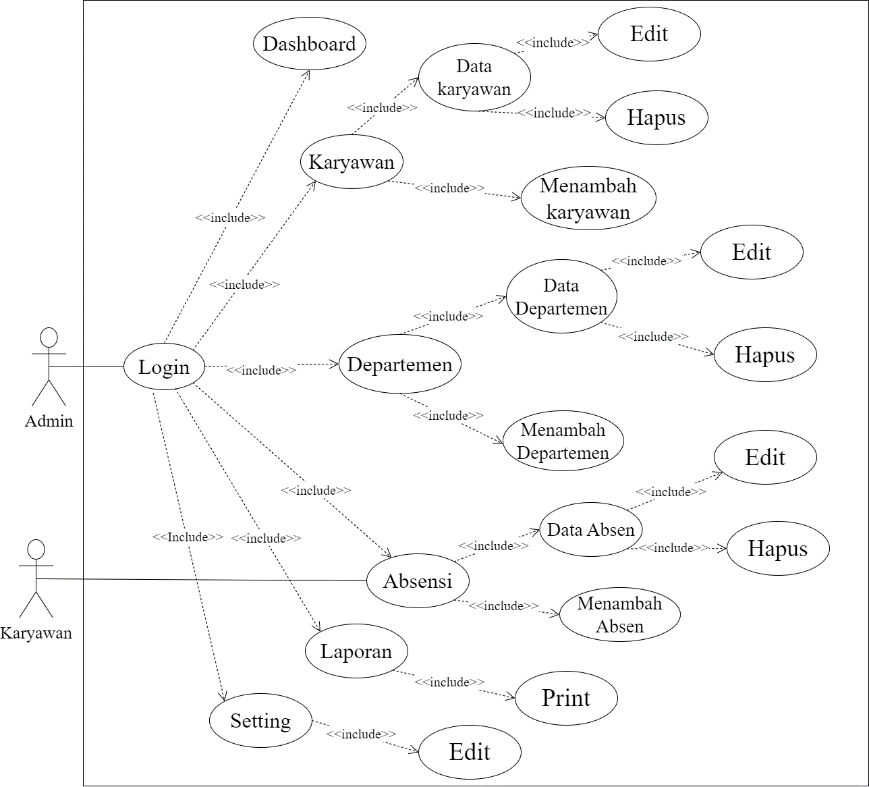
## Perancangan

Sebelum masuk dalam pembuatan aplikasi dibutuhkan perancangan yaitu merancang *software* dalam bentuk UML yang terdiri dari *use case*, *activity* *diagram*, dan *class diagram.*

* + 1. UML *(Unfied Modelling Language)*

1. *Use Case Diagram*

*Use case diagram* merupakan skenario dari interaksi antara pengguna dengan sistem. Dapat terlihat apa saja yang dapat dilakukan admin untuk mengelola data yang terdapat dari perancangan aplikasi absensi karyawan.



Gambar 4.1 Use Case Diagram Aplikasi Absensi

Penjelasan dari gambar *use case diagram* diatas dapat dijelaskan dalam tabel deskripsi berikut :

1. Deskripsi Aktor

Tabel 4.2 Deskripsi Aktor

|  |  |
| --- | --- |
| Aktor | Deskripsi |
| Admin | Admin merupakan pengguna yang dapat mengakses atau mengelola data pada sistem secara keseluruhan. |
| *User* | Adalah aktor yang memiliki hak akses terbatas. Dalam aplikasi absensi ini *user* hanya dapat mengakses absen kehadiran saja dengan menggunakan alat *fingerprint* untuk menginputkan data absen karyawan ke aplikasi untuk pelaporan data kehadiran karyawan |

1. Deskripsi *Use Case*

|  |  |
| --- | --- |
| *Use* *Case* | Deskripsi |
| *Login* | Sebelum mengakses aplikasi, proses awal diselesaikan dengan meng*input*kan *username* dan *password* pengguna. |
| *Dashboard* | Menu utama yang menampilkan dashboard dari aplikasi absensi yang berfungsi sebagai penghubung ke menu sekunder. |
| Karyawan | Proses untuk melihat data karyawan secara keseluruhan yang telah di*inputkan*. |
| Departemen | Proses untuk melihat data departemen secara keseluruhan untuk meng*input*kan departemen pada setiap karyawan |
| Absensi | Pada proses ini digunakan untuk melihat data kehadiran pada karyawan, juga dapat menambah absensi karyawan untuk absensi izin keterangan sakit saja. |
| Laporan | Proses pengolahan data karyawan yang telah di*input*kan dan di rekap perbulannya, proses ini untuk mencetak laporan kehadiran karyawan. |

1. **Skenario *usescase login***

Tabel 4.3 Deskripsi Use Case Login

|  |  |
| --- | --- |
| Nama | Login |
| Admin | Admin |
| **Skenario Utama** | |
| **Kondisi awal** | Aktor belum memiliki akses pada aplikasi |
| **Aktor** | **Sistem** |
| Mengakses halaman *login* | Menampilkan halaman *login* |
| Mengisi form *login* | Memvalidasi username dan password. Jika berhasil, maka akan menampilkan halaman utama |
| **Kondisi Akhir** | Hak akses pengguna sudah dipastikan sebagai admin maka akan menampilkan halaman utama pada aplikasi absensi. |

1. **Skenario *use case dashboard***

Tabel 4.4 Use Case Dashboard

|  |  |
| --- | --- |
| Nama | *Dashboard* |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses halaman *dashboard* |
| **Aktor** | **Sistem** |
| Admin mengisi form login terlebih dahulu | Setelah berhasil *login* admin akan menampilkan halaman *dashboard* |
| **Kondisi Akhir** | Menampilkan informasi pada halaman utama sesuai dengan hak akses pengguna |

1. **Skenario *use case* Karyawan**

Tabel 4.5 Use Case Karyawan

|  |  |
| --- | --- |
| Nama | Karyawan |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses data karyawan |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data karyawan | Menampilkan form input data karyawan |
| Admin dapat menambah karyawan | Input data karyawan yang akan ditambahkan dan submit |
| **Kondisi Akhir** | Data karyawan berhasil ditambahkan kedalam *database* |

1. **Skenario use case *form* edit karyawan**

Tabel 4.6 Use Case Form Edit Karyawan

|  |  |
| --- | --- |
| Nama | Edit karyawan |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses data karyawan |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data karyawan | Menampilkan tabel data karyawan |
| Admin memilih data karyawan yang akan di edit | Data berhasil di edit |
| **Kondisi Akhir** | Data karyawan berhasil diubah dari *database* |

1. **Skenario use case form *delete* karyawan**

Tabel 4.7 Use Case Form Delete Karyawan

|  |  |
| --- | --- |
| Nama | *Delete* karyawan |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses data karyawan |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data karyawan | Menampilkan tabel data karyawan |
| Admin memilih data karyawan yang akan di *delete* | Data berhasil di *delete* |
| **Kondisi Akhir** | Data karyawan berhasil di *delete* dari *database* |

1. **Skenario *use case* Departemen**

Tabel 4.8 Use Case Departemen

|  |  |
| --- | --- |
| Nama | Departemen |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses data departemen |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data departemen | Menampilkan form input data departemen |
| Admin dapat menambah departemen | Input data departemen yang akan ditambahkan dan submit |
| **Kondisi Akhir** | Data departemen berhasil ditambahkan kedalam *database* |

1. **Skenario *use case form* edit departemen**

Tabel 4.9 Use Case Form Edit Departemen

|  |  |
| --- | --- |
| Nama | Edit departemen |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses data departemen |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data departemen | Menampilkan tabel data departemen |
| Admin memilih data departemen yang akan di edit | Data berhasil di edit |
| **Kondisi Akhir** | Data departemen berhasil diubah dari *database* |

1. **Skenario *use case form delete* departemen**

Tabel 4.10 Use Case Form Delete Departemen

|  |  |
| --- | --- |
| Nama | *Delete* bagian |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses data departemen |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data departemen | Menampilkan tabel data departemen |
| Admin memilih data karyawan yang akan di *delete* | Data berhasil di *delete* |
| **Kondisi Akhir** | Data departemen berhasil di *delete* dari *database* |

1. **Skenario use case absensi**

Tabel 4.11 Use Case Absensi

|  |  |
| --- | --- |
| Nama | Absensi |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses data absensi |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data absen | Menampilkan form input data absen |
| Admin dapat menambah absen | Input data absen yang akan ditambahkan dan submit |
| **Kondisi Akhir** | Data absen berhasil ditambahkan kedalam *database* |

1. **Skenario *use case form* edit absen**

Tabel 4.12 Use Case Form Edit Absen

|  |  |
| --- | --- |
| Nama | Edit absen |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses data absen |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data absen | Menampilkan tabel data absen |
| Admin memilih data absen yang akan di edit | Data berhasil di edit |
| **Kondisi Akhir** | Data absen berhasil diubah dari *database* |

1. **Skenario *use case form delete* absen**

Tabel 4.13 Use Case Form Delete Absen

|  |  |
| --- | --- |
| Nama | *Delete* absen |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses *delete* absen |
| **Aktor** | **Sistem** |
| Admin memilih data absen yang akan dihapus | Menampilkan tabel data absen |
| Admin memilih data karyawan yang akan di *delete* | Data berhasil di *delete* |
| **Kondisi Akhir** | Data karyawan berhasil di *delete* dari *database* |

1. **Skenario *use case setting***

Tabel 4.14 Use Case Setting

|  |  |
| --- | --- |
| Nama | *Setting* |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses *setting* |
| **Aktor** | **Sistem** |
| Admin masuk ke menu data *setting* | Menampilkan form input data *setting* |
| **Kondisi Akhir** | Data absen berhasil ditambahkan kedalam *database* |

1. **Skenario *use case form* edit *setting***

Tabel 4.15 Use Case Form Edit Setting

|  |  |
| --- | --- |
| Nama | Edit *setting* |
| Aktor | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Aktor berhasil login sebagai admin dan dapat mengakses edit *setting* |
| **Aktor** | **Sistem** |
| Admin masuk ke menu *setting* | Menampilkan tabel data *setting* |
| Admin memilih data yang akan di edit | Data berhasil di edit |
| **Kondisi Akhir** | Data *setting* berhasil diubah dari *database* |

1. **Skenario use case laporan**

Tabel 4.16 Skenario Use Case Laporan

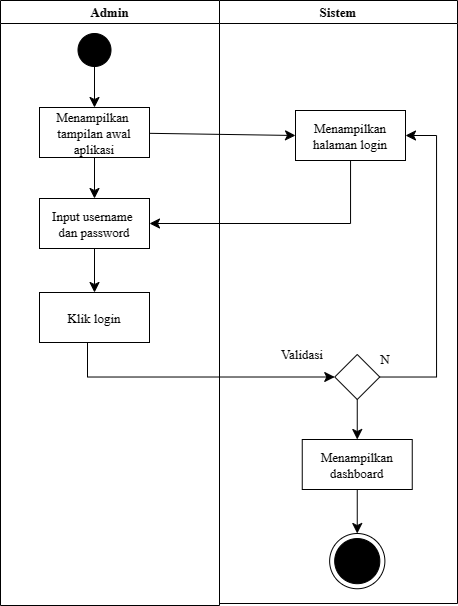
|  |  |  |
| --- | --- | --- |
| Nama | Laporan | |
| Aktor | Admin | |
| **Skenario Utama** | | |
| **Kondisi Awal** | | Aktor berhasil login sebagai admin dan dapat mengakses laporan |
| **Aktor** | | **Sistem** |
| Admin masuk ke menu laporan | | Menampilkan halaman print laporan |
| Admin klik print | | Data absensi karyawan berhasil ditampilkan |
| **Kondisi Akhir** | | Admin dapat mengunduh file absensi atau bisa hanya lihat data nya saja |

1. *Activity Diagram*

*Activity diagram* menggambarkan aliran aktivitas dalam sebuah sistem. Berikut ini adalah diagram aktivitas dari aplikasi absensi:

1. *Activity* *diagram* *Login*

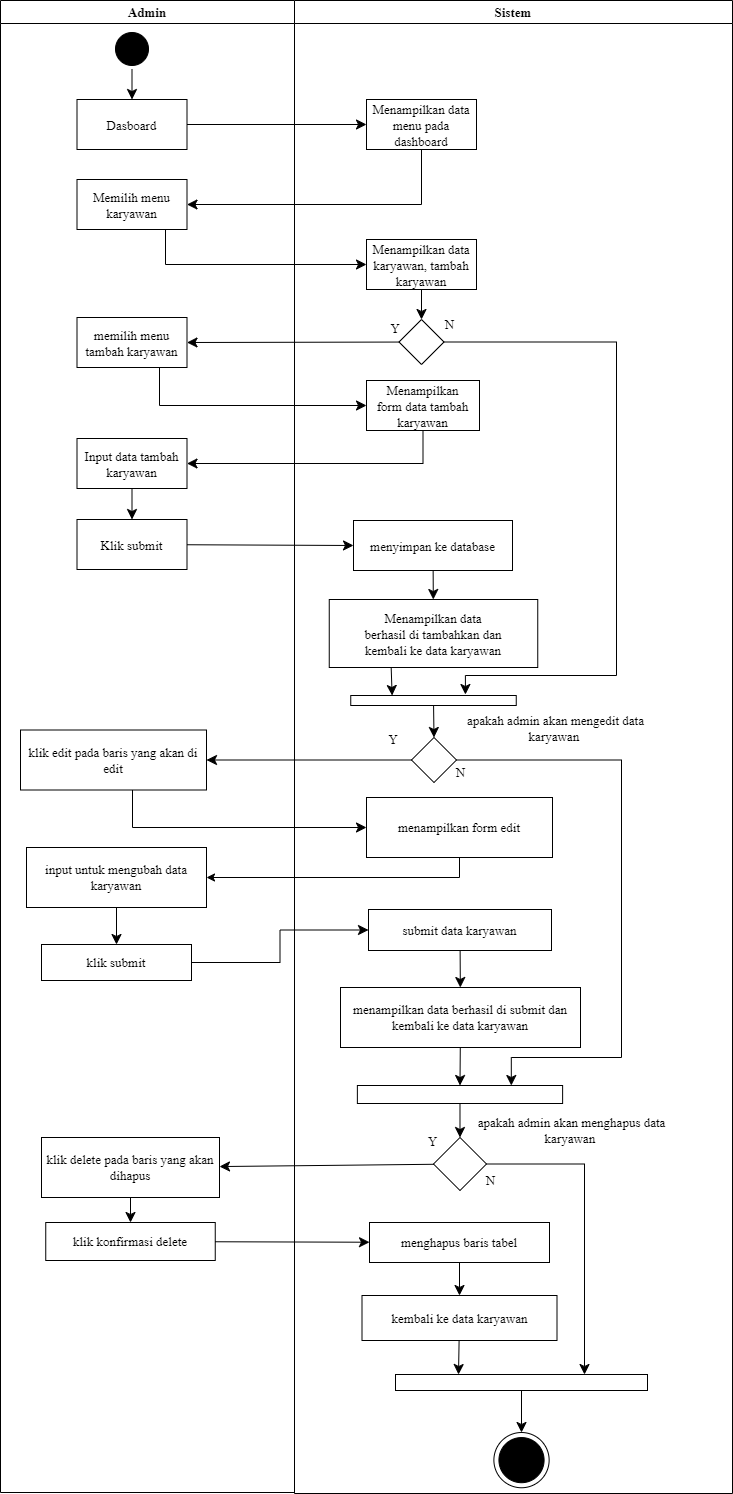
Pada gambar 4.2 menunjukan bahwa admin akan melakukan *login* pada aplikasi absensi dengan menginputkan *username* dan *password.* Setelah *login* kemudian sistem akan melakukan validasi, apabila sesuai dengan data yang sudah diinputkan maka akan menampilkan halaman *dashboard.* Tetapi jika data yang diiputkan tidak sesuai maka sistem akan menampilkan *form login* kembali dan menampilkan pemberitahuan didalamnya jika data yang di*input*kan tidak *valid*.



Gambar 4.2 Activity Diagram Login

1. *Activity* *diagram* karyawan

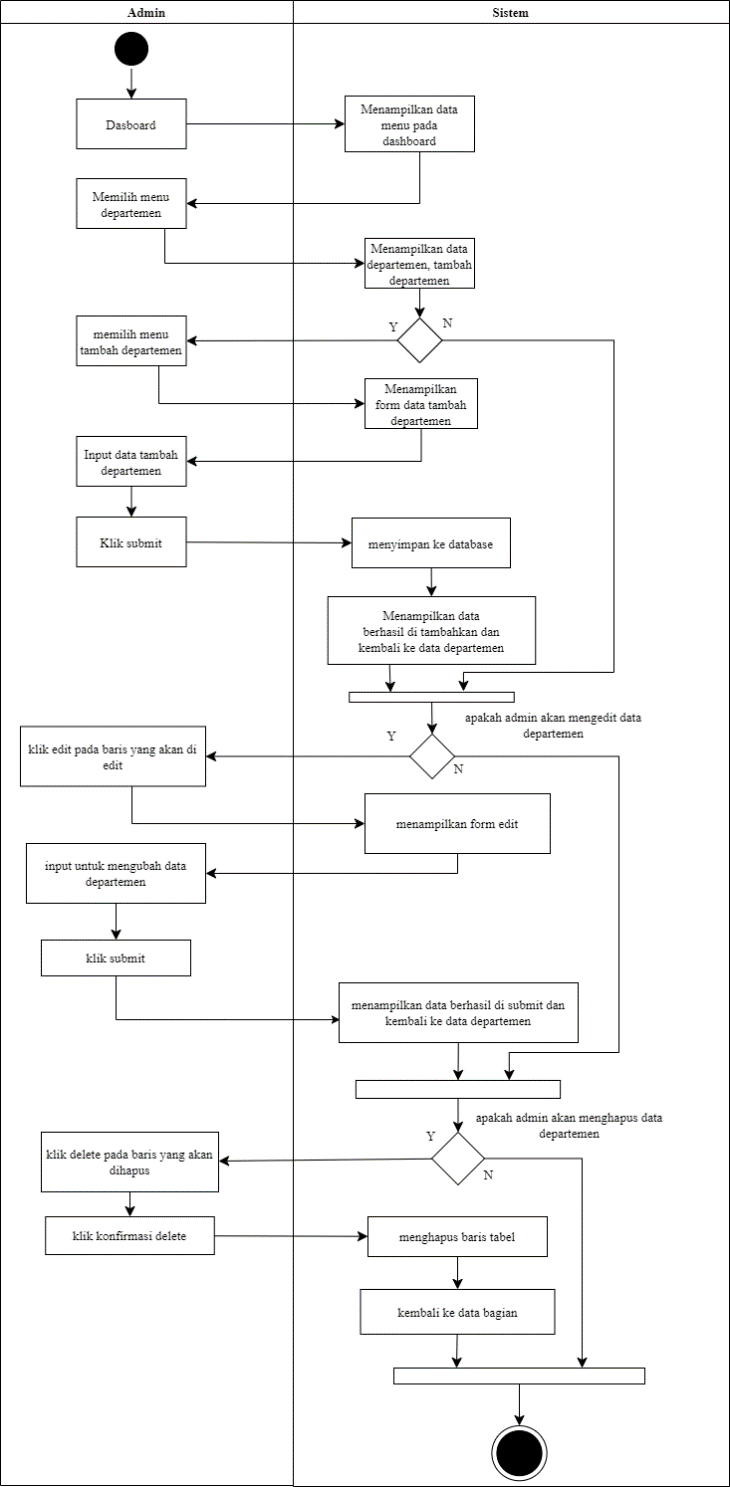
Pada gambar 4.3 menjelaskan bahwa admin telah *login* atau mengklik data karyawan. Admin dapat menambah, mengedit dan menghapus data karyawan dalam mengedit data karyawan harus meng*input*kan data yang dibutuhkan dan lengkap jika data yang dinputkan tidak lengkap maka tidak dapat dilakukan proses submit.



Gambar 4.3 Activity Diagram Karyawan

1. *Activity diagram* departemen

Pada gambar 4.4 menjelaskan bahwa jika admin mengklik menu departemen maka sistem akan menampilkan tabel departemen. Admin juga dapat melakukan tambah data dengan cara mengklik tambah departemen lalu mengisi data yang sesuai, mengubah data dengan cara mengklik pilihan edit pada tabel yang telah dipilih, serta dapat menghapus data dengan cara mengklik pilihan hapus pada tabel yg dipilih.



Gambar 4.4 Activity Diagram Departemen

1. *Activity diagram* absensi

Pada gambar 4.5 menjelaskan apabila admin mengklik menu absensi maka sistem akan menampilkan halaman data absen dan tambah absen. Dalam menu absensi juga dapat mengubah data dalam pilihan edit pada data absen, untuk menghapus data yang sudah tidak diperlukan ada pada pilihan hapus. Dalam menu absensi ini hanya untuk absensi izin keterangan.

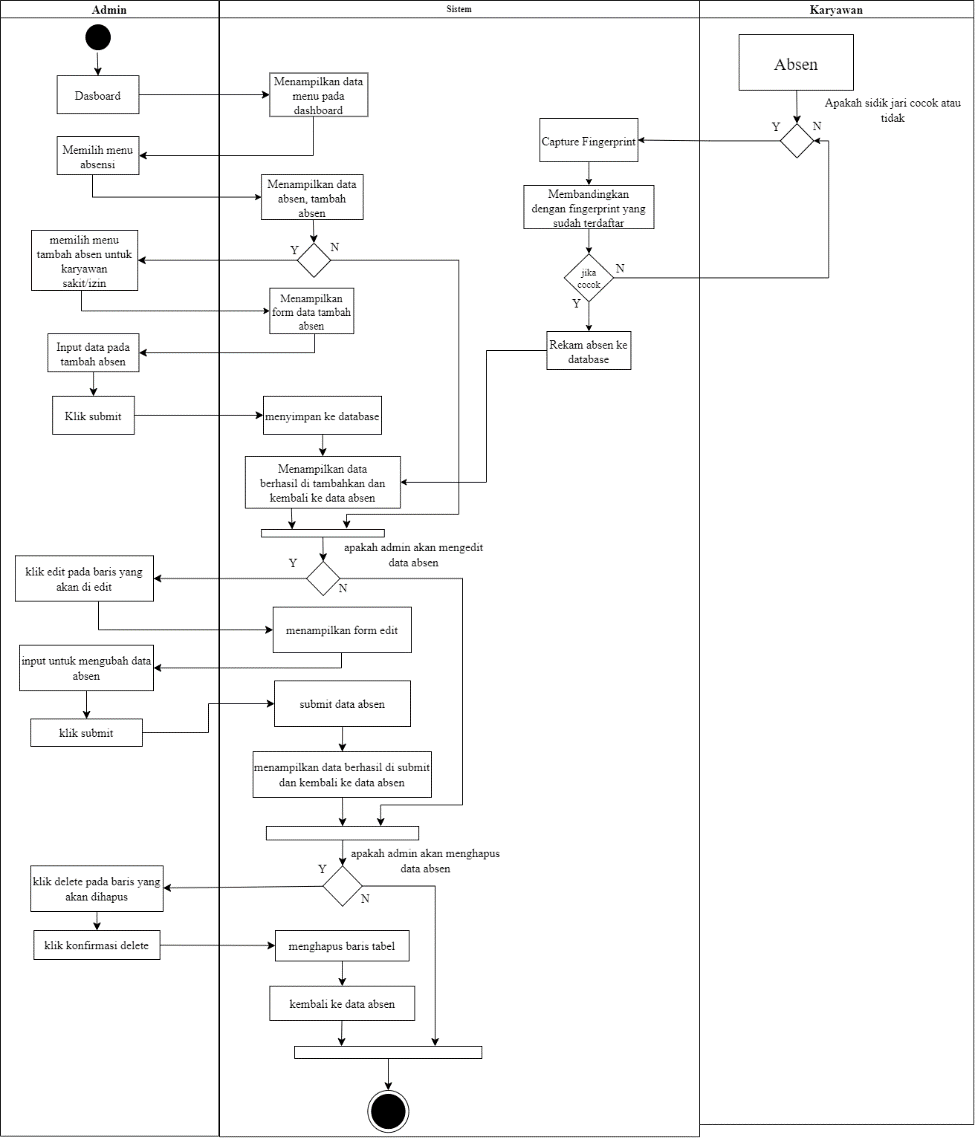
Pada tabel karyawan menggunakan metode *minutiae* yaitu pencocokan sidik jari maka proses kerja pada algoritma tersebut yaitu :

1. Pengambilan citra sidik jari, proses ini dilakukan menggunakan alat pemindai sidik jari kemudian diolah dalam bentuk digital.
2. Prosesan awal, citra sidik jari perlu diolah untuk menghilangkan noise dan mempertegas kontras.
3. Peningkatan, yaitu meningkatan kontras dan filtering untuk memperjelas detail-detail halus dalam citra.
4. Segmentasi, yaitu citra yang ditingkatkan kemudian dipecah seperti garis-garis dan ruang kosong antara garis-garis yang berbeda dalam sidik jari.
5. Pencarian *minutiae* *points*, yaitu untuk mencari titik-titik *minutiae* dalam citra. Proses ini melibatkan perubahan arah garis-garis atau titik-titik ujung garis.

* ***Ridge Ending* (Ujung Garis)**: Titik dimana garis-garis sidik jari berakhir.
* ***Bifurcation* (Persimpangan)**: titik di mana garis-garis sidik jari bercabang menjadi dua garis lain.

1. Pembersihan *minutiae,* yaitu *minutiae* yang ditemukan dapat mengandung *noise* atau *error*, pada proses ini melibatkan pembersihan hasil pencarian untuk menghapus minutiae palsu atau tidak valid.
2. Representasi dan penyimpanan, yaitu *minutiae* yang sah kemudian direpresentasikan dalam bentuk yang dapat digunakan untuk identifikasi dan perbandingan.
3. Saat melakukan identifikasi atau autentifikasi, setiap sidik jari yang diambil akan diubah menjadi representasi minutiae yang sesuai. Representasi ini kemudian dibandingkan dengan sidik jari yang sudah tersimpan dalam database.
4. Sistem akan membuat keputusan apakah sidik jari yang diuji cocok atau tidak cocok dengan yang ada dalam database.

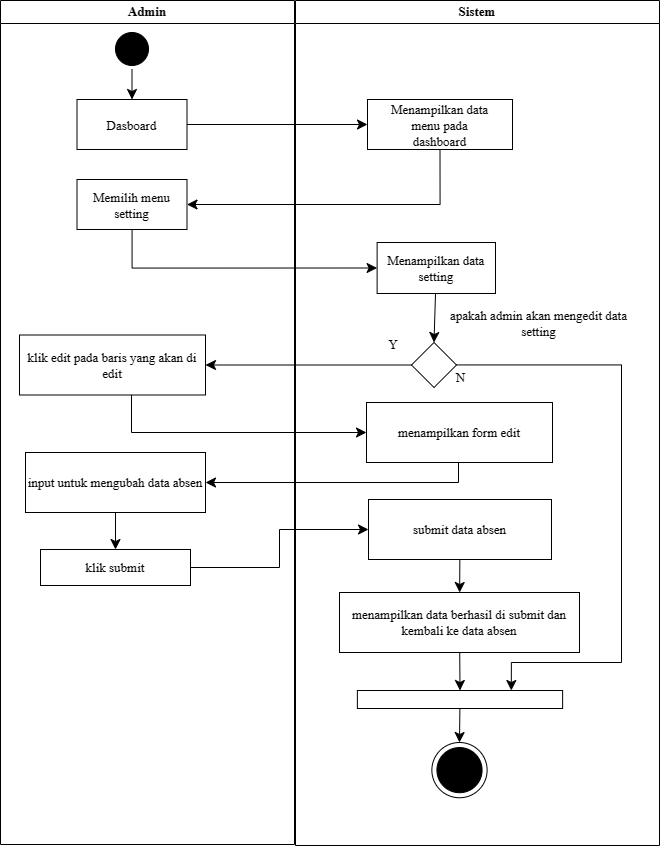
Setelah melakukan absensi maka proses selanjutnya pada sistem yaitu perbadingan seperti sidik jari cocok atau tidak cocok dengan yang ada dalam database, jika sidik jari cocok maka absensi dapat direkam pada database, jika sidik jari tidak cocok maka sistem akan kembali ke halaman absen.



Gambar 4.5 Activity Diagram Absensi

1. *Activity diagram setting*

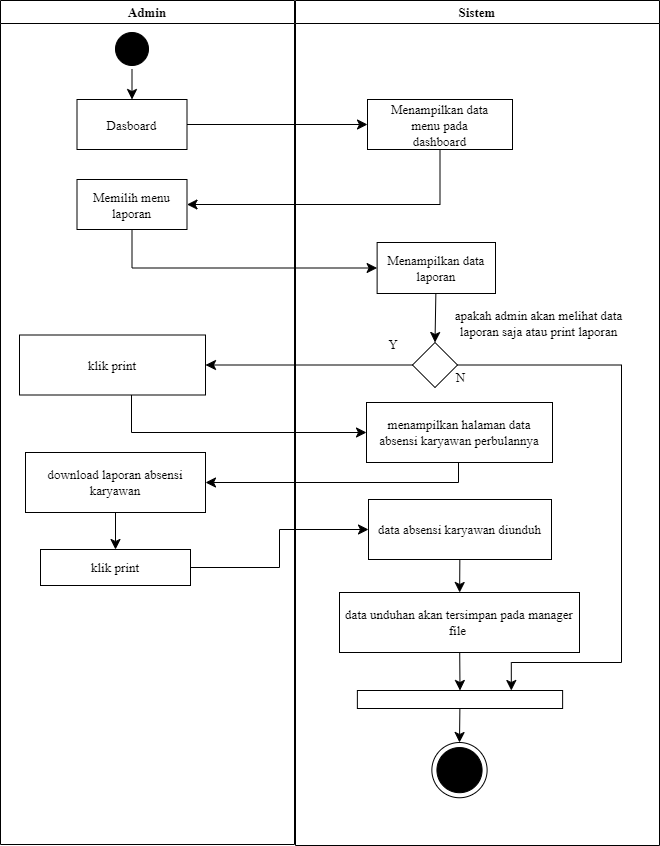
Pada gambar 4.6 menjelaskan apabila admin mengklik menu *setting* maka sistem akan menampilkan halaman *setting.* Admin juga dapat melakukan edit jam kerja pada menu *setting.*



Gambar 4.6 Activity Diagram Setting

1. *Activity diagram* laporan

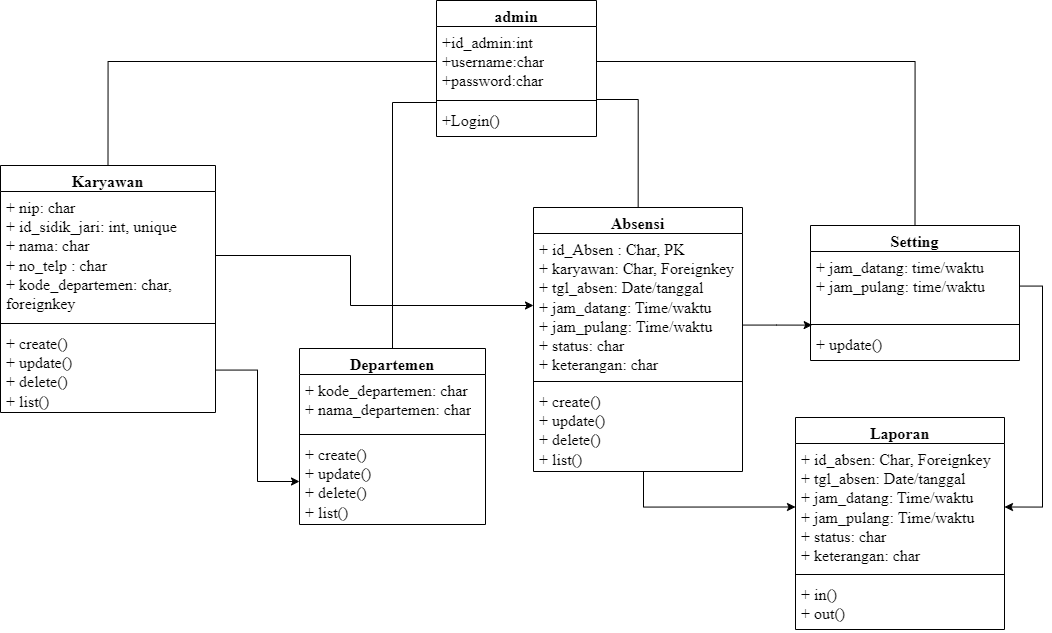
Pada gambar 4.7 menjelaskan halaman menu laporan pada aplikasi, data yang diperlukan dalam laporan yaitu meliputi nama karyawan, bulan dan tahun. Akan menghasilkan tambilan prin laporan pada aplikasi, pada print laporan tersebut hanya bisa dilihat saja atau jika perlu data laporan dapat diunduh hasil unduhan akan otomatis terunduh pada browser dan file tersebut akan muncul di manager file unduhan/*download*.



Gambar 4.7 Activity Diagram Laporan

1. *Class Diagram*

*Class diagram* adalah diagram yang menggambarkan struktur sebuah sistem dari definisi yang akan digunakan untuk membangun sistem seperti pada gambar dibawah ini:



Gambar 4.8 Class Diagram Aplikasi Absensi

* + 1. Struktur Tabel

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam aplikasi absensi menggunakan metode *miniutiae* sebagai pengukur :

1. Tabel karyawan

Tabel 4.17 Tabel Karyawan

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Keterangan** |
| Nip | Char, PK | 30 | Nomor Induk Pegawai |
| id\_sidik\_jari | Integer, unique |  | Id sidik jari |
| nama | Char | 50 | Nama Pegawai |
| no\_telp | Char | 24 | Nomor telepon pegawai |
| kode\_departemen | Char, unique | 5 | Nomor kode departemen |

1. Tabel departemen

Tabel 4.18 Tabel Departemen

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Size | Keterangan |
| kode\_departemen | Char | 5 | Kode departemen |
| nama\_departemen | Char | 30 | Nama departemen |

1. Tabel absen

Tabel 4.19 Tabel Absen

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Size | Keterangan |
| ID Absen | Char,PK |  | Nama karyawan |
| tgl\_absen | Date/tanggal |  | Tanggal absen |
| status | Char |  | Status kehadiran |
| Keterangan | Char | 300 | Keterangan absen karyawan |

1. Tabel setting

Tabel 4.20 Tabel Setting

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Size | Keterangan |
| jam\_datang | Time/waktu |  | Jam datang |
| jam\_pulang | Time/waktu |  | Jam pulang |

1. Tabel laporan

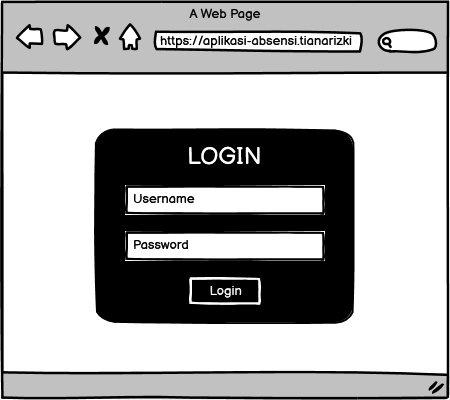
Tabel 4.21 Tabel Laporan

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Size | Keterangan |
| Id\_absen | Char, FK |  | Nomor entri absen |
| Tgl\_absen | Date/tanggal |  | Tanggal absen |
| Jam\_datang | Time/waktu |  | Jam datang |
| Jam\_pulang | Time/waktu |  | Jam pulang |
| Status | Char |  | Status kehadiran |
| Keterangan | Char | 300 | Keterangan absen karyawan |

* + 1. Desain Sistem

1. Desain Tampilan antarmuka *login*

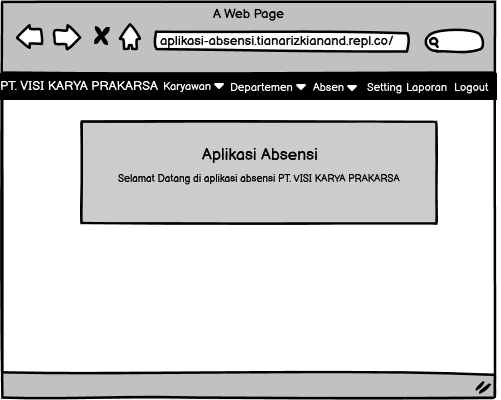
Berikut pada gambar 4.9 merupakan tampilan antarmuka login pada aplikasi absensi yang akan dirancang:



Gambar 4.9 User Interface login

1. Desain tampilan antarmuka *dashboard*

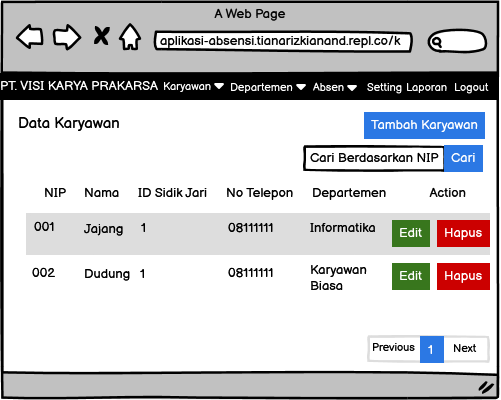
Berikut pada gambar 4.10 merupakan tampilan antarmuka *dashboard* pada aplikasi absensi yang akan dirancang:



Gambar 4.10 User Interface Dashboard

1. Desain tampilan antarmuka data karyawan

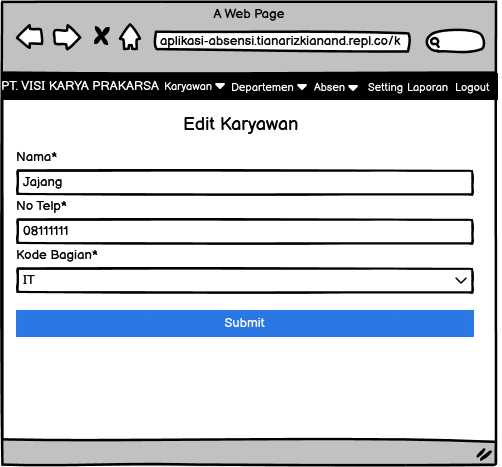
Berikut pada gambar 4.11 merupakan tampilan antarmuka data karyawan pada aplikasi absensi yang akan dirancang:



Gambar 4.11 User Interface Data Karyawan

1. Desain tampilan antarmuka edit karyawan

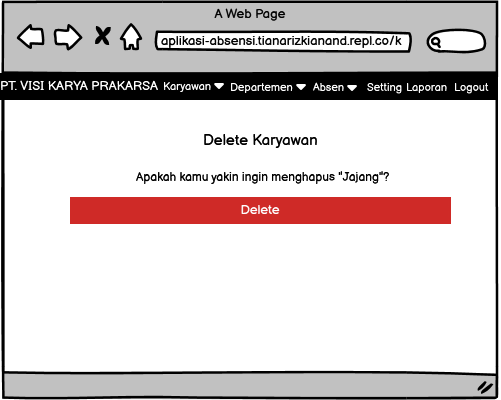
Berikut pada gambar 4.12 merupakan tampilan antarmuka edit karyawan pada aplikasi absensi yang akan dirancang:



Gambar 4.12 User Interface Edit Karyawan

1. Desain tampilan antarmuka *delete* karyawan

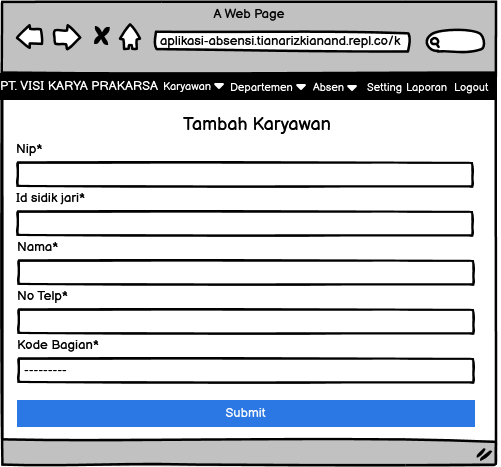
Berikut pada gambar 4.13 merupakan tampilan antarmuka *delete* karyawan pada aplikasi absensi yang akan dirancang:



Gambar 4.13 User Interface Delete Karyawan

1. Desain tampilan antarmuka tambah karyawan

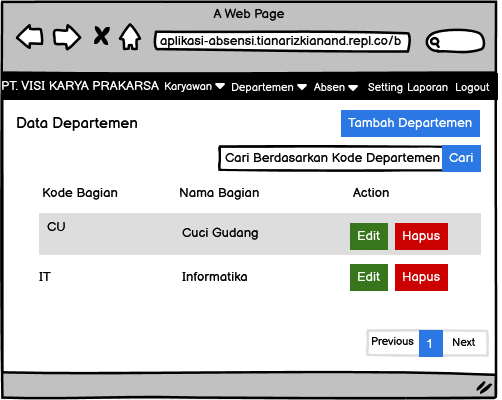
Berikut pada gambar 4.14 merupakan tampilan antarmuka tambah karyawan pada aplikasi absensi yang akan dirancang:



Gambar 4.14 User Interface Tambah Karyawan

1. Desain tampilan antarmuka data departemen

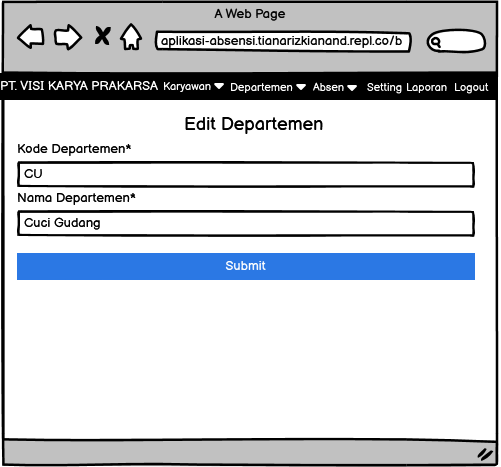
Berikut pada gambar 4.15 merupakan tampilan antarmuka data bagian pada aplikasi absensi yang akan dirancang:



Gambar 4.15 User Interface Departemen

1. Desain tampilan antarmuka edit departemen

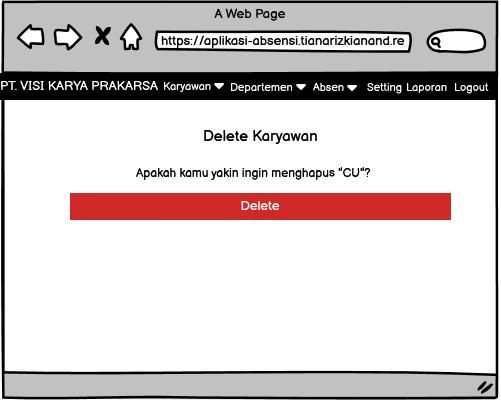
Berikut pada gambar 4.16 merupakan tampilan antarmuka edit bagian pada aplikasi absensi yang akan dirancang:



Gambar 4.16 User Interface Edit Departemen

1. Desain tampilan *delete* departemen

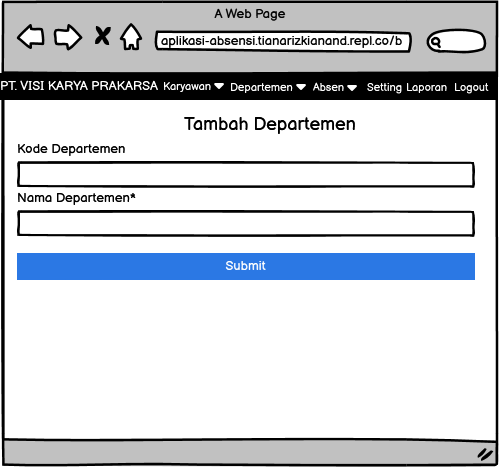
Berikut pada gambar 4.17 merupakan tampilan antarmuka *delete* bagian pada aplikasi absensi yang akan dirancang:



Gambar 4.17 User Interface Delete Departemen

1. Desain tampilan antarmuka tambah departemen

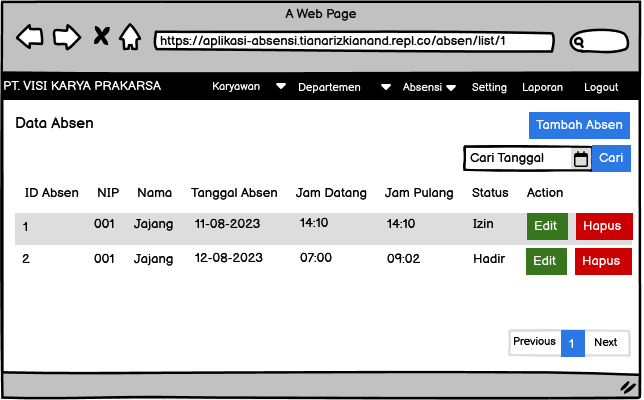
Berikut pada gambar 4.18 merupakan tampilan antarmuka tambah bagian pada aplikasi absensi yang akan dirancang:



Gambar 4.18 User Interface Tambah Departemen

1. Desain tampilan antarmuka data absensi

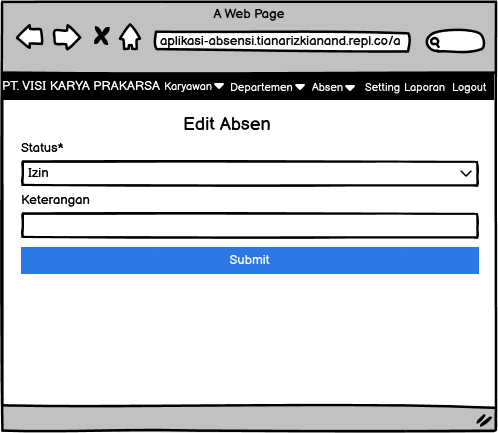
Berikut pada gambar 4.19 merupakan tampilan antarmuka data absen pada aplikasi absensi yang akan dirancang:



Gambar 4.19 User Interface Data Absen

1. Desain tampilan antarmuka edit absen

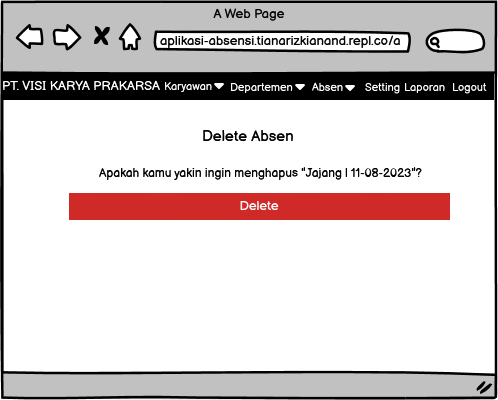
Berikut pada gambar 4.20 merupakan tampilan antarmuka edit absen pada aplikasi absensi yang akan dirancang:



Gambar 4.20 User Interface Edit Absen

1. Desain tampilan antarmuka *delete* absen

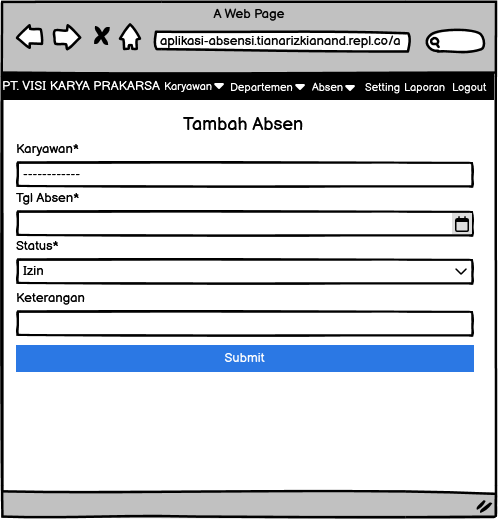
Berikut pada gambar 4.21 merupakan tampilan antarmuka *delete* absen pada aplikasi absensi yang akan dirancang:



Gambar 4.21 User Interface Delete Absen

1. Desain tampilan tambah absen

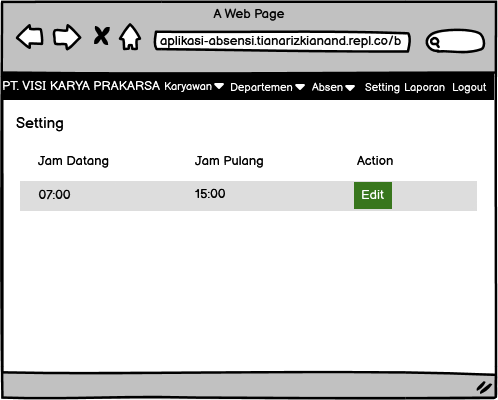
Berikut pada gambar 4.22 merupakan tampilan antarmuka tambah absen pada aplikasi absensi yang akan dirancang:



Gambar 4.22 User Interface Tambah Absen

1. Desain tampilan antarmuka *setting*

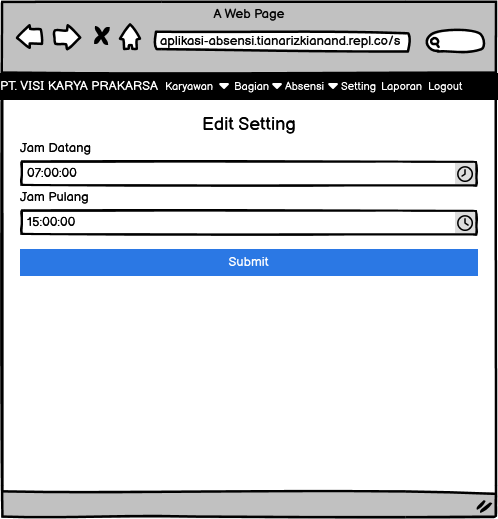
Berikut pada gambar 4.23 merupakan tampilan antarmuka *setting* pada aplikasi absensi yang akan dirancang:



Gambar 4.23 User Interface Setting

1. Desain tampilan antarmuka *edit setting*

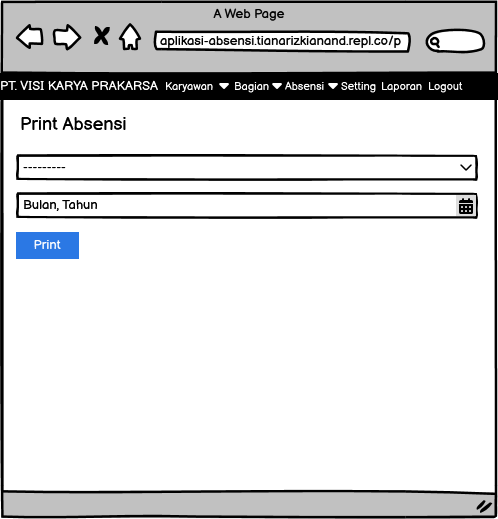
Berikut pada gambar 4.24 merupakan tampilan antarmuka *edit setting* pada aplikasi absensi yang akan dirancang:



Gambar 4.24 User Interface Edit Setting

1. Desain tampilan antarmuka laporan

Berikut pada gambar 4.25 merupakan tampilan antarmuka laporan pada aplikasi absensi yang akan dirancang:



Gambar 4.25 User Interface Laporan

# BAB V IMPLEMENTASI DAN PENGUJIAN

## 5.1 Implementasi

Setelah melaksanakan analisis dan perancangan maka selanjutnya adalah pengimplentasian untuk menjalankan analisis dan perancangan yang sudah dibuat ke dalan bentuk aplikasi.

* + 1. Listing Program

1. Listing program *dashboard*

|  |
| --- |
| {% extends './base.html' %}  {% block content %}  <div class="p-5 text-center bg-light">  <h1 class="mb-3">Aplikasi Absensi</h1>  <p class="text-center">  Selamat Datang di aplikasi absensi PT. VISI KARYA PRAKARSA  </p>  </div>  {% endblock %} |

1. Listing program karyawan

|  |
| --- |
| {% extends './base.html' %}  {% block content %}  <div class="container">  <div class="d-flex justify-content-between align-items-center mb-5">  <h3>Data Karyawan</h3>  <a href="{% url 'karyawancreate' %}" class="btn btn-primary" role="button">Tambah Karyawan</a>  </div>  <div class="row">  <form class="d-flex justify-content-end align-items-center mb-3" method="POST" action="{% url 'karyawansearch' 1 %}">  {% csrf\_token %}  <di>  <input  class="form-control me-2"  name="search\_query"  type="search"  placeholder="Cari Berdasarkan NIP"  aria-label="Search"  >  </di>  <button class="btn btn-primary" type="submit">Cari</button>  </form>  </div>  <div class="row">  <div class="col">  <table class="table table-striped">  <thead>  <tr>  <th scope="col">NIP</th>  <th scope="col">Nama</th>  <th scope="col">ID Sidik Jari</th>  <th scope="col">No Telepon</th>  <th scope="col">Bagian</th>  <th scope="col">Action</th>  </tr>  </thead>  <tbody>  {% for karyawan in page\_obj %}  <tr>  <td>{{ karyawan.nip }}</td>  <td>{{ karyawan.nama }}</td>  <td>{{ karyawan.id\_sidik\_jari }}</td>  <td>{{ karyawan.no\_telp }}</td>  <td>{{ karyawan.kode\_bagian.nama\_bagian }}</td>  <td>  <a href="{% url 'karyawanupdate' karyawan.nip %}" class="btn btn-success" role="button">Edit</a> </a>  <a href="{% url 'karyawandelete' karyawan.nip %}" class="btn btn-danger" role="button">Hapus</a>  </td>  </tr>  {% endfor %}  </tbody>  </table>  </div>  </div>  <div class="d-flex justify-content-end align-items-center mt-5">  <nav aria-label="Page navigation example">  <ul class="pagination">  {% if page\_obj.has\_previous %}  <li class="page-item">  <a class="page-link ac" href="{% url 'karyawanlist-by-page' page\_obj.previous\_page\_number %}">Previous</a>  </li>  {% else %}  <li class="page-item disabled">  <a class="page-link">Previous</a>  </li>  {% endif %}    {% for page\_number in page\_obj.adjusted\_elided\_pages %}  {% if page\_number == page\_obj.paginator.ELLIPSIS %}  <li class="page-item">  <a class="page-link">{{ page\_number }}</a>  </li>  {% else %}  <li class="page-item {% if page\_number == page\_obj.number %}active{% endif %}">  <a class="page-link" href="{% url 'karyawanlist-by-page' page\_number %}">{{ page\_number }}</a>  </li>  {% endif %}  {% endfor %}    {% if page\_obj.has\_next %}  <li class="page-item">  <a class="page-link" href="{% url 'karyawanlist-by-page' page\_obj.next\_page\_number %}">Next</a>  </li>  {% else %}  <li class="page-item disabled">  <a class="page-link">Next</a>  </li>  {% endif %}  </ul>  </nav>  </div>  </div>  {% endblock %} |

1. Listing program edit karyawan

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Edit Karyawan</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  {% csrf\_token %}  {{ form|crispy }}  <div class="row align-items-center mt-3">  <input class="col btn btn-primary" type="submit" />  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program delete karyawan

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Delete Karyawan</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  <div class="row align-items-center mt-3">  {% csrf\_token %}  <h4 class="text-center">Apakah kamu yakin ingin menghapus "{{ object }}"?</h4>  <input class="col btn btn-danger mt-3" type="Submit" value="Delete"/>  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program tambah karyawan

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Tambah Karyawan</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  {% csrf\_token %}  {{ form|crispy }}  <div class="row align-items-center mt-3">  <input class="col btn btn-primary" type="submit" />  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program bagian

|  |
| --- |
| {% extends './base.html' %}  {% block content %}  <div class="container">  <div class="d-flex justify-content-between align-items-center mb-5">  <h3>Data Bagian</h3>  <a href="{% url 'bagiancreate' %}" class="btn btn-primary" role="button">Tambah Bagian</a>  </div>  <div class="row">  <form class="d-flex justify-content-end align-items-center mb-3" method="POST" action="{% url 'bagiansearch' 1 %}">  {% csrf\_token %}  <div>  <input  class="form-control me-2"  name="search\_query"  type="search"  placeholder="Cari Berdasarkan Kode Bagian"  aria-label="Search"  >  </div>  <button class="btn btn-primary" type="submit">Cari</button>  </form>  </div>  <div class="row">  <div class="col">  <table class="table table-striped">  <thead>  <tr>  <th scope="col">Kode Bagian</th>  <th scope="col">Nama Bagian</th>  <th scope="col">Action</th>  </tr>  </thead>  <tbody>  {% for bagian in page\_obj %}  <tr>  <td>{{ bagian.kode\_bagian }}</td>  <td>{{ bagian.nama\_bagian }}</td>  <td>  <a href="{% url 'bagianupdate' bagian.id %}" class="btn btn-success" role="button">Edit</a> </a>  <a href="{% url 'bagiandelete' bagian.id %}" class="btn btn-danger" role="button">Hapus</a>  </td>  </tr>  {% endfor %}  </tbody>  </table>  </div>  </div>  <div class="d-flex justify-content-end align-items-center mt-5">  <nav aria-label="Page navigation example">  <ul class="pagination">  {% if page\_obj.has\_previous %}  <li class="page-item">  <a class="page-link ac" href="{% url 'bagian-by-page' page\_obj.previous\_page\_number %}">Previous</a>  </li>  {% else %}  <li class="page-item disabled">  <a class="page-link">Previous</a>  </li>  {% endif %}    {% for page\_number in page\_obj.adjusted\_elided\_pages %}  {% if page\_number == page\_obj.paginator.ELLIPSIS %}  <li class="page-item">  <a class="page-link">{{ page\_number }}</a>  </li>  {% else %}  <li class="page-item {% if page\_number == page\_obj.number %}active{% endif %}">  <a class="page-link" href="{% url 'bagianlist-by-page' page\_number %}">{{ page\_number }}</a>  </li>  {% endif %}  {% endfor %}    {% if page\_obj.has\_next %}  <li class="page-item">  <a class="page-link" href="{% url 'karyawanlist-by-page' page\_obj.next\_page\_number %}">Next</a>  </li>  {% else %}  <li class="page-item disabled">  <a class="page-link">Next</a>  </li>  {% endif %}  </ul>  </nav>  </div>  </div>  {% endblock %} |

1. Listing program edit bagian

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Edit Bagian</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  {% csrf\_token %}  {{ form|crispy }}  <div class="row align-items-center mt-3">  <input class="col btn btn-primary" type="submit" />  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program delete bagian

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Delete Bagian</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  <div class="row align-items-center mt-3">  {% csrf\_token %}  <h4 class="text-center">Apakah kamu yakin ingin menghapus "{{ object }}"?</h4>  <input class="col btn btn-danger mt-3" type="Submit" value="Delete"/>  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program tambah bagian

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Tambah Bagian</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  {% csrf\_token %}  {{ form|crispy }}  <div class="row align-items-center mt-3">  <input class="col btn btn-primary" type="submit" />  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program absensi

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block extra\_css %}  {{ form.media.css }}  {% endblock %}  {% block extra\_js %}  {{ form.media.js }}  {% endblock %}  {% block content %}  <div class="container">  <div class="d-flex justify-content-between align-items-center mb-5">  <h3>Data Absen</h3>  <a href="{% url 'absencreate' %}" class="btn btn-primary" role="button">Tambah Absen</a>  </div>  <div class="row">  <form class="d-flex justify-content-end align-items-baseline" method="POST" action="{% url 'absensearch' 'filter' 1 %}">  {% csrf\_token %}  <div>  {{ form | crispy }}  </div>  <button class="btn btn-primary" type="submit">Cari</button>  </form>  </div>  <div class="row">  <div class="col">  <table class="table table-striped">  <thead>  <tr>  <th scope="col">NIP</th>  <th scope="col">Nama</th>  <th scope="col">Tanggal Absen</th>  <th scope="col">Jam Datang</th>  <th scope="col">Jam Pulang</th>  <th scope="col">Status</th>  <th scope="col">Action</th>  </tr>  </thead>  <tbody>  {% for absen in page\_obj %}  <tr>  <td>{{ absen.karyawan.nip }}</td>  <td>{{ absen.karyawan.nama }}</td>  <td>{{ absen.tgl\_absen|date:"d-m-Y" }}</td>  <td>{{ absen.jam\_datang|date:"H:i" }}</td>  <td>{{ absen.jam\_pulang|date:"H:i" }}</td>  <td>{{ absen.status }}</td>  <td>  <a href="{% url 'absennupdate' absen.id %}" class="btn btn-success" role="button">Edit</a>  <a href="{% url 'absendelete' absen.id %}" class="btn btn-danger" role="button">Hapus</a>  </td>  </tr>  {% endfor %}  </tbody>  </table>  </div>  </div>  <div class="d-flex justify-content-end align-items-center mt-5">  <nav aria-label="Page navigation example">  <ul class="pagination">  {% if page\_obj.has\_previous %}  <li class="page-item">  <a class="page-link ac" href="{% url 'absenlist-by-page' page\_obj.previous\_page\_number %}">Previous</a>  </li>  {% else %}  <li class="page-item disabled">  <a class="page-link">Previous</a>  </li>  {% endif %}    {% for page\_number in page\_obj.adjusted\_elided\_pages %}  {% if page\_number == page\_obj.paginator.ELLIPSIS %}  <li class="page-item">  <a class="page-link">{{ page\_number }}</a>  </li>  {% else %}  <li class="page-item {% if page\_number == page\_obj.number %}active{% endif %}">  <a class="page-link" href="{% url 'absenlist-by-page' page\_number %}">{{ page\_number }}</a>  </li>  {% endif %}  {% endfor %}    {% if page\_obj.has\_next %}  <li class="page-item">  <a class="page-link" href="{% url 'absenlist-by-page' page\_obj.next\_page\_number %}">Next</a>  </li>  {% else %}  <li class="page-item disabled">  <a class="page-link">Next</a>  </li>  {% endif %}  </ul>  </nav>  </div>  </div>  {% endblock %} |

1. Listing program edit absen

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block extra\_css %}  {{ form.media.css }}  {% endblock %}  {% block extra\_js %}  {{ form.media.js }}  {% endblock %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Edit Absen</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  {% csrf\_token %}  {{ form | crispy }}  <div class="row align-items-center mt-3">  <input class="col btn btn-primary" type="submit" />  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program delete absen

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Delete Absen</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  <div class="row align-items-center mt-3">  {% csrf\_token %}  <h4 class="text-center">Apakah kamu yakin ingin menghapus "{{ object }}"?</h4>  <input class="col btn btn-danger mt-3" type="Submit" value="Delete"/>  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program tambah absen

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block extra\_css %}  {{ form.media.css }}  {% endblock %}  {% block extra\_js %}  {{ form.media.js }}  {% endblock %}  {% block content %}  <div class="container">  <div class="d-flex justify-content-between align-items-center mb-5">  <h3>Data Absen</h3>  <a href="{% url 'absencreate' %}" class="btn btn-primary" role="button">Tambah Absen</a>  </div>  <div class="row">  <form class="d-flex justify-content-end align-items-baseline" method="POST" action="{% url 'absensearch' 'filter' 1 %}">  {% csrf\_token %}  <div>  {{ form | crispy }}  </div>  <button class="btn btn-primary" type="submit">Cari</button>  </form>  </div>  <div class="row">  <div class="col">  <table class="table table-striped">  <thead>  <tr>  <th scope="col">NIP</th>  <th scope="col">Nama</th>  <th scope="col">Tanggal Absen</th>  <th scope="col">Jam Datang</th>  <th scope="col">Jam Pulang</th>  <th scope="col">Status</th>  <th scope="col">Action</th>  </tr>  </thead>  <tbody>  {% for absen in page\_obj %}  <tr>  <td>{{ absen.karyawan.nip }}</td>  <td>{{ absen.karyawan.nama }}</td>  <td>{{ absen.tgl\_absen|date:"d-m-Y" }}</td>  <td>{{ absen.jam\_datang|date:"H:i" }}</td>  <td>{{ absen.jam\_pulang|date:"H:i" }}</td>  <td>{{ absen.status }}</td>  <td>  <a href="{% url 'absennupdate' absen.id %}" class="btn btn-success" role="button">Edit</a>  <a href="{% url 'absendelete' absen.id %}" class="btn btn-danger" role="button">Hapus</a>  </td>  </tr>  {% endfor %}  </tbody>  </table>  </div>  </div>  <div class="d-flex justify-content-end align-items-center mt-5">  <nav aria-label="Page navigation example">  <ul class="pagination">  {% if page\_obj.has\_previous %}  <li class="page-item">  <a class="page-link ac" href="{% url 'absenlist-by-page' page\_obj.previous\_page\_number %}">Previous</a>  </li>  {% else %}  <li class="page-item disabled">  <a class="page-link">Previous</a>  </li>  {% endif %}    {% for page\_number in page\_obj.adjusted\_elided\_pages %}  {% if page\_number == page\_obj.paginator.ELLIPSIS %}  <li class="page-item">  <a class="page-link">{{ page\_number }}</a>  </li>  {% else %}  <li class="page-item {% if page\_number == page\_obj.number %}active{% endif %}">  <a class="page-link" href="{% url 'absenlist-by-page' page\_number %}">{{ page\_number }}</a>  </li>  {% endif %}  {% endfor %}    {% if page\_obj.has\_next %}  <li class="page-item">  <a class="page-link" href="{% url 'absenlist-by-page' page\_obj.next\_page\_number %}">Next</a>  </li>  {% else %}  <li class="page-item disabled">  <a class="page-link">Next</a>  </li>  {% endif %}  </ul>  </nav>  </div>  </div>  {% endblock %} |

1. Listing program setting

|  |
| --- |
| {% extends './base.html' %}  {% block content %}  <div class="container">  <div class="d-flex justify-content-between align-items-center mb-5">  <h3>Setting</h3>  </div>  <div class="row">  <div class="col">  <table class="table table-striped">  <thead>  <tr>  <th scope="col">Jam Datang</th>  <th scope="col">Jam Pulang</th>  <th scope="col">Action</th>  </tr>  </thead>  <tbody>  {% for s in setting %}  <tr>  <td>{{ s.jam\_datang|date:"H:i" }}</td>  <td>{{ s.jam\_pulang|date:"H:i" }}</td>  <td>  <a href="{% url 'settingupdate' s.id %}" class="btn btn-success" role="button">Edit</a>  </td>  </tr>  {% endfor %}  </tbody>  </table>  </div>  </div>  </div>  {% endblock %} |

1. Listing program edit setting

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block extra\_css %}  {{ form.media.css }}  {% endblock %}  {% block extra\_js %}  {{ form.media.js }}  {% endblock %}  {% block content %}  <div class="container">  <div class="row align-items-center">  <h3 class="col text-center mb-5">Edit Setting</h3>  </div>  <div>  <form class="row justify-content-md-center" action="" method="POST">  {% csrf\_token %}  {{ form | crispy }}  <div class="row align-items-center mt-3">  <input class="col btn btn-primary" type="submit" />  </div>  </form>  </div>  </div>  {% endblock %} |

1. Listing program laporan

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block extra\_css %}  {{ form.media.css }}  {% endblock %}  {% block extra\_js %}  {{ form.media.js }}  {% endblock %}  {% block content %}  <div class="container">  <div class="d-flex justify-content-between align-items-center mb-5">  <h3>Print Absensi</h3>  </div>  <div class="row">  <form class="container" method="POST" action="{% url 'print-absen' %}">  {% csrf\_token %}  <div>  {{ form | crispy }}  </div>  <button class="btn btn-primary" type="submit">Print</button>  </form>  </div>  </div>  {% endblock %} |

1. Listing laporan

|  |
| --- |
| {% extends './base.html' %}  {% load crispy\_forms\_tags %}  {% block extra\_css %}  {{ form.media.css }}  {% endblock %}  {% block extra\_js %}  {{ form.media.js }}  {% endblock %}  {% block content %}  <div class="container">  <div class="d-flex justify-content-between align-items-center mb-5">  <h3>Print Absensi</h3>  </div>  <div class="row">  <form class="container" method="POST" action="{% url 'print-absen' %}">  {% csrf\_token %}  <div>  {{ form | crispy }}  </div>  <button class="btn btn-primary" type="submit">Print</button>  </form>  </div>  </div>  {% endblock %} |

5.1.2 Implementasi Sistem

Implementasi sistem adalah tahap penerapan sistem yang akan dilakukan jika sistem telah disetujui termasuk program yang telah dibuat pada tahap perancangan sistem agar siap untuk dioperasikan. Adapun waktu dan tempat penerapan sistem yang sudah dibuat sebagai berikut :

1. Waktu dan Tempat Implementasi

|  |  |
| --- | --- |
| Tempat : | PT. Visi Karya Prakarsa |
| Alamat : | Sekejati, Kec. Buah Batu, Kota Bandung, Jawa Barat, di Jl. Jupiter Barat 36 No.37. |
| Waktu : | Bulan Mei |

5.1.3 Spesifikasi Sistem

Spesifikasi sistem menjelaskan tentang spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam perancangan aplikasi absensi *fingerprint*.

1. Spesifikasi perangkat keras

Di bawah ini merupakan spesifikasi perangkat keras yang digunakan sebagai berikut:

Tabel 5.1 Spesifikasi perangkat keras

|  |  |
| --- | --- |
| Prosesor | AMD Ryzen 5 5500U with Radeon Graphics, 2100 Mhz, 6 Core(s), 12 Logical Processor(s) |
| RAM | 8 GB |
| SSD | 250 GB |

1. Spesifikasi perangkat lunak

Di bawah ini merupakan spesifikasi perangkat keras yang digunakan sebagai berikut:

Tabel 5.2 Spesifikasi perangkat lunak

|  |  |
| --- | --- |
| Sistem Operasi | Windows 11 |
| Server | Replit |
| Aplikasi Pembuatan | Replit |
| Browser | Microsoft Chrome |
| Bahasa Pemograman | Phyton versi 3.10.11 |
| Framework back-end | Django versi 3.2.13 |
| Framework front-end | Bootstrap versi 5.0 |

5.1.4 Instalasi Sistem

Instalasi sistem ini memuat penjelasan mengenai langkah-langkah yang harus dilakukan untuk instalasi aplikasi dan instalasi *database.*

1. Instalasi Aplikasi
   * + 1. Memulai project dengan mengetik “django-admin startproject <nama-project>
       2. Lakukan migrate dengan cara masuk ke folder <nama-project> dan ketikan di terminal “phyton3 manage.py migrate”
       3. Program django sudah bisa dijalankan dengan mengetik “phyton3 manage.py runserver” di terminal.
       4. Buat app project dengan mengetik “python3 manage.py startapp <nama-app>”
       5. Masukkan app tadi yang sudah di buat ke INSTALLED\_APP di forlder setting.py
       6. Buat superuser dengan mengetik “python3 manage.py createsuperuser” di terminal
       7. Masukkan *username*, email dan *password*
       8. Buat model *database* di file models.py
       9. *Apply* *database* dengan mengetik “pyhton3 manage.py makemigrations” di terminal
       10. Kemudian ketik “python3 manage.py migrate”
       11. *Register* *models* ke admin *site*
       12. Tambahkan beberapa *library* yang akan kita gunakan di django-project/settings.py INSTALLED\_APP list di antaranya :

‘bootstrap5’,

‘crispy\_forms’,

‘django\_bootstrap5’,

‘crispy\_bootstrap5’,

‘bootstrap\_datepicker\_plus’,

‘rest\_framework’,

* + - 1. Buat *template*, *views* dan urls
      2. Aplikasi sudah bisa di gunakan

1. Browser (Google Chrome)

Browser ini berfungsi untuk mengaktifkan dan menjalankan aplikasi absensi yang sudah dibuat. Berikut ini merupakan tahapan-tahapan instalasi Browser Google Chrome, seperti di bawah ini :

* 1. Unduh *installer* Google Chrome.
  2. Jika diminta, klik jalankan atau simpan.
  3. Jika memilih simpan maka lakukan klik dua kali pada tombol *download* untuk memulai proses instalasi.
  4. Tunggu hingga proses instalasi selesai.
  5. Setelah instalasi selesai, Google Chrome sudah bisa digunakan.

1. Instalasi *Database*

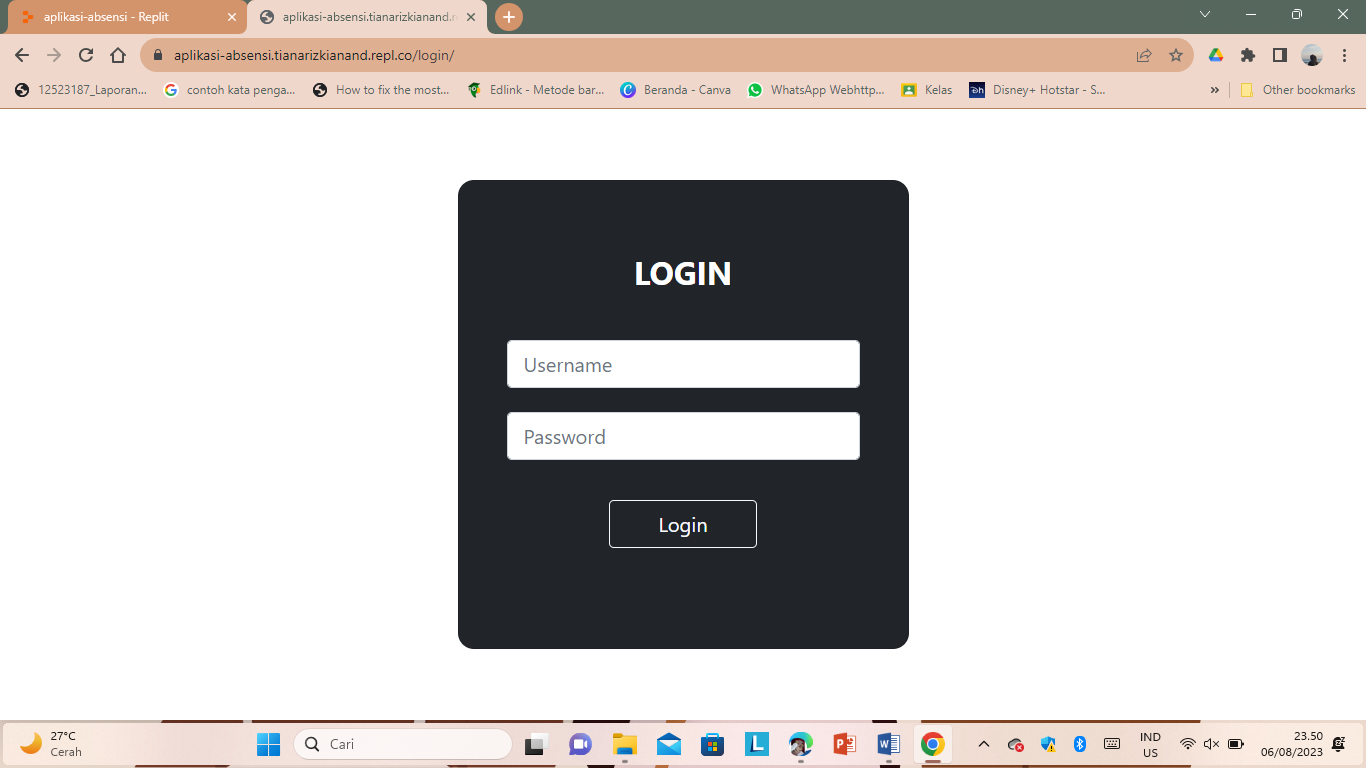
Berkaitan dengan *database* yang digunakan pada aplikasi absensi yaitu SQLite, SQLite sudah otomatis di dalam aplikasi django maka harus dilakukan yaitu mengaktifkan django agar dapat menjalankan database dan memanggil db browser for SQLite.

5.1.5 Menjalankan Sistem

Pada bagian ini akan dijelaskan bagaimana cara-cara menjalankan sistem aplikasi absensi.

1. Halaman *login*

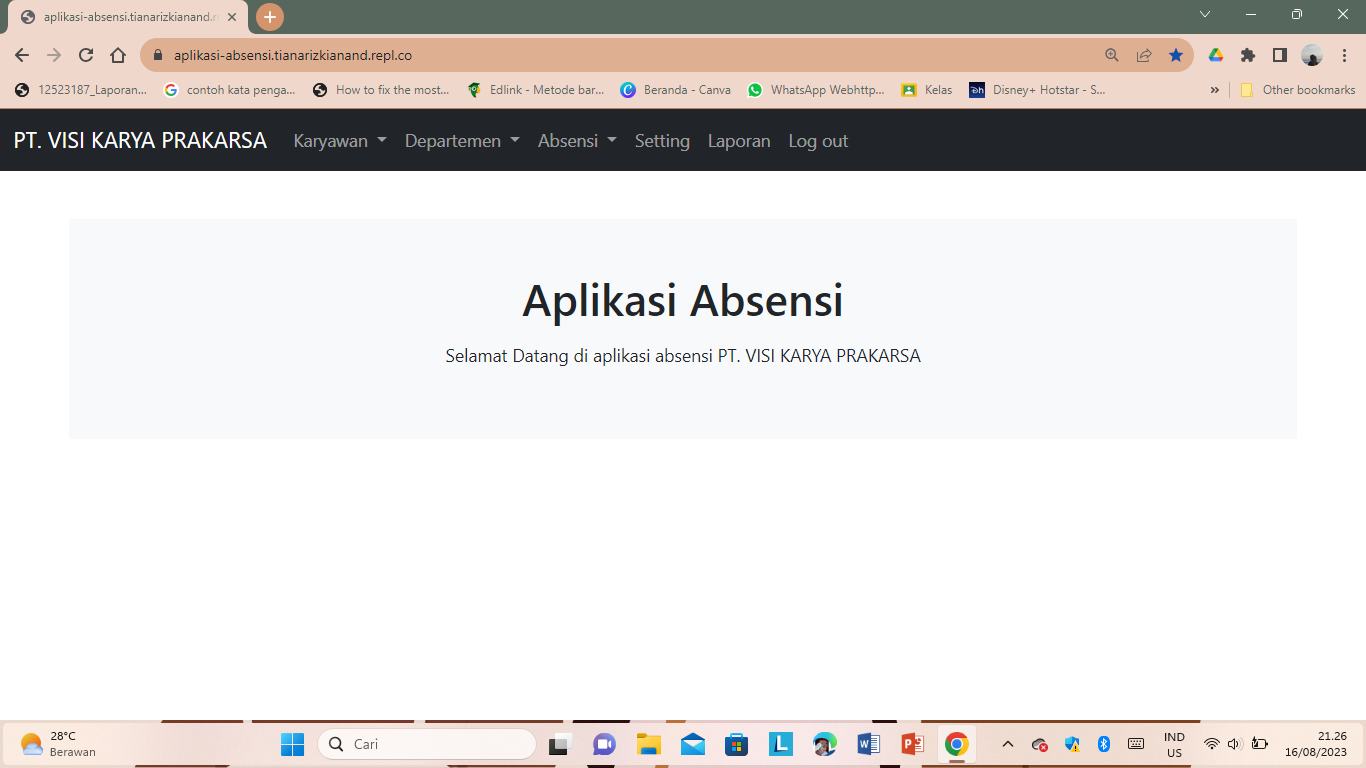
Sebelum memasuki menu pada aplikasi absensi admin harus melakukan *login* dengan memasukan *username* dan *password* aplikasi. Halaman *login* yang diperlukan untuk dapat mengakses nya yaitu menggunakan *username* dan *password* untuk dapat melihat menu-menu pada aplikasi, setelah melakukan *login* maka akan memunculkan halaman *dashboard* dari aplikasi seperti pada gambar di bawah ini :



Gambar 5.1 Halaman login

1. Halaman *Dashboard*

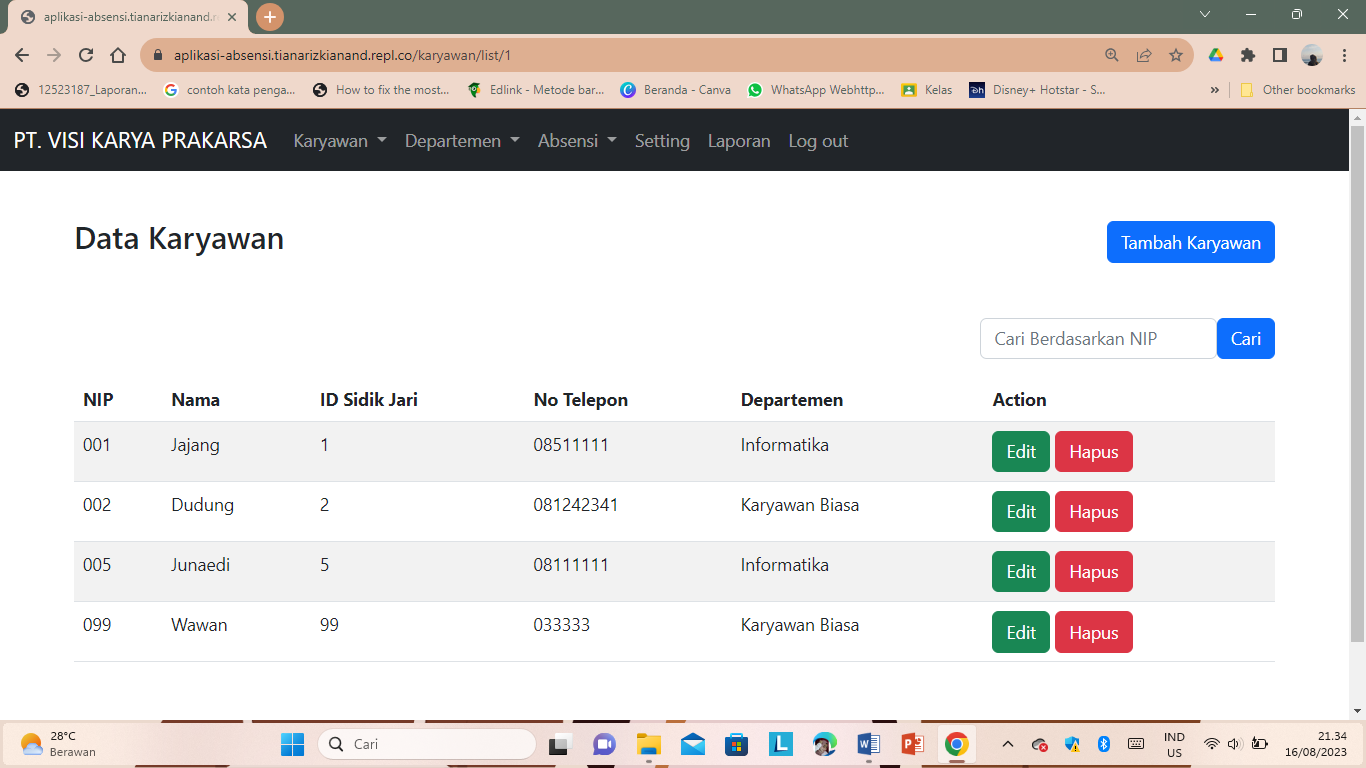
Setelah admin memasuki aplikasi maka tampilan yang pertama kali muncul yaitu halaman *dashboard* sebagai halaman utama, pada halaman *dashboard* aplikasi absensi memuat data karyawan, departemen, absensi, *setting, laporan* dan *log out.* Pada gambar 5.3menampilkan halaman dashboard yang merupakan tampilan awal ketika sudah melakukan *login* pada aplikasi



Gambar 5.2 Halaman Dashboard

1. Halaman karyawan

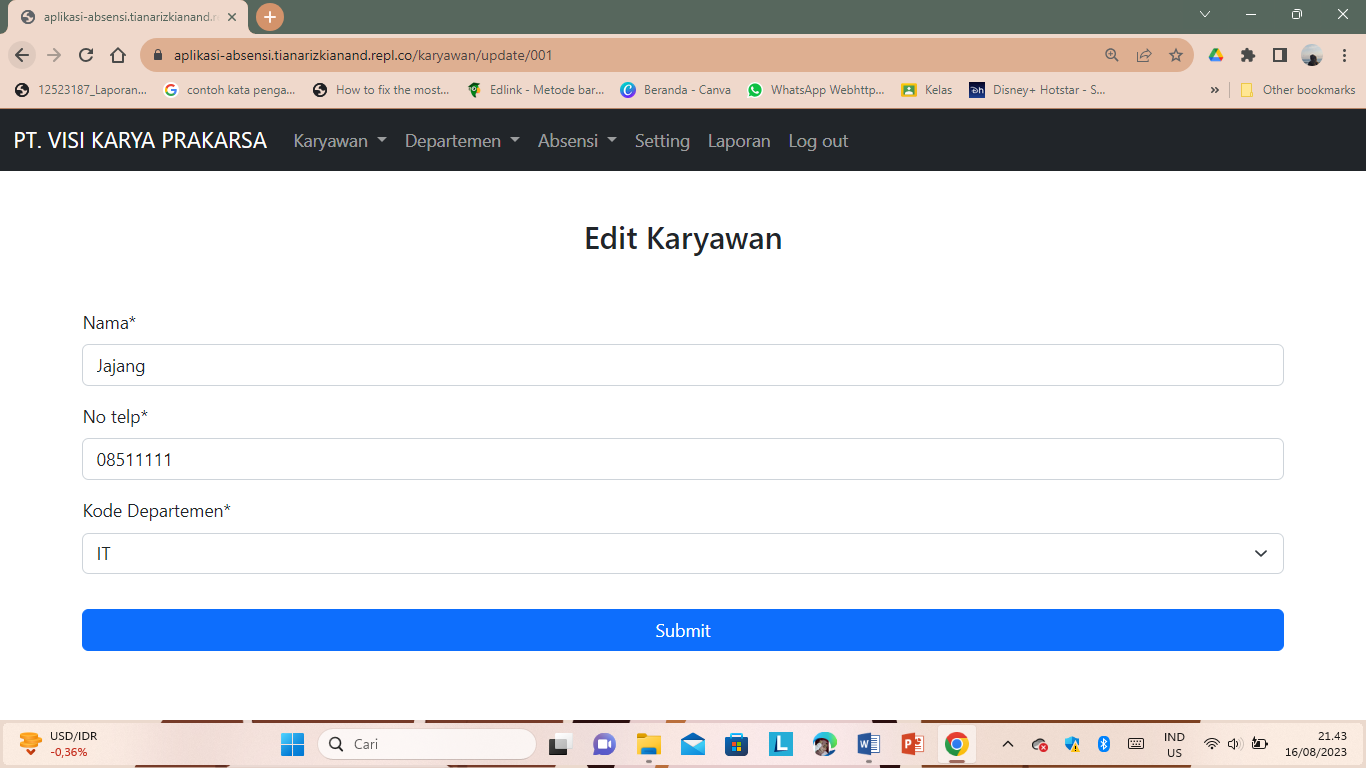
Selain pada halaman *dashboard*  pada halaman karyawan aplikasi absensi ini memuat data karyawan yang dimana data bisa di edit maupun di delete sekalipun, yang berfungsi untuk merekap data diri karyawan pada PT. Visi Karya Prakarsa, edit pada *action* untuk mengubah data yang sudah diinputkan, *delete* untuk menghapus data yang sudah tidak diperlukan sistem akan memunculkan *form.* Pada halaman menu karyawan akan menampilkan beberapa data nama karyawan dengan nip, id sidik jari, no telepon, departemen dan aksi data-data diatas muncul ketika admin sudah menginputkan data yang lengkap.seperti pada gambar di bawah ini :



Gambar 5.3 Halaman Karyawan

1. Halaman edit karyawan

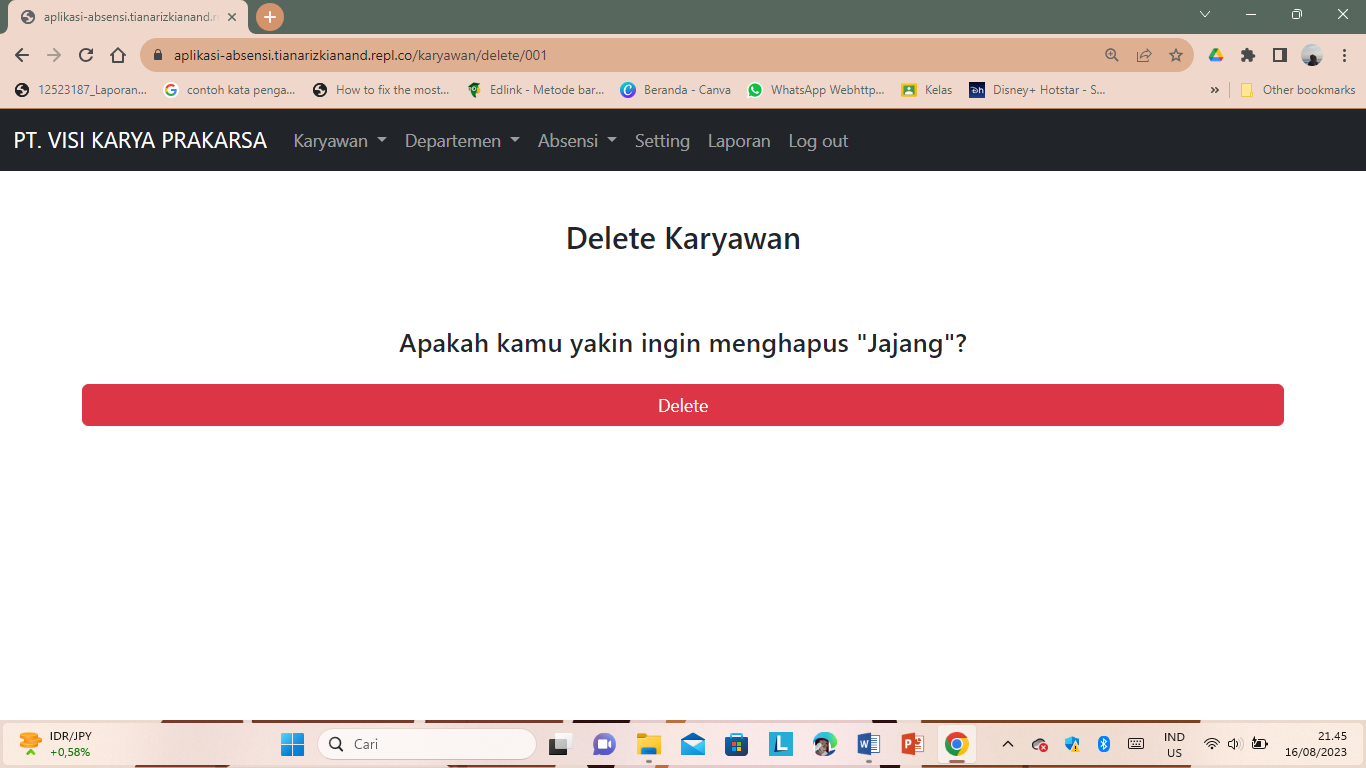
Pada gambar 5.4 admin dapat mengubah data karyawan bila terjadi kesalahan dalam mengisi data ataupun mengupdate data diri karyawan pada PT. Visi Karya Prakarsa, gambar 5.4 menunjukan halaman edit karyawan yang dimana admin hanya klik data yang akan diedit bila terjadi kesalahan dalam nama karyawan atau no telepon karyawan atapun kode pada departemen pada karyawan dan kode bagian harus disesuaikan dengan data yang sudah tertera maka proses edit karyawan baru bisa dilakukan.seperti pada gambar di bawah ini :



Gambar 5.4 Halaman Edit Karyawan

1. Halaman *delete* karyawan

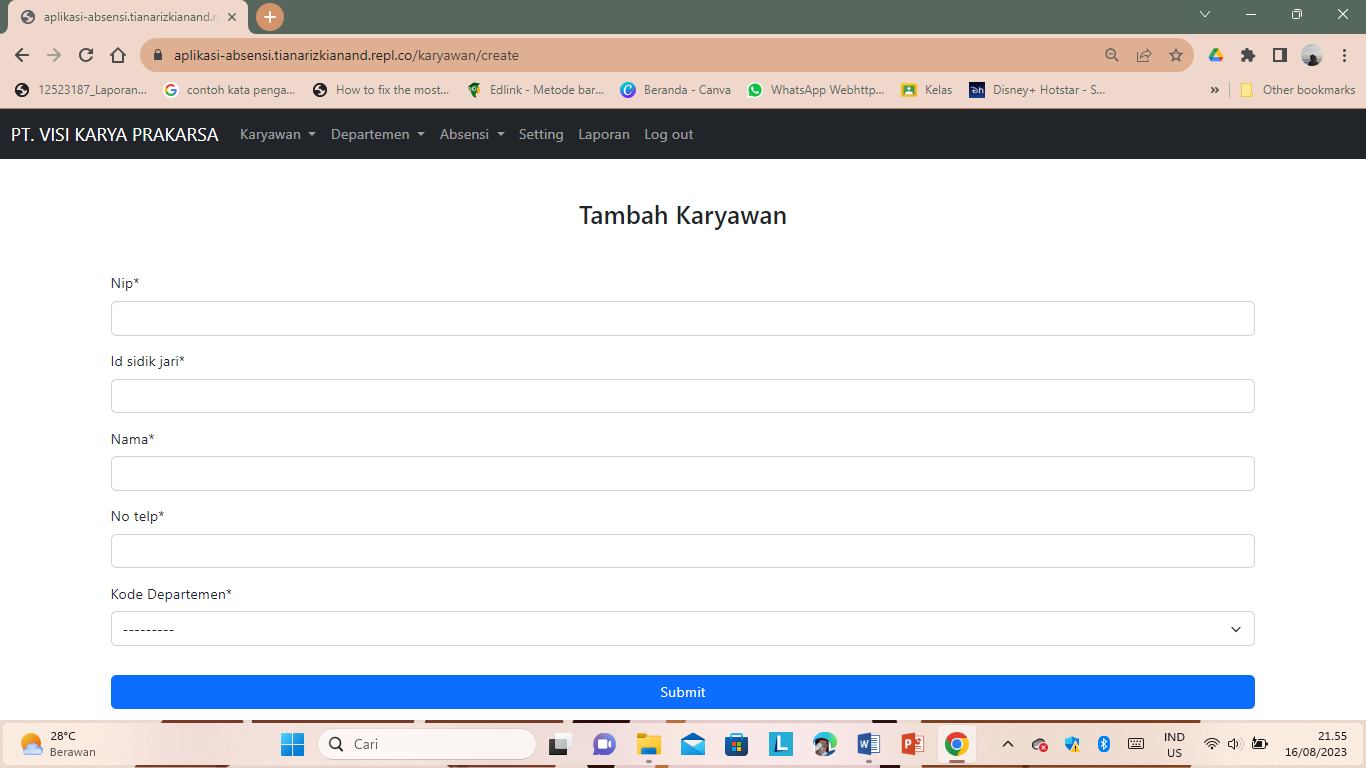
Pada gambar 5.5 berfungsi untuk menghapus data karyawan yang sudah tidak digunakan, pada proses delete akan memunculkan konfirmasi ulang pada admin untuk memastikan kembali bahwa data tersebut akan di *delete* agar menghindari data terhapus tanpa di sengaja oleh admin, terdapat konfirmasi ulang jika data akan dihapus setelah data dihapus maka data akan terhapus pada menu data karyawan. seperti pada gambar di bawah ini :



Gambar 5.5 Halaman Delete Karyawan

1. Halaman tambah karyawan

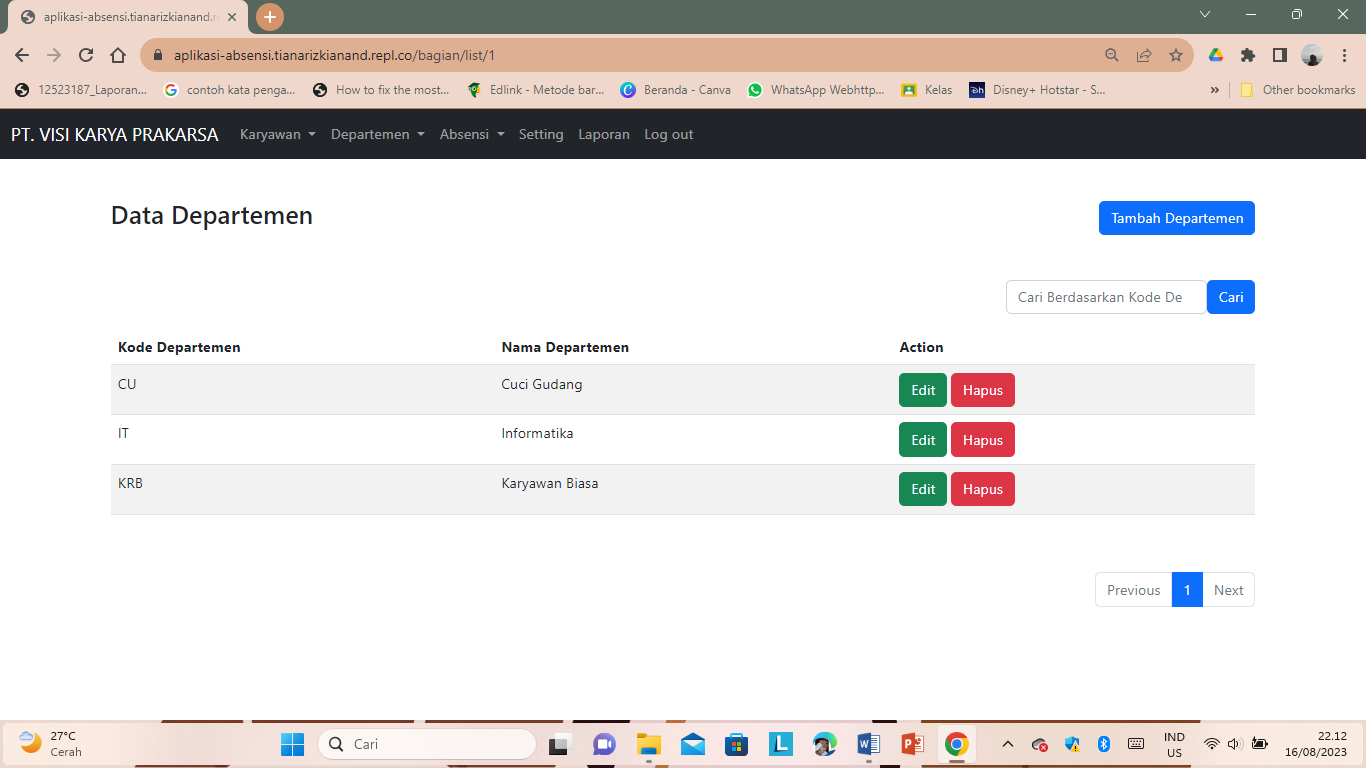
Pada gambar 5.6 tambah karyawan pada aplikasi absensi admin dapat menambahkan data karyawan dengan cara mengklik tombol tambah karyawan kemudian sistem akan memunculkan *form* tambah karyawan, untuk menambah karyawan admin harus menginputkan beberapa data, data yang sudah diinputkan akan muncul pada data karyawan.seperti pada gambar di bawah ini :



Gambar 5.6 Halaman Tambah Karyawan

1. Halaman departemen

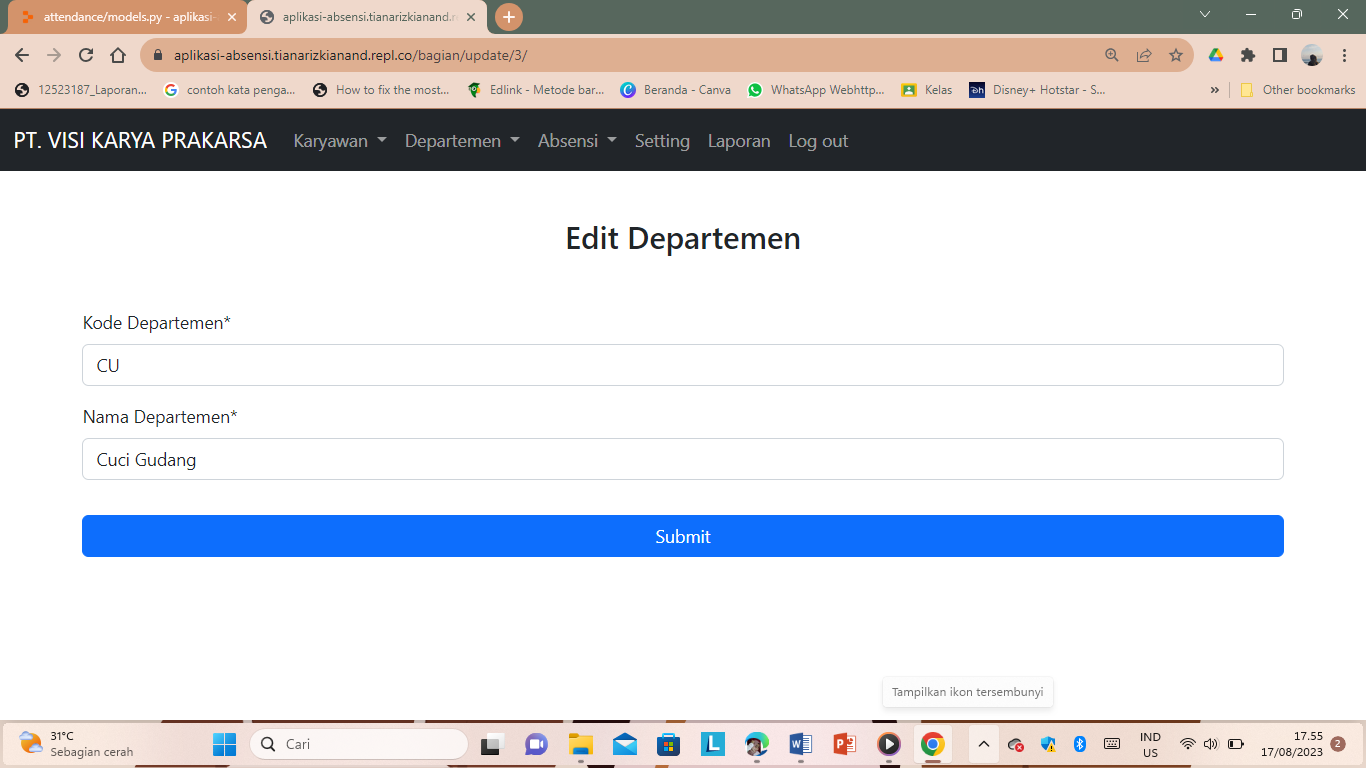
Pada gambar 5.7 departemen yaitu departemen pada persusahaan PT.Visi Karya Prakarsa yang terdapat data nama-nama departemen dan kode departemen pada setiap posisi karyawan pada perusahaan tersebut, nama departemen dan aksi. Dalam fitur aksi, fitur edit pada *action* untuk mengubah data yang sudah diinputkan, *delete* untuk menghapus data yang sudah tidak diperlukan seperti pada gambar dibawah ini :



Gambar 5.7 Halaman Departemen

1. Halaman edit bagian

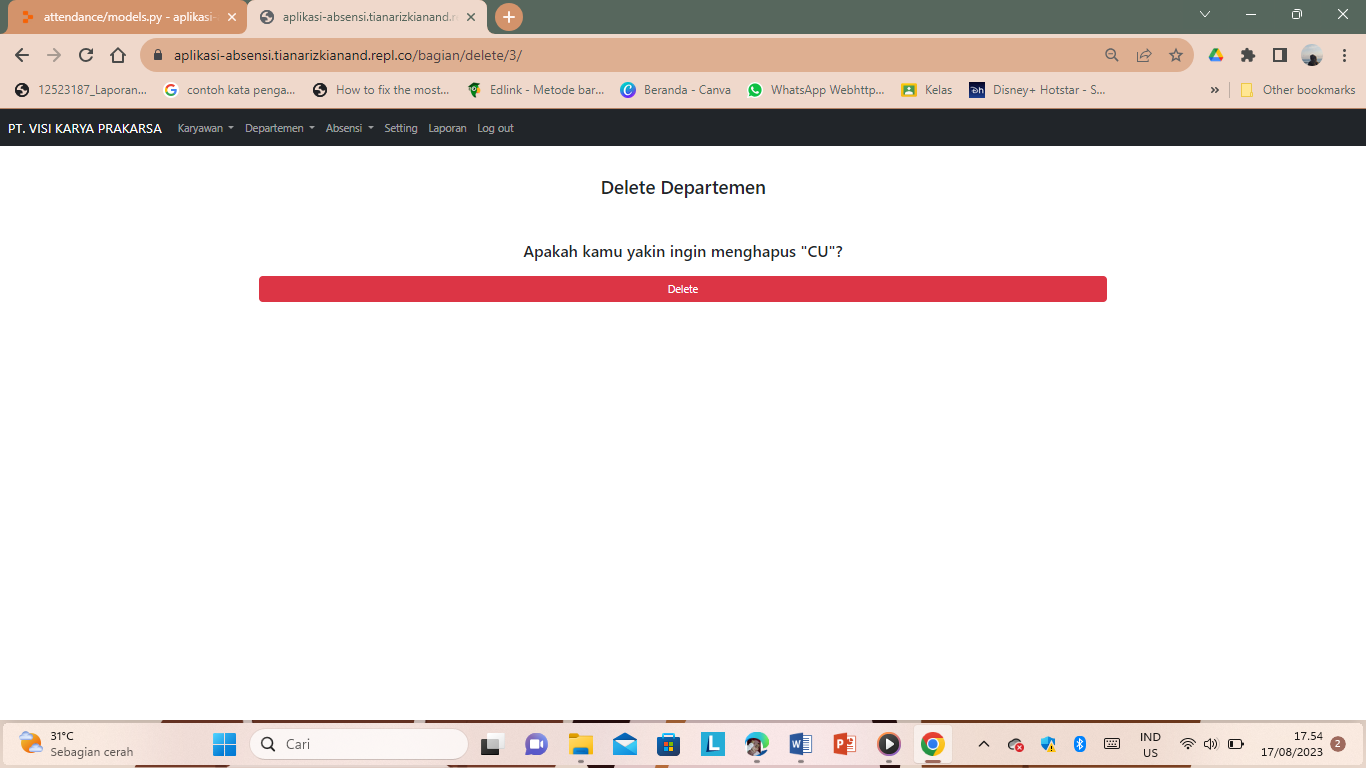
Pada gambar 5.8 ini berfungsi untuk mengedit posisi setiap karyawan yang sudah diinputkan, dan apabila data yang diinputkan tidak sesuai dengan kode bagian proses edit bagian pada aplikasi tidak dapat di submit seperti pada gambar dibawah ini :



Gambar 5.8 Halaman Edit Departemen

1. Halaman *delete* departemen

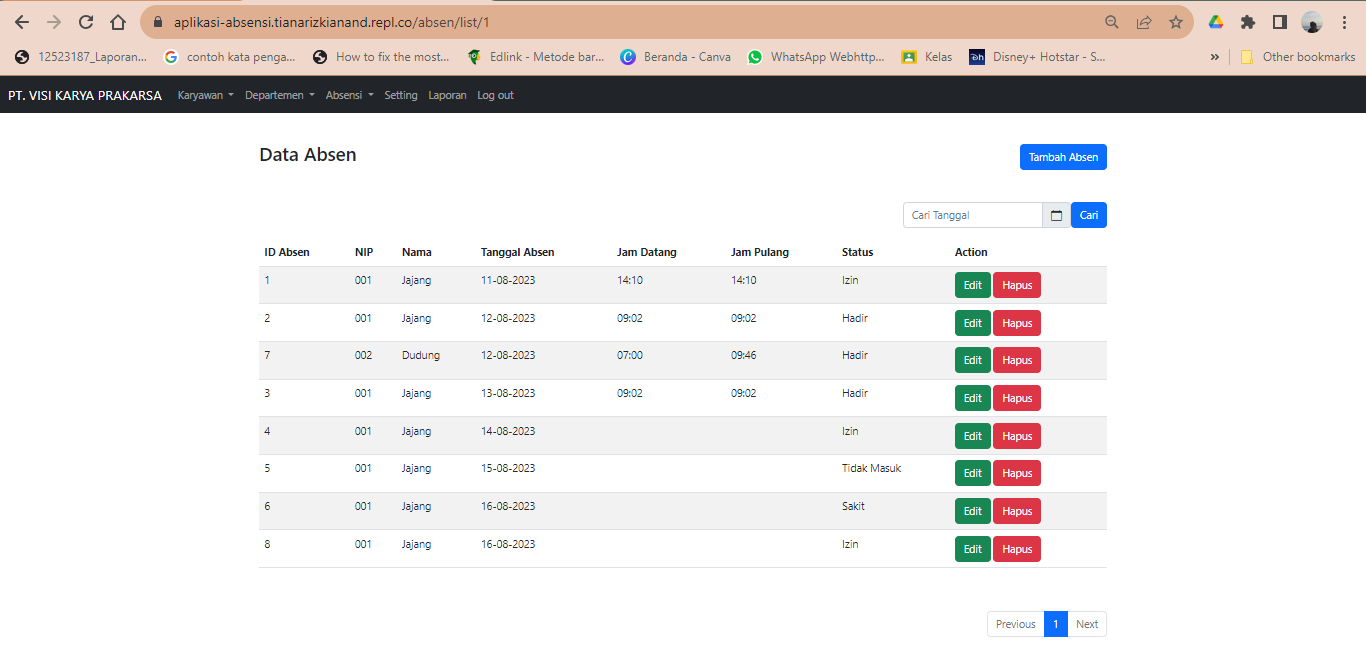
Pada gambar 5.9 proses delete akan memunculkan konfirmasi ulang pada admin untuk memastikan kembali bahwa data tersebut akan di *delete* agar menghindari data terhapus tanpa di sengaja oleh admin seperti pada gambar di bawah ini :



Gambar 5.9 Halaman Delete Departemen

1. Halaman absensi

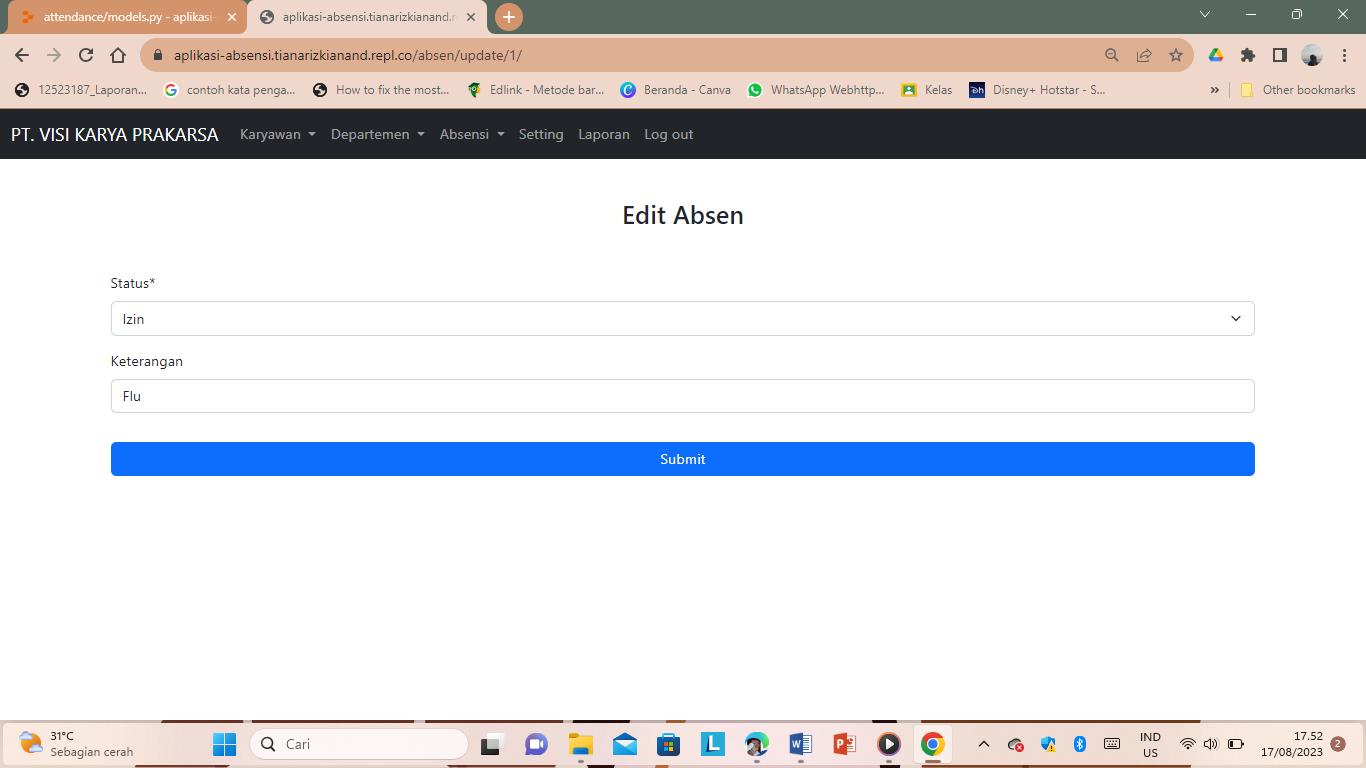
Pada gambar 5.10 absensi terdapat data absen karyawan yang menampilkan data-data yang harus diinputkan. Edit pada *action* untuk mengubah data yang sudah diinputkan, *delete* untuk menghapus data yang sudah tidak diperlukan, data-data yang diperlukan seperti pada gambar dibawah ini :



Gambar 5.10 Halaman Absensi

1. Halaman edit absen

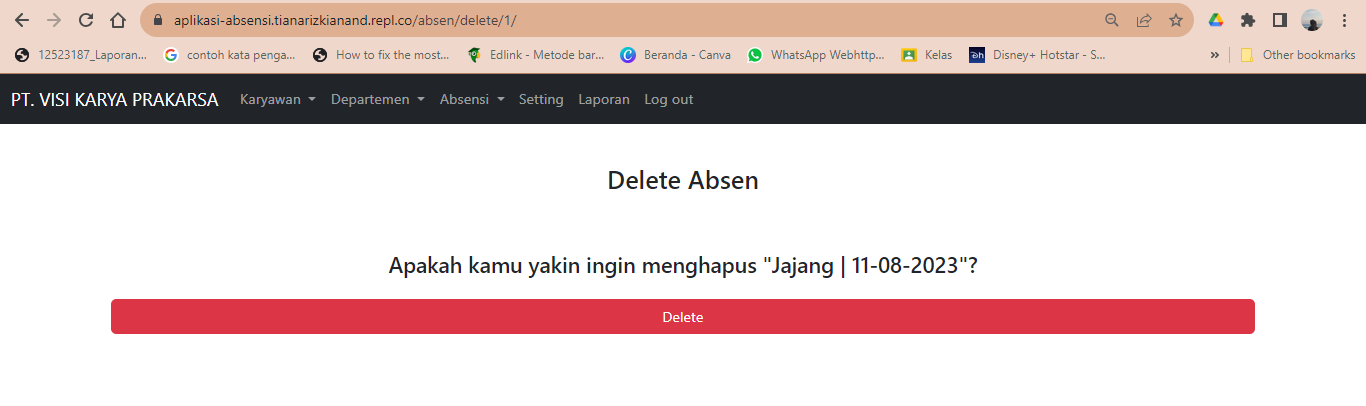
Pada gambar 5.11 edit absen pada aplikasi absensi untuk mengedit data absensi karyawan pada aplikasi harus menginputkan data, seperti pada gambar di bawah ini :



Gambar 5.11 Halaman Edit Absen

1. Halaman delete absen

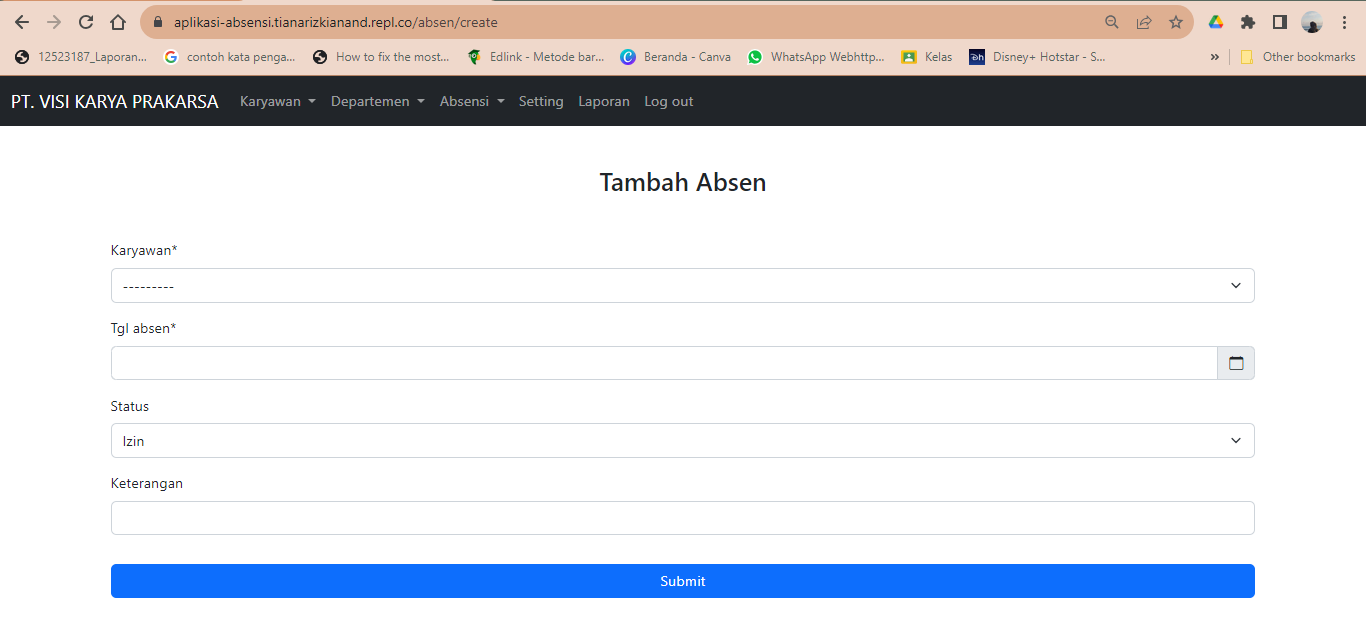
Pada gambar 5.12 proses delete akan memunculkan konfirmasi ulang pada admin untuk memastikan kembali bahwa data tersebut akan di *delete* agar menghindari data terhapus tanpa di sengaja oleh admin, seperti pada gambar di bawah ini :



Gambar 5.12 Halaman Delete Absen

1. Halaman tambah absen

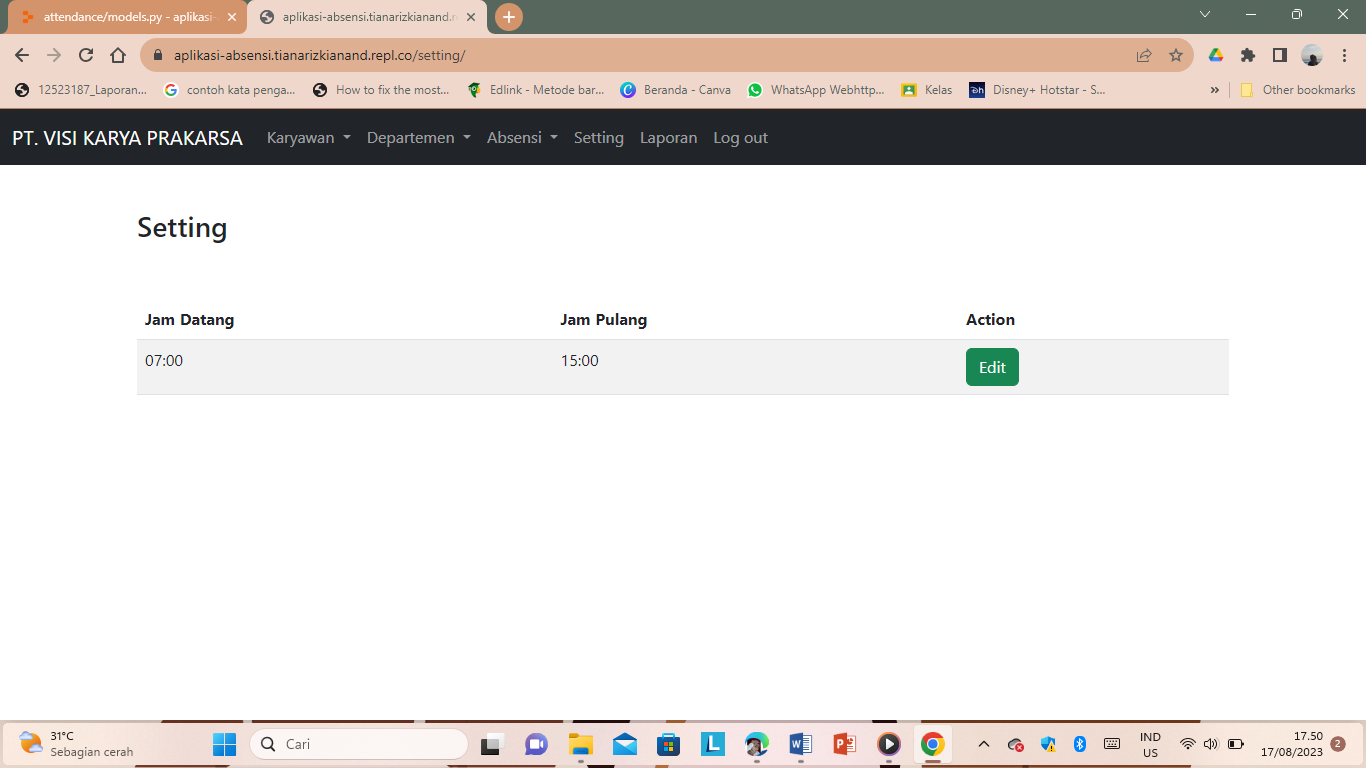
Pada gambar 5.13 tambah absen pada aplikasi absensi untuk menambah data absensi karyawan dengan menginputkan data secara manual pada aplikasi absensi. Data-data yang perlu diinputkan harus dilengkapi maka penambahan absen pada aplikasi ini akan diproses oleh sistem. Edit pada *action* untuk mengubah data yang sudah diinputkan, *delete* untuk menghapus data yang sudah tidak diperlukan seperti pada gambar dibawah ini :



Gambar 5.13 Halaman Tambah Absen

1. Halaman *setting*

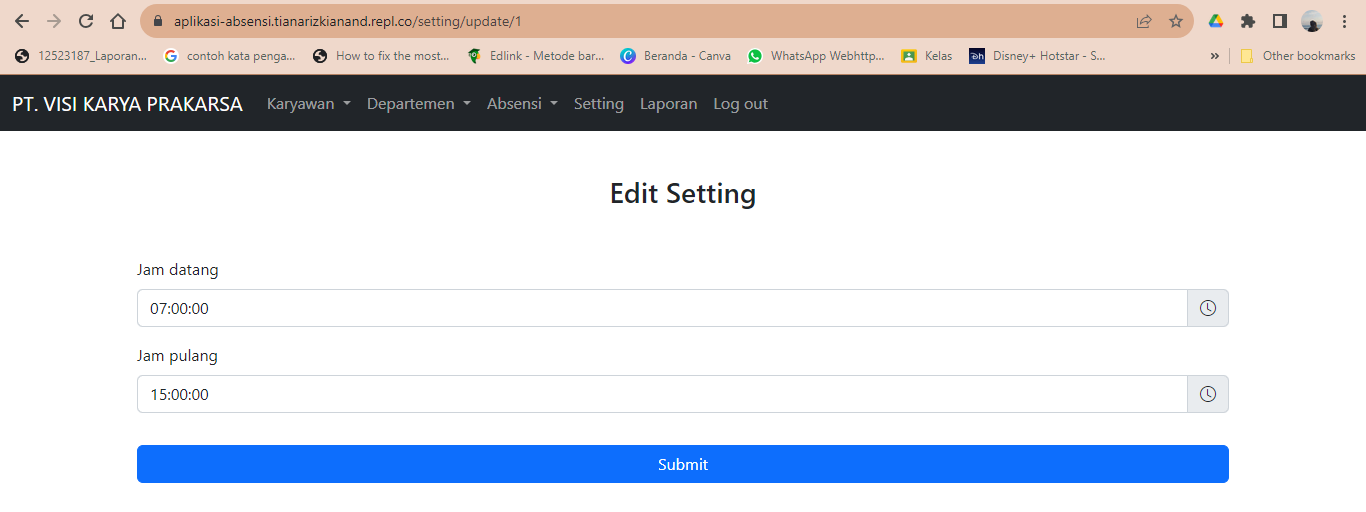
Pada gambar 5.14 *setting* ini menampilkan jam datang, jam pulang dan *action*. Edit pada *action* untuk mengubah data yang sudah diinputkan seperti pada gambar dibawah ini :



Gambar 5.14 Halaman Setting

1. Halaman edit *setting*

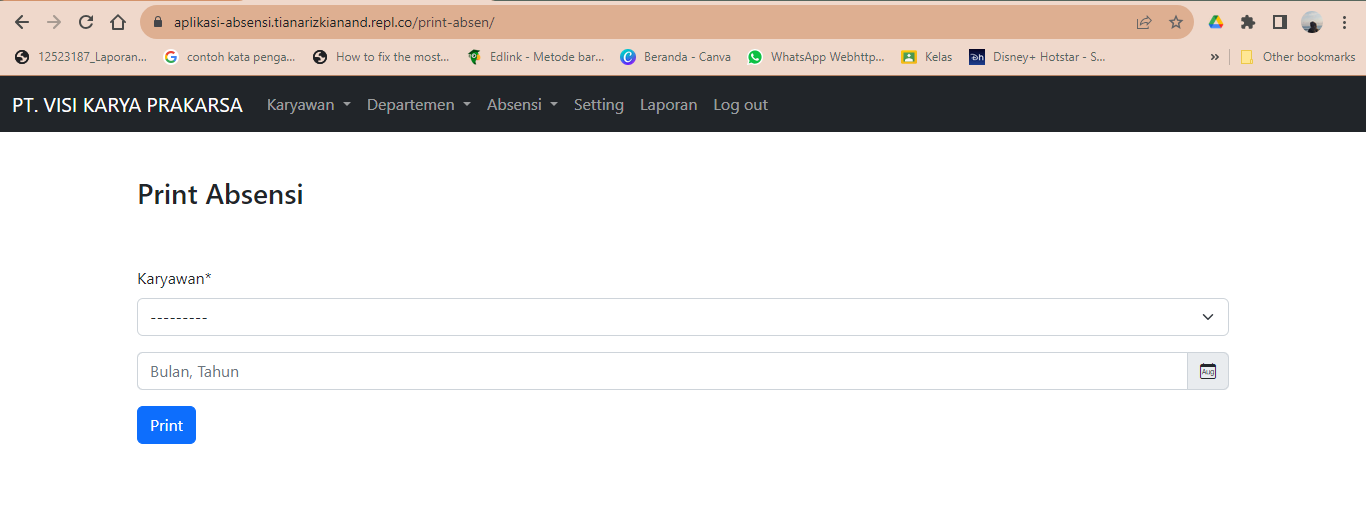
Pada gambar 5.15 ini untuk mengubah jam kerja pada perusahaan PT. Visi Karya Prakarsa yang dimana harus menginputkan jam masuk dan jam keluar yang terbaru, edit pada setting ini dapat diubah ketika pihak perusahaan ingin mengubah jam kerja pada perusahaan, pengubahan pada halaman ini akan menampilkan *update* jam kerja pada halaman *setting* nantinya, tampilan edit *setting* seperti gambar berikut ini :



Gambar 5.15 Halaman Edit Setting

1. Halaman laporan

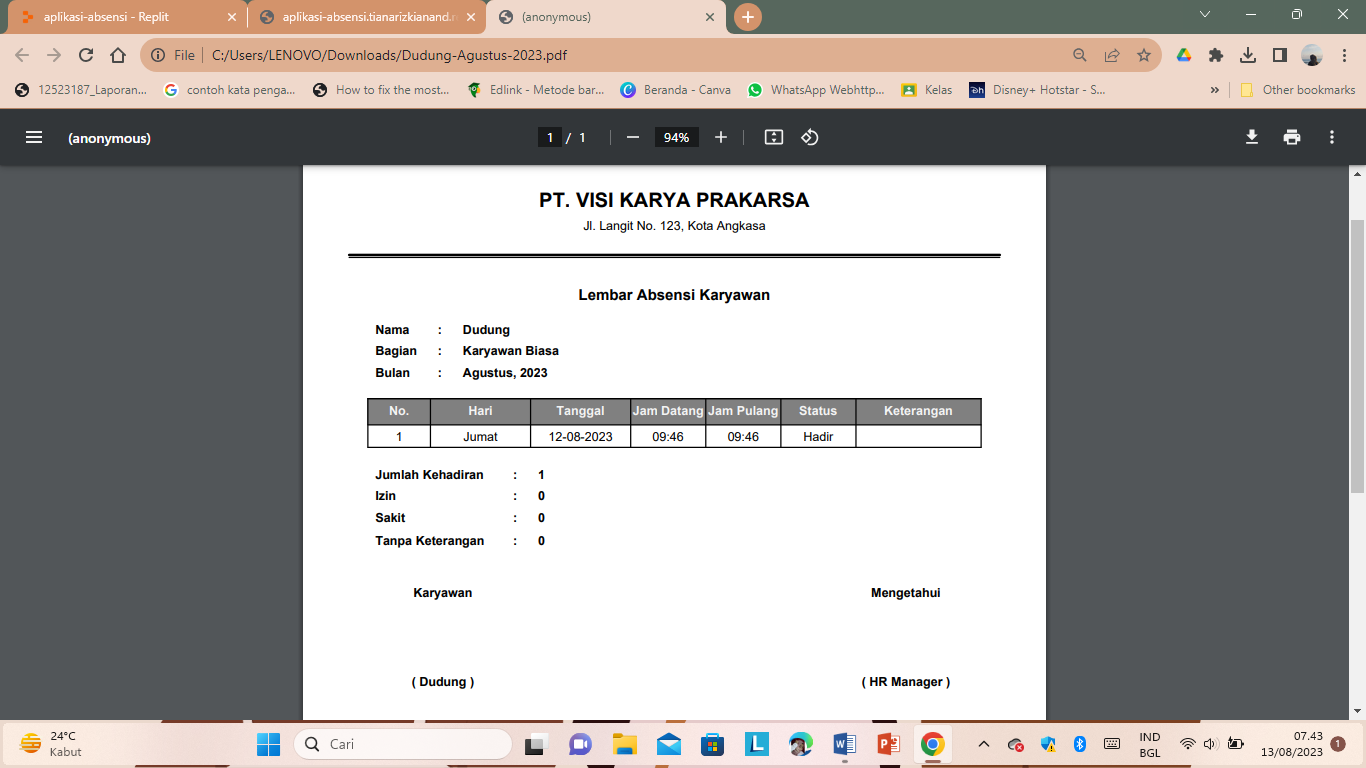
Pada gambar 5.16 ini berisi laporan absensi perbulan setiap karyawan, print laporan absensi pada aplikasi ini dapat otomatis di unduh pada browser. Untuk dapat mengetahui hasil dari laporan ini admin harus mengisi nip, bulan dan tahun terlebih dahulu.



Gambar 5.16 Halaman Laporan

1. Halaman hasil laporan

Pada gambar 5.17 ini berisi laporan karyawan setiap, hasil laporan akan menampilkan jumlah kehadiran karyawan.



Gambar 5.17 Halaman Hasil Print Laporan

## Pengujian

Setelah dilakukan tahap implementasi pada rancang bangun aplikasi, maka tahapan selanjutnya yang dilakukan adalah tahap pengujian. Pada tahapan ini bertujuan untuk melihat apakah aplikasi yang dibuat dengan analisis serta perancangan yang sebelumnya sudah sesuai dan berfungsi sebagaimana mestinya atau belum.

Tabel 5.3 Tabel pengujian aplikasi absensi karyawan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Item uji | Skenario Pengujian | Hasil yang diharapkan | Hasil pengujian | kesimpulan |
| 1. | *Form Login* | Mengisi *username* dan *password* | Admin dapat masuk ke aplikasi absensi | Sesuai harapan | Valid |
| 2. | Halaman *Dashboard* | Klik *login* kemudian masuk ke halaman *dashboard* | Admin dapat mengakses halaman *dashboard* dan melihat menu-menu yang terdapat pada menu *dashboard* | Sesuai Harapan | Valid |
| 3. | Karyawan | Klik karyawan pada halaman *dashboard* | Admin dapat mengakses data karyawan dan tambah karyawan | Sesuai harapan | Valid |
| 4. | Tambah Karyawan | Klik tambah karyawan pada menu tambah karyawan | Admin dapat menambahkan data karyawan | Sesuai harapan | Valid |
| 5. | Edit data karyawan | Klik edit pada baris data yang dipilih | Admin dapat melakukan edit data karyawan | Sesuai harapan | Valid |
| 6. | *Delete* data karyawan | Klik *delete* pada baris data yang dipilih | Admin dapat menghapus data karyawan | Sesuai harapan | Valid |
| 7. | Departemen | Klik bagian pada halaman *dashboard* | Admin dapat mengakses data departemen dan tambah departemen | Sesuai harapan | Valid |
| 8. | Tambah departemen | Klik tambah bagian pada menu tambah departemen | Admin dapat menambahkan data departemen | Sesuai harapan | Valid |
| 9. | Edit data departemen | Klik edit pada baris data yang dipilih | Admin dapat melakukan edit data departemen | Sesuai harapan | Valid |
| 10. | *Delete* data departemen | Klik *delete* pada baris data yang dipilih | Admin dapat menghapus data departemen | Sesuai harapan | Valid |
| 11. | Absensi | Klik absensi pada halaman *dashboard* | Admin dapat mengakses data absen dan tambah absen. | Sesuai harapan | Valid |
| 12. | Tambah absen | Klik tambah absen pada menu tambah absen | Admin dapat menambahkan data absen | Sesuai harapan | Valid |
| 13. | Edit data absen | Klik edit pada baris data yang dipilih | Admin dapat melakukan edit data absen | Sesuai harapan | Valid |
| 14. | *Delete* data absen | Klik *delete* pada baris data yang dipilih | Admin dapat menghapus data absen | Sesuai harapan | Valid |
| 15. | *Setting* | Klik setting pada halaman *dashboard* | Admin dapat mengakses data *setting* dan mengedit data | Sesuai harapan | Valid |
| 16. | Edit data *Setting* | Klik edit pada baris data yang dipilih | Admin dapat melakukan edit data *setting* | Sesuai harapan | Valid |
| 17. | Laporan | Klik laporan pada dashboard | Admin dapat mengakses laporan untuk print absensi perbulannya pada aplikasi lalu hasil print akan otomatis di unduh di browser | Sesuai harapan | Valid |
| 19. | *Logout* | Klik *logout* | Admin dapat keluar dari aplikasi absensi | Sesuai harapan | Valid |

Berdasarkan tabel 5.3 yang menampilkan tabel hasil pengujian, maka dapat disimpulkan bahwa hasil pengujian aplikasi sudah berhasil dan sesuai dengan perancangan dan analisis yang dilakukan, semua fitur berjalan dengan baik sesuai dengan fungsi yang seharusnya dan penyusun harapkan. Tetapi pada alat belum terimplementasi dikarenakan alat fingerprint tidak terkoneksi, karena micropython termasuk keluaran terbaru, dan belum menemukan library yang cocok dan library nya sangat terbatas, library untuk sidik jari hanya ada satu dan ketika dicoba pada bread board terjadi error.

# BAB VI KESIMPULAN DAN SARAN

## Kesimpulan

Berdasarkan penelitian yang dilakukan penulis melalui beberapa tahapan yang dilakukan pada bab-bab sebelumnya, maka penulis dapat menyimpulkan bahwa :

1. Dari hasil penelitian diatas, pada perancangan sistem absensi dengan menggunakan *fingerprint* ini berhasil membuat sistem absensi yang dapat mencetak hasil laporan absensi bulanan dengan waktu yang cepat dan mendapatkan hasil yang akurat.
2. Untuk mengurangi kecurangan dalam absensi karyawan dengan menggunakan sistem yang terkomputerisasi, yaitu dengan menggunakan teknologi *fingerprint* dengan algoritma *minutiae,* sistem ini akan mengurangi resiko rekaman sidik jari palsu atau yang di manipulasi. Dalam perancangan aplikasi ini mencakup data karyawan, departemen karyawan, absensi karyawan, serta laporan absensi perbulannya. Dengan adanya aplikasi ini diharapkan dapat membantu PT. Visi Karya Prakarsa untuk mengetahui setiap rekap absensi karyawan dalam periode yang ditentukan dan untuk mengurangi kecurangan absensi.
3. Dengan adanya studi penelitian ini aplikasi absensi dapat di implementasikan secara praktis dan mudahsehingga pembuatan laporan absensi dapat dilakukan dengan cepat karena aplikasi absensi dapat diakses dari browser mana saja dan kapan saja. Aplikasi absensi ini hanya menggunakan satu *user* saja yaitu admin, sehingga laporan absensi karyawan yang bertujuan untuk direkap kepada atasan sebagai pihak yang berkepentingan dengan aktivitas tersebut, serta laporan absensi selanjutnya hanya dapat melihat laporan dari aplikasi dan tentunya yang akan dijelaskan oleh admin.

## 6.2 Saran

Berdasarkan hasil penelitian perancangan aplikasi absensi menggunakan *fingerprint*, aplikasi yang penulis buat ini masih memiliki kekurangan yang dapat diperbaiki atau bahkan dikembangkan untuk mendapatkan hasil yang lebih baik lagi. Maka peneliti merekomendasikan atau menyarankan beberapa hal, yaitu :

1. Alat pada *fingerprint* memiliki kekurangan pada proses penyambungan dengan *database* maupun pada apikasi, yang dimana kekurangan ini dapat diperbaiki dan dikembangkan dengan menggunakan alat *fingerprint* lain yang lebih terbaru.
2. Pada perancangan alat ini hanya menggunakan alat sensor citra sidik jari sederhana. Dan adanya pelatihan sumber data manusia sehingga dapat mengoperasikan aplikasi ini dan ada pengembangan dari aplikasi terintegrasi yang dapat mendukung kegiatan pada perusahaan. Pada aplikasi ini juga hanya dapat diakses oleh manager yang bersangkutan dengan rekap data absensi, untuk karyawan hanya memiliki akses untuk absensi saja.
3. Analisis laporan absensi dalam aplikasi perlu ditingkatkan kembali agar aplikasi dapat menghasilkan laporan yang lebih informatif seperti menambahkan hasil pekerjaan.

# 

# DAFTAR PUSTAKA

Dewanti, I. P., Prasasti, A. L., & ... (2020). Pengenalan Sidik Jari Manusia Terdistorsi Menggunakan Algoritma Sift Based Minutia Descriptor (smd). *EProceedings …*, *7*(2), 4774–4783. https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/12379%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/download/12379/12157

Faiza Alif Fakhrina, Rahmadwati, Rahmadwati, W. (2016). Thinning Zhang-Suen dan Stentiford untuk Menentukan Ekstraksi Ciri (Minutiae) Sebagai Identifikasi Pola Sidik Jari. *Majalah Ilmiah Teknologi Elektro*, *15*(2), 127–133. https://doi.org/10.24843/mite.1502.19

Harumy, T.H.F., Julham Sitorus, M. L. (2018). Sistem Informasi Absensi Pada Pt . Cospar Sentosa Jaya Menggunakan Bahasa Pemprograman Java. *Jurnal Teknik Informartika*, *5*(1), 63–70.

Meyliana, A. (2020). Perancangan Sistem Informasi Presensi Karyawan Dengan Metode Prototype Menggunakan Fingerprint. *Speed - Sentra Penelitian Engineering Dan Edukasi*, *12*(2), 1–3.

Muhyidin, M. A., Sulhan, M. A., & Sevtiana, A. (2020). Perancangan Ui/Ux Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma. *Jurnal Digit*, *10*(2), 208. https://doi.org/10.51920/jd.v10i2.171

Nasution, A. P., Osmond, A. B., Saputra, R. E., Harumy, T.H.F., Julham Sitorus, M. L., Baharuddin, M. M., Azis, H., & Hasanuddin, T. (2018). Deteksi Kepribadian Anak Dengan Pengolahan Citra Sidik Jari Menggunakan Metode Minutiae Detection of Children ’ S Personality With Fingerprint Image. *Jurnal Teknik Informartika*, *5*(3), 6102–6109. https://doi.org/10.33096/ilkom.v11i3.489.269-274

PRASASTI, A. L., IRAWAN, B., FAJRI, S. E., RENDIKA, A., & HADIYOSO, S. (2020). Perbandingan Ekstraksi Fitur dan Proses Matching pada Autentikasi Sidik Jari Manusia. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, *8*(1), 95. https://doi.org/10.26760/elkomika.v8i1.95

Purba, G. (2021). Kombinasi Metode Minutiae Dan K-Nearest Neighbor Untuk Pengenalan Pola Pada Citra Fingerprint. *Nasional Teknologi Informasi Dan Komputer)*, *5*(1), 228–234. https://doi.org/10.30865/komik.v5i1.3712

Putra, G. R., Situmorang, B. H., & Hidayatullah, S. (2023). Identifikasi Biometrika Menggunakan Ekstraksi Minutiae Pada Citra Sidik Jari. *Jurnal Teknoinfo*, *17*(1), 289. https://doi.org/10.33365/jti.v17i1.2376

Rizaldi, F. M. (2022). *SISTEM PRESENSI KARYAWAN DENGAN TEKNOLOGI GPS BERBASIS WEB ( STUDI KASUS PT . MERSIFARMA ) PROGRAM STUDI TEKNIK INFORMATIKA SUKABUMI AGUSTUS 2022*.

Sikumbang, M. A. R., Habibi, R., & Pane, S. F. (2020). Sistem Informasi Absensi Pegawai Menggunakan Metode RAD dan Metode LBS Pada Koordinat Absensi. *Jurnal Media Informatika Budidarma*, *4*(1), 59. https://doi.org/10.30865/mib.v4i1.1445

# LAMPIRAN

**Lampiran 1 Hasil Wawancara**

Narasumber : Lucky Luqman Nurhakim

Jabatan/Posisi : CTO

Hari/Tanggal : 28 Maret 2023

Instansi : PT. Visi Karya Prakarsa

“Perancangan Aplikasi Absensi Fingerprint Menggunakan Algoritma Minutiae Untuk Meningkatkan Akurasi Dan Efisiensi Absensi Karyawan Pada PT. Visi Karya Prakarsa" adalah judul penelitian yang menggunakan wawancara ini sebagai satu-satunya sumber data. Berikut adalah daftar pertanyaan beserta jawabannya :

|  |  |  |
| --- | --- | --- |
| No | Pertanyaan | Jawaban |
| 1 | Perusahaan bergerak dibidang apa ? | Perusahaan yang bergerak di bidang Teknologi Informasi ini menyediakan layanan produk dan jasa. |
| 2 | Bagaimana struktur pada perusahaan ? | Komisaris dan adviser bertugas sebagai penasihat klien atau ada masalah di perusahaan dapat dikomunikasikan dengan komisaris dan adviser. CEO sebagai pemimpin yang menahkodai perusahaan seperti strategi – strategi pada perusahaan dipegang oleh CEO. COO yang bergerak di operasional seperti marketing, sales, operasional itu sendiri dan *team* bisnis semua itu ada dibawah naungan COO. CTO berperan aktif pada produk di RNB atau di produk sendiri, dan developer itu sendiri ada pada naungan CTO. CTO secara aktif terlibat dengan produk, apakah itu di manufaktur sendiri atau tidak. |

|  |  |  |
| --- | --- | --- |
| 3 | Bagaimana strategi pemasaran dari perusahaan? | E-commerce, belanja online, pemasaran digital, dan penggunaan media sosial, termasuk YouTube, Instagram, dan beberapa kali mengadakan acara dan berafiliasi dengan bisnis lain. |
| 4 | Bagaimana proses klien melakukan kontrak kerja dengan perusahaan saat akan menggunakan layanan ? | Proses kontrak dengan klien dilakukan dalam sebuah pertemuan. Perusahaan yang bersangkutan mempresentasikan produk dan layanannya selama pertemuan yang dimaksud. Kemudian dilakukan proses jual beli dan negosiasi. Setelah selesai, luncurkan proses instalasi atau konfigurasikan aplikasi di lokasi klien. |
| 5 | Kendala apa yang terjadi dalam proses klien saat akan melakukan kontrak kerja dengan perusahaan ? | Pada PT. Visi Karya Prakarsa terdapat salah satu layanan yang disebut dengan *Antrique* yang dimana *antrique* adalah aplikasi antrian. Kendala yang ada pada klien ketika menggunakan jasa *antrique* yaitu jika klien tidak memiliki mesin antrian maka klien otomatis harus membuat terlebih dahulu mesin antrian, jika sudah klien bisa melakukan set up pada aplikasi. |
| 6 | Sistem absensi karyawan pada PT. Visi Karya Prakarsa ini bagaimana ? | Menggunakan sistem manual by report yang dilakukan karyawan setiap pagi melakukan daily meeting mengumpulkan data hasil pekerjaan karyawan setiap divisi setelah selesai melakukan kolek baru di report ke manajemen untuk report. |
| 7 | Bagaimana sistem pelaporan absensi karyawan setiap bulannya ? | Tidak diketahui untuk report per bulannya dikarenakan menggunakan sistem manual yang memungkinkan . |

|  |  |  |
| --- | --- | --- |
| 8 | Apakah ada kendala dalam proses pelaporan data absensi dan saat melakukan absensi manual ? | Ada, dikarenakan laporan absensi dilakukan per hari maka pelaporan bulanan tidak diketahui, absensi karyawan setiap harinya yang memungkinkan terjadi kehilangan pada absensi karyawan jika menggunakan sistem report manual. |
| 9 | Untuk data absensi karyawan ini biasanya membutuhkan data apa saja ? | Data yang dibutuhkan yaitu menggunakan data hasil pekerjaan setiap harinya. |

|  |  |
| --- | --- |
| Pewawancara | Narasumber |
|  |  |
| Tiana Rizki Ananda | Lucky Luqman Nurhakim |

**Lampiran 2 Hasil Dokumentasi**

Narasumber : Lucky Luqman Nurhakim

Jabatan/Posisi : CTO *(Chief Technology Officer)*

Instansi : PT. Visi Karya Prakarsa

Lokasi : Jl. Jupiter Barat No.37, Sekejati, Kecamatan Buahbatu

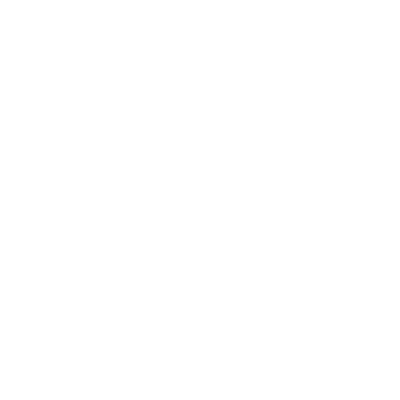
Dokumentasi wawancara dengan salah satu karyawanpada PT. Visi Karya Prakarsa.



# 

|  |
| --- |
| RIWAYAT HIDUP PENULIS |

|  |  |
| --- | --- |
| **Data Diri** | |
| Nama | : Tiana Rizki Ananda |
| Tempat/ Tanggal Lahir | : Bandung, 16 Mei 2001 |
| Jenis Kelamin | : Perempuan |
| Agama | : Islam |
| Status | : Belum Menikah |
| Alamat | : Kp. Sukaasih Kec. Majalaya, Des. Majasetra, Kab. Bandung, Jawa Barat |
| Nama Ayah | : Soni Rustandi (Alm) |
| Nama Ibu | : Rita Nurhayati |



**Hobby**

Bermain basket, jogging, berenang fotografi dan *editing*

**Riwayat Pendidikan**

* TK Al-Hidayah 2006 - 2007
* SDN Cimahi Mandiri 3 2007 - 2013
* SMPN 1 Ibun 2013 - 2016
* SMK 2 LPPM RI Majalaya 2016 - 2019
* Universitas Bale Bandung (S1- Teknik Informatika) 2019 - 2023

**Organisasi**

* Kominfo Hima Prodi Teknik Informatika 2020 – 2021

**Motto**

* *Choose urself*
* Kesuksesan yang besar dimulai dari langkah yang kecil

***Contact***

|  |  |
| --- | --- |
| * No. Telepon | : 0851-5733-9593 |
| * Instagram | : tianarizkiananda |
| * Facebook | : Tiana Rizki Ananda |
| * E-mail | : rizkiatiana@gmail.com |