**APLIKASI WEB PRESENSI BERBASIS GPS**

**(Studi Kasus: CV Asian Technology Inspira)**

**KERJA PRAKTIK**

**Oleh**

**NAMA : ALVIKA AJI PRAHASTA NPM : 41155050190037**

**NAMA : RADITYA FIRMANSYAH NPM : 41155050190065**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS LANGLANGBUANA**

**2022**

# LEMBAR PENGESAHAN

**WEB SISTEM PRESENSI DENGAN GPS**

**(Studi Kasus: CV Asian Technology Inspira)**

**Oleh**

**NAMA : ALVIKA AJI PRAHASTA NPM : 41155050190037**

**NAMA : RADITYA FIRMANSYAH NPM : 41155050190065**

Menyetujui

Pembimbing Prodi Pembimbing Lapangan



(Agustinus Andriyanto, S.T., M.T., Ph.D.) (Della Kharisma Dewi, S.M)

NIDN: <NIDN> NIK: 3910531130171

Mengetahui

Ketua Program Studi

Teknik Informatika

(Yiyi Supendi, S.Kom., M.T)

NIDN: <NIDN>

# KATA PENGANTAR

Puji syukur penyusun panjatkan atas kehadirat Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya kepada penyusun sehingga penyusun dapat menyelesaikan laporan Kerja Praktek ini yang berjudul *“Web Sistem Presensi dengan GPS (Studi Kasus: CV. Asian Technology Inspira)”*.

Penelitian ini bermaksud untuk memudahkan perusahaan tersebut untuk mengelola data presensi yang semula konvensional menjadi digital. Tugas Kerja Praktek ini diajukan untuk melengkapi salah satu syarat dalam menempuh Ujian Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Langlangbuana.

Penyusun menyadari sepenuhnya bahwa penyusunan laporan ini masih jauh dari kesempurnaan dan terdapat kekurangan-kekurangan yang harus diperbaiki. Penyusun meminta masukan berupa pandangan serta kritik yang bersifat membangun semua pihak. Semoga Allah SWT. membalas kebaikan semua pihak yang telah banyak membantu dalam penyusunan laporan ini. Aamiin.

Bandung, 19 Mei 2022

Penyusun

# RINGKASAN

Presensi merupakan suatu model untuk mencatat jam kedatangan dan jam kepulangan seseorang. Presensi pada umumnya digunakan pada lingkungan yang formal seperti sekolah, ataupun lingkungan pekerjaan. Tujuan dari penggunaan absensi adalah untuk memantau data kehadiran serta sebagai sarana untuk mengukur tingkat kedisiplinan seseorang. Hal ini tentunya cukup penting, terutama pada lingkungan pekerjaan, karena apabila sebuah perusahaan memiliki karyawan yang disiplin, maka tingkat produktifitas perusahaan tersebut akan semakin besar pula. Terdapat berbagai metode dalam melakukan presensi, mulai dari pencatatan data secara manual, melalui mesin *finger print*, maupun melalui sistem. Di CV Asian Technology Inspira Presensi dilakukan dengan cara mengirimkan lokasi kita dan keterangan hadirnya menggunakan *Telegram*, tentunya ini sangat tidak efektif dan konvensional karena lokasi di *Telegram* tersebut dapat dimanipulasi. Atas dasar permasalahan inilah peneliti mencoba untuk mengembangkan aplikasi sistem presensi berbasis web dan mengangkatnya sebagai judul kerja praktek. Aplikasi ini diharapkan mampu untuk meminimalisir resiko manipulasi saat melakukan presensi. Selain hal tersebut, sistem ini juga diharapkan mampu untuk membantu HRD dalam mengolah data presensi secara mudah dan *modern*.

Kata Kunci: Sistem Presensi, *GPS*, Web Aplikasi

# DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc120696087)

[KATA PENGANTAR ii](#_Toc120696088)

[RINGKASAN iii](#_Toc120696089)

[DAFTAR ISI iv](#_Toc120696090)

[DAFTAR GAMBAR vii](#_Toc120696091)

[DAFTAR TABEL x](#_Toc120696092)

[DAFTAR SINGKAT DAN ISTILAH xii](#_Toc120696093)

[DAFTAR SIMBOL xiii](#_Toc120696094)

[BAB I PENDAHULUAN I-1](#_Toc120696095)

[I.1 Latar Belakang I-1](#_Toc120696096)

[I.2 Maksud dan Tujuan I-3](#_Toc120696097)

[I.3 Ruang Lingkup I-3](#_Toc120696098)

[I.4 Keluaran dan kerja praktik I-3](#_Toc120696099)

[I.5 Lokasi dan Waktu Pelaksanaan I-4](#_Toc120696100)

[I.6 Sistematika Penulisan I-5](#_Toc120696101)

[BAB II KAJIAN PUSTAKA II-1](#_Toc120696102)

[II.1 Definisi Sistem II-1](#_Toc120696103)

[II.2 GPS (Global Positioning System) II-1](#_Toc120696104)

[II.3 Presensi Karyawan II-2](#_Toc120696105)

[II.4 *Web Development Tools* II-2](#_Toc120696106)

[II.5 *Database Management System (DBMS)* II-9](#_Toc120696107)

[II.6 *Website* II-10](#_Toc120696108)

[II.7 *Hierarchical Model View Controller (HMVC)* II-10](#_Toc120696109)

[II.8 Rekayasa Perangkat Lunak II-11](#_Toc120696110)

[II.9 *Rapid Application Development (RAD)* II-12](#_Toc120696111)

[BAB III PROSES BISNIS DAN ARSITEKTUR SISTEM III-1](#_Toc120696112)

[III.1 Proses Bisnis III-1](#_Toc120696113)

[III.2 Arsitektur Sistem III-3](#_Toc120696114)

[BAB IV HASIL DAN PEMBAHASAN IV-1](#_Toc120696115)

[IV.1 Requirement Model IV-1](#_Toc120696116)

[IV.1.1 Identifikasi Hak Akses IV-1](#_Toc120696117)

[IV.1.2 Identifikasi Kebutuhan Fungsional IV-1](#_Toc120696118)

[*IV.1.3* *Use case Model* IV-3](#_Toc120696119)

[IV.2 Design Model IV-10](#_Toc120696120)

[*IV.2.1* *Diagram Activity* IV-10](#_Toc120696121)

[*IV.2.2* *Diagram Sequence* IV-20](#_Toc120696122)

[IV.2.3 *Class* Diagram IV-25](#_Toc120696123)

[IV.2.4 Perancangan Basis Data IV-26](#_Toc120696124)

[IV.2.5 Perancangan Tampilan IV-27](#_Toc120696125)

[IV.2.6 Implementasi Antarmuka IV-40](#_Toc120696126)

[*IV.2.7* Implementasi *Code* IV-51](#_Toc120696127)

[IV.3 Test Model IV-55](#_Toc120696128)

[IV.3.1 Rencana Pengujian IV-55](#_Toc120696129)

[IV.3.2 Implementasi *Code Controller* IV-60](#_Toc120696130)

[BAB V SIMPULAN DAN SARAN V-1](#_Toc120696131)

[V.1 Simpulan V-1](#_Toc120696132)

[V.2 Saran V-2](#_Toc120696133)

[DAFTAR PUSTAKA xix](#_Toc120696134)

[LAMPIRAN A: KODE PROGRAM CONTROLLER PRESENSI MASUK DAN PULANG xxi](#_Toc120696135)

[LAMPIRAN B: KODE PROGRAM CONTROLLER DETAIL PRESENSI xxv](#_Toc120696136)

[LAMPIRAN C: KODE PROGRAM CONTROLLER HAPUS PRESENSI xxvi](#_Toc120696137)

[LAMPIRAN D: KODE PROGRAM CONTROLLER EXPORT PRESENSI xxvii](#_Toc120696138)

[LAMPIRAN E: KODE PROGRAM CONTROLLER FILTER PRESENSI xxx](#_Toc120696139)

# DAFTAR GAMBAR

[Gambar I. 1 Lokasi Kerja Praktek I-4](#_Toc120345668)

[Gambar II. 1 Skema metode pengembangan perangkat lunak RAD (Kendall, 2010)…………………………………………………………………………..II-12](#_Toc120345684)

[Gambar III. 1 Gambar Proses Bisnis Berjalan…………………………………III-2](#_Toc120345700)

[*Gambar III. 2* Proses Bisnis Usulan III-3](#_Toc120345701)

[Gambar III. 3 Gambar Proses Arsitektur Sistem III-3](#_Toc120345702)

[Gambar IV. 1 Diagram *use case*…………………………………………….....IV-4](#_Toc120345707)

[Gambar IV. 2 Diagram activity login IV-10](#_Toc120345708)

[Gambar IV. 3 Diagram *activity* kelola data *user* IV-11](#_Toc120345709)

[Gambar IV. 4 Diagram *activity setting* jam IV-12](#_Toc120345710)

[Gambar IV. 5 Diagram *activity* isi presensi IV-13](#_Toc120345711)

[Gambar IV. 6 Diagram *activity* cek lokasi IV-15](#_Toc120345712)

[Gambar IV. 7 Diagram *activity* tampil data presensi sendiri IV-16](#_Toc120345713)

[Gambar IV. 8 Diagram *activity* kelola data presensi *user* IV-17](#_Toc120345714)

[Gambar IV. 9 Diagram *activity export* data presensi ke *pdf* IV-18](#_Toc120345715)

[Gambar IV. 10 Keterangan diagram *activity export* data presensi ke *pdf* IV-19](#_Toc120345716)

[Gambar IV. 11 *Sequence* diagram *login* IV-20](#_Toc120345717)

[Gambar IV. 12 *Sequence* diagram tambah *user* IV-20](#_Toc120345718)

[Gambar IV. 13 *Sequence* diagram edit *user* IV-21](#_Toc120345719)

[Gambar IV. 14 *Sequence* diagram *delete user* IV-21](#_Toc120345720)

[Gambar IV. 15 *Sequence* diagram *setting* jam. IV-22](#_Toc120345721)

[Gambar IV. 16 *Sequence* diagram tampil data presensi sendiri. IV-22](#_Toc120345722)

[Gambar IV. 17 *Sequence* diagram kelola data presensi *user*. IV-23](#_Toc120345723)

[Gambar IV. 18 *Sequence* diagram isi presensi. IV-23](#_Toc120345724)

[Gambar IV. 19 *Sequence* diagram *filter* presensi. IV-24](#_Toc120345725)

[Gambar IV. 20 *Sequence* diagram *export* presensi *to pdf*. IV-24](#_Toc120345726)

[Gambar IV. 21 *Class* diagram. IV-25](#_Toc120345727)

[Gambar IV. 22 Perancangan antarmuka halaman *login*. IV-28](#_Toc120345728)

[Gambar IV. 23 Perancangan antarmuka halaman *dashboard HRD.* IV-29](#_Toc120345729)

[Gambar IV. 24 Perancangan antarmuka halaman *dashboard* Karyawan. IV-30](#_Toc120345730)

[Gambar IV. 25 Perancangan antarmuka halaman cek lokasi. IV-30](#_Toc120345731)

[Gambar IV. 26 Perancangan antarmuka halaman data kehadiran sendiri. IV-31](#_Toc120345732)

[Gambar IV. 27 Perancangan antarmuka halaman data detail presensi. IV-32](#_Toc120345733)

[Gambar IV. 28 Perancangan antarmuka halaman data pegawai. IV-32](#_Toc120345734)

[Gambar IV. 29 Perancangan antarmuka tambah pegawai. IV-33](#_Toc120345735)

[Gambar IV. 30 Perancangan antarmuka halaman detail pegawai. IV-34](#_Toc120345736)

[Gambar IV. 31 Perancangan antarmuka halaman *edit* pegawai. IV-35](#_Toc120345737)

[Gambar IV. 32 Perancangan antarmuka halaman presensi pegawai. IV-36](#_Toc120345738)

[Gambar IV. 33 Perancangan antarmuka halaman filter presensi pegawai. IV-36](#_Toc120345739)

[Gambar IV. 34 Perancangan antarmuka halaman *export* presensi pegawai IV-37](#_Toc120345740)

[Gambar IV. 35 Perancangan antarmuka halaman atur jam kerja. IV-38](#_Toc120345741)

[Gambar IV. 36 Perancangan antarmuka halaman edit jam kerja IV-38](#_Toc120345742)

[Gambar IV. 37 Perancangan antarmuka halaman *profile*. IV-39](#_Toc120345743)

[Gambar IV. 38 Implementasi halaman *login.* IV-40](#_Toc120345744)

[Gambar IV. 39 Implementasi halaman *dashboard HRD* IV-41](#_Toc120345745)

[Gambar IV. 40 Implementasi halaman *dashboard* karyawan IV-42](#_Toc120345746)

[Gambar IV. 41 Implementasi halaman cek lokasi IV-43](#_Toc120345747)

[Gambar IV. 42 Implementasi halaman data kehadiran sendiri IV-44](#_Toc120345748)

[Gambar IV. 43 Implementasi halaman detail presensi IV-45](#_Toc120345749)

[Gambar IV. 44 Implementasi halaman data pegawai IV-45](#_Toc120345750)

[Gambar IV. 45 Implementasi halaman tambah pegawai IV-46](#_Toc120345751)

[Gambar IV. 46 Implementasi halaman detail pegawai IV-46](#_Toc120345752)

[Gambar IV. 47 Implementasi halaman edit pegawai IV-47](#_Toc120345753)

[Gambar IV. 48 Implementasi halaman presensi pegawai IV-48](#_Toc120345754)

[Gambar IV. 49 Implementasi halaman *filter* presensi pegawai IV-49](#_Toc120345755)

[Gambar IV. 50 Implementasi halaman *export* presensi pegawai IV-50](#_Toc120345756)

[Gambar IV. 51 Implementasi halaman atur jam kerja IV-50](#_Toc120345757)

[Gambar IV. 52 Implementasi halaman edit jam kerja IV-51](#_Toc120345758)

[Gambar IV. 53 Implementasi halaman *profile* IV-51](#_Toc120345759)

[Gambar IV. 54 Implementasi *code controller* presensi IV-52](#_Toc120345760)

[Gambar IV. 55 Implementasi *code controller admin* IV-52](#_Toc120345761)

[Gambar IV. 56 Implementasi *code controller auth* IV-53](#_Toc120345762)

[Gambar IV. 57 Implementasi *code controller* jam IV-53](#_Toc120345763)

[Gambar IV. 58 Implementasi *code model* presensi IV-54](#_Toc120345764)

[Gambar IV. 59 Implementasi *code model auth* IV-54](#_Toc120345765)

[Gambar IV. 60 Implementasi *code model* jam IV-55](#_Toc120345766)

[Gambar IV. 61 Implementasi *code model user* IV-55](#_Toc120345767)

# DAFTAR TABEL

[Tabel I. 1 Waktu Pelaksanaan I-4](#_Toc120346602)

[Tabel IV. 1 Identifikasi hak akses……………………………………………...IV-1](#_Toc120346617)

[Tabel IV. 2 Kebutuhan fungsional HRD IV-2](#_Toc120346618)

[Tabel IV. 3 Kebutuhan fungsional Karyawan IV-3](#_Toc120346619)

[Tabel IV. 4 Skenario use case login IV-4](#_Toc120346620)

[Tabel IV. 5 Skenario use case kelola data user IV-5](#_Toc120346621)

[Tabel IV. 6 Skenario use case setting jam presensi IV-6](#_Toc120346622)

[Tabel IV. 7 Skenario use case isi presensi IV-6](#_Toc120346623)

[Tabel IV. 8 Skenario use case cek lokasi IV-7](#_Toc120346624)

[Tabel IV. 9 Skenario use case tampil data presensi sendiri IV-7](#_Toc120346625)

[Tabel IV. 10 Skenario use case kelola data presensi user IV-8](#_Toc120346626)

[Tabel IV. 11 Skenario use case export data presensi ke pdf IV-9](#_Toc120346627)

[Tabel IV. 12 Skenario use case filter presensi user IV-9](#_Toc120346628)

[Tabel IV. 13 Keterangan diagram activity login IV-10](#_Toc120346629)

[Tabel IV. 14 Keterangan diagram activity kelola data user IV-11](#_Toc120346630)

[Tabel IV. 15 Keterangan diagram activity setting jam IV-12](#_Toc120346631)

[Tabel IV. 16 Keterangan diagram activity isi presensi IV-13](#_Toc120346632)

[Tabel IV. 17 Keterangan diagram activity cek lokasi IV-15](#_Toc120346633)

[Tabel IV. 18 Keterangan diagram activity tampil data presensi sendiri IV-16](#_Toc120346634)

[Tabel IV. 19 Keterangan diagram activity kelola data presensi user IV-17](#_Toc120346635)

[Tabel IV. 20 Keterangan diagram activity export data presensi ke pdf IV-18](#_Toc120346636)

[Tabel IV. 21 Keterangan diagram *Activity filter* presensi *user* IV-19](#_Toc120346637)

[Tabel IV. 22 Peracangan Tabel *User* IV-26](#_Toc120346638)

[Tabel IV. 23 Peracangan Tabel *User* IV-27](#_Toc120346639)

[Tabel IV. 24 Peracangan Tabel Jam IV-27](#_Toc120346640)

[Tabel IV. 25 Pengujian Sistem IV-56](#_Toc120346641)

[Tabel IV. 26 Hasil pengujian *login* IV-60](#_Toc120346642)

[Tabel IV. 27 Hasil pengujian isi presensi IV-61](#_Toc120346643)

[Tabel IV. 28 Hasil pengujian cek lokasi IV-61](#_Toc120346644)

[Tabel IV. 29 Hasil pengujian data kehadiran sendiri IV-62](#_Toc120346645)

[Tabel IV. 30 Hasil pengujian kelola data pegawai IV-62](#_Toc120346646)

[Tabel IV. 31 Hasil pengujian kelola data presensi *user* IV-63](#_Toc120346647)

[Tabel IV. 32 Hasil pengujian *filter* presensi *user* IV-63](#_Toc120346648)

[Tabel IV. 33 Hasil pengujian *export* data presensi ke *pdf* IV-64](#_Toc120346649)

[Tabel IV. 34 Hasil pengujian *setting* jam presensi IV-64](#_Toc120346650)

[Tabel IV. 35 Hasil pengujian *profile* IV-65](#_Toc120346651)

# DAFTAR SINGKAT DAN ISTILAH

|  |  |
| --- | --- |
| **Singkatan** | **Kepanjangan** |
| RAD | *Rapid Application Development* |
| UML | *Unified Modelling Language* |
| GPS | *Global Positioning System* |
| CI | *Codeigniter* |
| PHP | *Perl Hypertext Preprocessor* |
| OOP | *Object Oriented Programming* |

# DAFTAR SIMBOL

1. **Simbol – Simbol Dalam *Usecase* Diagram**

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Deskripsi** |
| **1.** | ***Use case*** | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *use case*. |
| **2.** | **Aktor / *Actor*** | Mempresentasikan seseorang atau sesuatu (seperti perangkat,sistem lain) yang berinteraksi dengan sistem. *Actor* hanya berinteraksi dengan *use case* tetapi tidak memiliki kontrol atas *use case*. |
| **3.** | **Assosiasi / *association*** | Komunikasi antara *actor* dan *Use Case* yang berpartisipasi pada *Use Case* atau *Use Case* memiliki interaksi dengan *actor*. |
| **4.** | **Extensi/*extend*** | Relasi *Use Case* tambahan ke sebuah *Use Case* dimana *Use Case* yang ditambahkan dapat berdiri sendiri walau tanpa *Use Case* tambahan itu, mirip dengan prinsip *in heritance* pada pemrograman berorientasi objek, biasanya *Use Case* tambahan memiliki nama depan yang sama dengan *Use Case* yang ditambahkan, misal Arah panah mengarah pada *Use Case* yang ditambahkan, biasanya *Use Case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *Use Case* yang menjadi induknya. |
| **5.** | **Generalisasi/ *generalization*** | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *Use Case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  arah panah mengarah pada *Use Case* yang menjadi generalisasinya (umum). |
| **6.** | **Menggunakan /*include* / *uses*** | Relasi *Use Case* tambahan ke sebuah *Use Case* dimana *Use Case* yang ditambahkan memerlukan *Use Case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *Use Case* ini. Ada dua sudut pandang yang cukup besar mengenai *include* di *Use Case* :  - *Include* berarti *Use Case* yang ditambahkan akan selalu di panggil saat *Use Case* tambahan dijalankan, misal pada kasus berikut :    - *Include* berarti *Use Case* yang tambahan akan selalu melakukan pengecekan apakah *Use Case* yang di tambahkan telah dijalankan sebelum *Use Case* tambahan di jalankan, misal pada kasus berikut :    Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan. |

**Sumber : Rosa A.S dan M. Shalahudin 2014:156**

1. **Simbol – Simbol Dalam *Class* Diagram**

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Deskripsi** |
| **1.** | **Class** | Kelas pada struktur sistem |
| **2.** | **Antarmuka/*interface*** | Sama dengan konsep *interface* dalam pemrograman berorientasi objek |
| **3.** | **Asosiasi/*association*** | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |
| **4.** | **Asosiasi berarah/ *directed association*** | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicit* |
| **5.** | **Generalisasi** | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus) |
| **6.** | **Kebergantungan/*dependensi*** | Relasi antar kelas dengan makna kebergantungan antar kelas |
| **7.** | **Agrasi / *aggregation*** | Relasi antar kelas dengan makna semua-bagian (*whole-part*) |

**Sumber : Rosa A.S dan M. Shalahudin 2014:146**

1. **Simbol – Simbol Dalam *Sequence* Diagram**

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Deskripsi** |
| **1.** | **Aktor**    **Atau**    **Tanpa Waktu Aktif** | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor. |
| **2.** | **Garis hidup/ *lifeline*** | Menyatakan kehidupan suatu objek |
| **3.** | **Waktu aktif** | Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubungdengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya    Maka cek Status*Login*() dan open() dilakukan didalam metode *login*(). Aktor tidak memiliki waktu aktif |
| **4.** | **Pesan tipe *create*** | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| **5.** | **Pesan tipe *call*** | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,    Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi |
| **6.** | **Pesan tipe *sand*** | Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim |
| **7.** | **Pesan tipe *returm*** | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| **8.** | ***Boundary*** | Berupa tepi dari sistem, seperti user interface atau suatu alat yang berinteraksi dengan sistem yang lain. |
| **9.** | ***Control*** | *Element* mengatur aliran dari informasi untuk sebuah skenario. Objek ini umumnya mengatur perilaku dan perilaku bisnis. |
| **10.** | **Entitas** | Elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau model object. |

**Sumber: Rosa A.S dan M. Shalahudin 2014:165**

1. **Simbol – Simbol Dalam *Activity* Diagram**

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Deskripsi** |
| **1.** | **Status Awal** | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| **2.** | **Aktivitas** | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| **3.** | **Percabangan/*decision*** | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| **4.** | **Penggabungan/*join*** | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| **5.** | **Status Akhir** | Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| **6.** | ***Swimlane*** | Memisahkan organisasi bisnis yang bertanggunga jawab terhadap aktivitas yang terjadi. |

**Sumber: Rosa A.S dan M. Shalahudin 2014:162**

# BAB I PENDAHULUAN

## Latar Belakang

Perkembangan teknologi informasi dan komunikasi pada saat ini sudah semakin maju. Teknologi komputer baik dari segi *hardware* (perangkat keras) ataupun *software* (perangkat lunak) kini sudah banyak mengalami perubahan dan inovasi. Dengan bermodalkan kecanggihan tersebut, apapun dapat dilakukan dengan komputer ataupun gadget terlebih lagi jika perangkat terhubung ke dalam jaringan. Dengan hadirnya jaringan internet, apapun dapat dilakukan secara *real time*, sehingga membantu kita untuk selalu terhubung.

Penggunaan teknologi juga sangat bermanfaat pada kehidupan sehari-hari. Semuanya menjadi lebih mudah dengan hadirnya inovasi baru yang terus dikembangkan, terlebih pada zaman yang sudah modern ini, baik di kalangan anak-anak sampai dengan kalangan dewasa, hampir semuanya pasti sudah memiliki gadget dan sudah mengenal jaringan internet. Semua pekerjaan akan lebih mudah dilakukan dengan bantuan gadget yang serba canggih.

Presensi merupakan suatu dokumen yang berisi tentang daftar hadir seseorang yang bertujuan untuk mencatat ataupun merekam jam hadir untuk keperluan tertentu. Presensi biasanya digunakan pada lingkungan pekerjaan, sekolah, ataupun pada event tertentu. Jenis-jenis presensi sangat beragam, mulai dari presensi catatan tangan yang dicatat secara manual, presensi *fingerprint*, ataupun presensi dengan menggunakan sistem. Namun pada dasarnya, tujuannya sama yaitu untuk memantau serta mencatat daftar kehadiran.

Metode presensi dengan menggunakan sistem perlu suatu cara untuk melakukan verifikasi bahwa karyawan tersebut memang benar berada di lingkungan perusahaan atau sedang *WFH (Work From Home)* saat melakukan presensi. Hal ini dilakukan agar karyawan terdeteksi lokasinya sehingga akan tau posisi karyawan tersebut apakah bekerja di perusahaan ataupun di rumah / *work from home* (*WFH*). Salah satu cara untuk memverifikasi keberadaan karyawan adalah menggunakan bantuan *GPS (Global Positioning System).*

Menurut (Yuniati et al., 2017:151) pada penelitian mereka yang berjudul "Implementasi Modul *Global Positioning System (GPS)* pada *Sistem Tracking Bus Rapid Transit (BRT)* Lampung Menuju *Smart Transportation*", *GPS (Global Positioning System)* adalah suatu sistem navigasi yang menggunakan sinyal satelit untuk melakukan triangulasi posisi yang hendak ditentukan. *GPS* akan menghasilkan output berupa data posisi koordinat lintang *(lattitude)* dan data posisi koordinat bujur *(longitude).*

CV. Asian Technology Inspira merupakan perusahaan yang bergerak dibidang Teknologi, untuk jasanya yaitu jasa pembuatan *Website*. CV. Asian Technology berdiri dari tahun 2012 dan berlokasi di jalan Buah Batu kabupaten Bandung.

Permasalahan yang muncul pada CV. Asian Technology Inspira berkaitan dengan presensi secara manual, di mana presensi dilakukan melalui aplikasi *chat* *Telegram*. Saat akan isi presensi di aplikasi *Telegram,* karyawan dapat mengatur *maps* posisi markernya, sehingga ada celah manipulasi pada titik lokasinya. Selain itu bagian HRD harus mencatat laporan presensi pada tiap-tiap karyawan setiap bulannya, tentunya ini sangat tidak efesien karena sangat menghabiskan waktu dan tenaga.

Untuk dapat mengatasi permasalahan tersebut, CV. Asian Technology Inspira memerlukan sebuah cara baru ataupun cara yang lebih efisien untuk melakukan presensi, salah satunya adalah dengan melakukan presensi melalui sistem. Sistem ini diharapkan mampu menggantikan teknologi Telegram yang masih manual. Dengan hadirnya sistem baru ini, setiap karyawan yang ingin melakukan presensi hanya perlu melakukannya pada gadget masing-masing.

Berdasarkan uraian diatas, peneliti ingin melakukan pengembangan suatu perangkat lunak / *software* sistem presensi. Sistem ini dirancang dengan memanfaatkan teknologi GPS (Global Positioning System), agar setiap karyawan dapat melakukan presensi di lokasi kantor, *Work From Home (WFH)*, sakit, dan cuti. Sistem ini nantinya juga dapat mempermudah HRD untuk mengolah data presensi. Jika sebelumnya setiap pagi HRD membuat laporan presensi untuk menyimpan data presensi, kini sudah tidak perlu melakukan hal itu lagi karena data presensi sudah dapat langsung diakses langsung dari komputernya, ataupun dari gadgetnya, kapan saja dan di mana saja. Sistem ini juga dilengkapi fitur-fitur yang dapat membantu pekerjaan HRD dalam mengolah data, salah satunya dapat mencetak data absensi. Oleh karena itu, penyusun mengangkat judul “Aplikasi Web Presensi Berbasis GPS” untuk penelitian ini.

## Maksud dan Tujuan

Berdasarkan penjabaran diatas, maka maksud dan tujuan penelitian ini adalah:

1. Mengembangkan sistem presensi karyawan untuk mendigitalisasikan proses bisnis yang ada di CV. Asian Technology Inspira.
2. Melakukan inovasi dalam metode presensi karyawan melalui sistem berbasis website untuk dapat lebih memudahkan HRD dalam mengolah data presensi secara real time.
3. Memberi kemudahan bagi karyawan pengguna sistem untuk melakukan presensi

## Ruang Lingkup

Ruang lingkup pada aplikasi Web sistem presensi dengan *GPS* memiliki ruang lingkup sebagai berikut:

1. Aplikasi ini memiliki fitur – fitur presensi, pengolahan data karyawan, *setting* jam masuk dan jam keluar, pengolahan presensi karyawan, mendeteksi lokasi, mengisi presensi masuk dan pulang, *filter* data presensi dan meng-*export* data presensi ke *pdf*.
2. Aplikasi ini sudah mendukung *mobile friendly*.
3. Tidak membahas keamanan aplikasi

## Keluaran dan kerja praktik

Adapun keluaran dari laporan ini berupa:

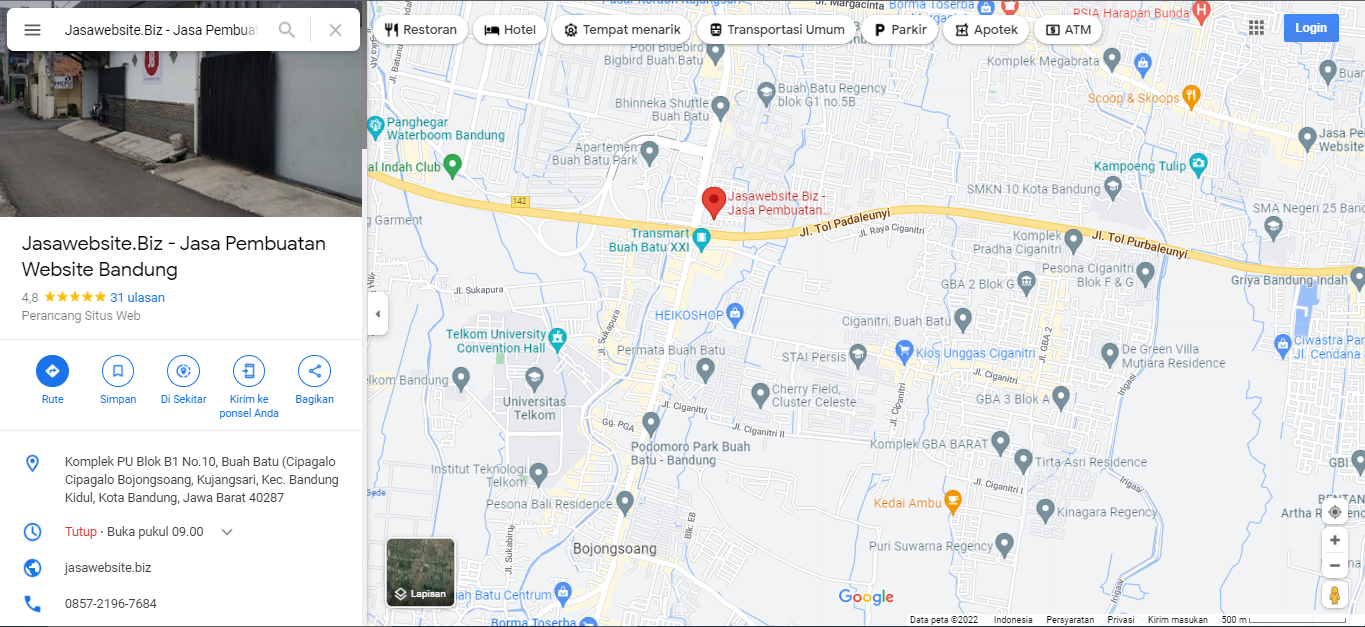
1. Laporan penelitian
2. Aplikasi Web Presensi Berbasis GPS

## Lokasi dan Waktu Pelaksanaan

1. **Lokasi Pelaksanaan**

Kegiatan Kerja praktik yang akan dilaksanakan di Kue Balok Brownies Mahkota yang beralamat di Komplek PU Blok B1 No.10, Buah Batu (Cipagalo Bojongsoang, Kujangsari, Kec. Bandung Kidul, Kota Bandung, Jawa Barat 40287).

Berikut lokasi pada pencarian google maps:



Gambar I. 1 Lokasi Kerja Praktek

1. **Waktu Pelaksanaan**

Waktu Pelaksanaan akan dilaksanakan dalam waktu bulan terhitung dari bulan maret hingga bulan juli 2022

Tabel I. 1 Waktu Pelaksanaan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Uraian Kegiatan | Bulan 1 | | | | Bulan 2 | | | | Bulan 3 | | | |
| 1 | Kebutuhan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Penulisan Bab I |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Pendahuluan |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Penulisan Bab II |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Kajian Pustaka |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Penulisan Bab III |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Analisis sistem dan perancangan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Penulisan Bab IV |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Implementasi Sistem |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Implementasi program, Pengujian program |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Penulisan Bab V |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | Kesimpulan dan Saran |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 | Tahapan Utama Kegiatan | | | | | | | | | | | | |

## **Sistematika Penulisan**

Untuk memberikan gambaran mengenai laporan yang akan dibuat, maka digunakan sistematika penulisan laporan sebagai berikut.

**BAB I PENDAHULUAN**

Pada bab ini berisi Latar Belakang Masalah, Rumusan Masalah, Maksud dan Tujuan, Ruang Lingkup, Keluaran Kerja Praktik dan Sistematika Penulisan.

**BAB II KAJIAN PUSTAKA**

Bab ini berisi tentang teori-teori terkait yang akan digunakan dalam penelitian seperti, teori penelitian, perancangan dan pembuatan sistem serta referensi-referensi yang digunakan menjadi sumber berasal dari jurnal, buku, dan *internet*.

**BAB III PROSES BISNIS DAN ARSITEKTUR SISTEM**

Pada bab ini terdapat proses bisnis sistem yang sedang berjalan dan sistem yang akan datang dibuat sesuai kondisi yang sedang ada di tempat kerja praktik. Arsitektur sistem untuk projek kerja praktik yang berupa perangkat lunak adalah arsitektur perangkat lunak yang dikembangkan dalam pelaksanaan kerja praktik.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini berisi proses pengerjaan penelitian sesuai dengan *model* proses yang digunakan, disajikan pada tiap sub bab. Pada setiap tahap harus disajikan artifak-artifak sesuai dengan permodelan dan *tools* yang dipakai.

**BAB V SIMPULAN DAN SARAN**

Pada bab ini memuat simpulan yang disesuaikan dengan poin-poin tujuan kerja praktik, beserta saran yang bersifat membangun yang berkaitan dengan penelitian ini, untuk pengembangan aplikasi yang lebih baik.

# BAB II KAJIAN PUSTAKA

1. Definisi Sistem

Menurut Romney dan Steinbart (2015:3), sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar. Secara garis besar, sistem merupakan definisi dari segala sesuatu baik berbentuk objek yang abstrak maupun objek yang nyata, yang saling terkait dan menjadi satu kesatuan, serta bekerja sama dengan sedemikian rupa untuk mencapai suatu tujuan tertentu.

1. GPS (Global Positioning System)

*Global Positioning System* atau biasa disebut dengan GPS merupakan suatu sistem navigasi yang memanfaatkan teknologi satelit untuk mendapatkan suatu posisi tertentu secara langsung. Dengan menggunakan alat ini maka dimungkinkan user dapat melacak posisi kendaraan, armada ataupun mobil dalam keadaan *real-time*. Untuk mendapatkan posisi lokasi biasanya mengambil dari *latitude* dan *longtitude* *user-*nya.

Menurut (Milner, 2016:9) dalam penelitiannya yang berjudul *"What is GPS?"*, teknologi GPS bukan hanya digunakan untuk mencari arah, namun dapat juga digunakan untuk berbagai macam hal, yang tentunya berhubungan dengan koordinat posisi, navigasi, dan juga operasi waktu. Teknologi ini juga dapat digunakan di manapun, di seluruh penjuru bumi, dan juga dapat digunakan dalam kondisi cuaca apapun.

Perkasa (2019) dalam penelitiannya yang berjudul *"Penggunaan Global Positioning System (GPS) Untuk Dasar Survey Pada Mahasiswa"* mengemukakan bahwa sistem GPS terbagi menjadi tiga segmen, diantaranya adalah sebagai berikut:

1. Satelit, berfungsi untuk menerima sinyal dan menyimpan segala data yang ditransmisikan oleh stasiun pengontrol. Selain itu, satelit juga memancarkan sinyal dan informasi secara terus menerus ke *receiver* yang dimiliki oleh *user.*
2. Pengontrol, berfungsi untuk mengendalikan satelit dari bumi. Pengontrol dapat melakukan pengecekan terhadap kesehatan satelit, prediksi orbit dan waktu, dan lain sebagainya.
3. Penerima (*receiver*), berfungsi untuk menerima data dari satelit pemancar. Setelah sinyal berhasil ditangkap oleh *receiver*, selanjutnya *receiver* akan memproses data tersebut dan menjadikannya sebagai koordinat posisi.
4. Presensi Karyawan

Menurut Bastian (2007: 117), presensi karyawan adalah suatu kegiatan mendokumentasikan kehadiran karyawan di perusahaan. Setiap hari kerja pegawai diharuskan melakukan presensi pada waktu datang dan pulang, dalam satu periode waktu.

1. *Web Development Tools*

*Web Development Tools* memungkinkan pengembang web menguji dan men-*debug* kode mereka. Mereka dapat menyesuaikan *tools* apa saja yang dibutuhkan saat membuat *website*, agar *website*-nya sesuai dengan yang diharapkan.

Berikut adalah beberapa *Web Development Tools* yang digunakan oleh penulis.

**II.4.1** *Hypertext Markup Language (HTML)*

Menurut Nugroho (2013:5) *HTML* adalah kependekan dari *HyperText Markup Language*, merupakan sebuah bahasa *scripting* yang berguna untuk menuliskan halaman web. Struktur dokumen *HTML* terdiri dari *tag* pembuka dan *tag* penutup. *HTML* versi 1.0 dibangun oleh W3C, dan terus mengalami perkembangan. *HyperText Markup Language* dikenal sebagai bahasa kode berbasis teks untuk membuat sebuah halaman web. Keberadaannya dikenal dengan adanya ekstensi \*.*htm* atau \*.*html.*

**II.4.2 *Hypertext Preprocessor (PHP)***

Kustiyaningsih & Anamisa (2011) berpendapat bahwa *Hypertext Preprocessor (PHP)* adalah *script* yang bersifat *server-side* yang telah dimasukan ke dalam *HTML*. Pada aturannya, *server* akan melakukan proses apabila ada kebutuhan dari *client*. Dalam hal ini *client* menggunakan kode – kode PHP untuk mengirimkan kebutuhan ke *server*.

Prasetio (2014:122) menjelaskan PHP adalah bahasa *script* yang telah ditanam di sisi *server*. PHP adalah bahasa *script* yang didesain untuk web. Aplikasi ini bersifat *server-side* yang artinya harus di-*install* dan ditanamkan di web *server*. Dengan penggunaan PHP, tampilan halaman web menjadi lebih dinamis. Namun jika dilihat lagi struktur dari PHP, kita dapat membangun aplikasi-aplikasi berberbasis web karena PHP menyediakan *interpreter* dalam *high level program language*.

**II.4.3 *Cascading Style Sheet (CSS)***

Menurut Wahyudi (2020), *Cascading Style Sheet (CSS)* adalah suatu bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam *web*, sehingga tampilan *web* akan lebih rapi, terstruktur dan seragam. Selain itu, *HTML* versi 4.0 dapat digunakan untuk menangani masalah tampilan pada *HTML* seperti jenis, ukuran dan warna *font*, posisi teks, batas tulisan atau *margin*, warna *background*, dan sebagainya. Penting untuk diperhatikan adalah cara meletakkan *CSS* dan juga bahasa berbasis *web* lain untuk memudahkan manajemen *file*, *editing*, dan *maintenance*.

**II.4.4 *Javascript (JS)***

Kustiyahningsih dalam Imaniawan (2018) berpendapat bahwa *Javascript* adalah bahasa yang berbentuk kumpulan skrip yang berjalan pada suatu dokumen *HTML*. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap *HTML* dengan mengizinkan pengeksekusian perintah-perintah di sisi *user.* Variabel atau fungsi dengan nama *TEST* berbeda dengan variabel dengan nama *test* dan setiap instruksi diakhiri dengan karakter titik koma (;) di sisi *browser,* bukan di sisi *server web*. *Javascript* adalah bahasa yang “*case sensitive*” artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh karakter titik koma.

Menurut Sunyoto dalam Fachrezi, dkk (2020) *Javascript* adalah bahasa *scripting* yang popular di internet dan dapat bekerja di sebagian besar *browser* popoler seperti *Internet Explorer* (IE), *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan tag *SCRIPT*. Beberapa hal tentang *Javascript*:

1. *Javascript* didesain untuk menambah interaksi suatu *web*.
2. *Javascript* merupakan sebuah bahasa *scripting*.

Bahasa *scripting* merupakan bahasa pemograman yang ringan.

1. *Javascript* berisi baris kode yang dijalankan di komputer (*web browser*).
2. *Javascript* biasanya disisipkan (*embedded)* dalam halaman *HTML*.
3. *Javascript* adalah bahasa *interpreter* (yang berarti skrip dieksekusi tanpa proses kompilasi).
4. Setiap orang dapat menggunakan *Javascript* tanpa membayar lisensi.

**II.4.5 *Web Application* *Framework***

Budi Raharjo (2015) mendefinisikan *Web Application Framework (WAF),* atau biasa disebut dengan *web framework* adalah suatu gabungan *code* berupa pustaka *(library)* dan alat *(tool)* yang dipadupadankan sesuai aturan yang ada menjadi satu kerangka kerja *(framework)* guna mempermudah dan mempercepat kinerja pengembangan aplikasi *web*. Proses pengembangan *web* itu sendiri dapat dikerjakan dengan berbagai macam bahasa pemrograman; seperti *PHP, Python, Ruby, Java* dan lainnya.

Adapun beberapa keuntungan menggunakan *framework* sebagai berikut:

1. Menghemat waktu pengembangan

Dengan struktur *library* yang telah disediakan oleh *framework* maka tidak perlu lagi memikirkan hal-hal tersebut.

1. *Reuse of code*

Dengan menggunakan *framework* maka pekerjaan kita akan memiliki struktur yang baku, sehingga kita dapat menggunakan kembali di proyek-proyek lainnya.

1. Bantuan komunitas

Ada komunitas-komunitas yang siap membantu jika ada permasalahan.

1. Peningkatan kualitas code

Sebuah *framework* merupakan kumpulan *best* *practice* yang sudah teruji. Jadi kita dapat meningkatkan kualitas kode kita.

**II.4.6 *Codeigniter***

Menurut Supriyono dan Muslimah, (2018), *Codeigniter* adalah kerangka kerja pengembangan aplikasi *PHP* berdasarkan arsitektur yang terstruktur. *Codeigniter* memiliki tujuan untuk memberikan alat bantu yang dibutuhkan seperti *helpers and libraries* untuk mengimplementasi tugas yang biasa dilakukan. Dengan demikian, pengembangan proyek menjadi lebih mudah dan cepat dan pengembang tidak perlu menulis lagi dari awal.

*Codeigniter* menjadi sebuah *framework* *PHP* dengan model *MVC* (*Model*, *View*, *Controller*) untuk membangun *website* dinamis dengan menggunakan *PHP* yang dapat mempercepat pengembang untuk membuat sebuah aplikasi *web*. Selain ringan dan cepat, *Codeigniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih *Codeigniter* sebagai *framework* pilihannya. *Codeigniter* pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis.

Adapun keunggulan *Codeigniter,* diantaranya:

1. Ukuran yang kecil

Keseluruhan *framework* *Codeigniter* hanya sebesar 2MB. Ukuran sekecil ini tentu memudahkan kita untuk mempelajari *Codeigniter*, menjalankannya, juga melakukan *update*.

1. Sangat cepat

Sebagai pengguna, kita cenderung menyukai aplikasi yang cepat, baik saat *loading* atau saat menggunakannya. *Codeigniter* memiliki daya respon *loading* kurang dari 50 milisecond.

1. Fleksibel

*Codeigniter* memiliki komponen dan fitur-fitur yang berdiri sendiri tanpa terlalu banyak bergantung pada satu sama lain, membuat *Codeigniter* sangat mudah untuk di-*upgrade* dan diperbaiki jika ada masalah.

1. Arsitektur yang bersifat *Model-View-Controller* (*MVC*)

Ketika membuat dan mengerjakan aplikasi berbasis *web*, desain arsitektur *MVC* merupakan standar industri untuk sebuah *framework*. *MVC* sangat berguna untuk memisahkan data, *business logic*, dan presentasi.

1. Dapat dimodifikasi

Pada mulanya *Codeigniter* sudah terdapat *library* dan bantuan dari awal mula kita pakai. Jika apa yang *programmer* cari tidak ada, *programmer* dapat dengan mudah menambahkan *library* dan bantuan *programmer* sendiri.

1. Mudah dipelajari dan dikuasai

Bagi pengguna yang sudah menguasai *PHP*, *Codeigniter* adalah *framework* yang sangat mudah untuk dipelajari. Dalam waktu yang singkat, pengguna dapat menciptakan dan mengembangkan aplikasi menggunakan *Codeigniter.*

**II.4.7 *MySQL***

*My Structured Query Language* (MySQL) merupakan salah satu aplikasi *Database Management System* (DBMS) yang mampu menjadi penerima dan pengirim suatu data dengan begitu cepat. Perintah dasar MySQl terdiri dari *insert, update, delete*, dan *select*. MySQL sendiri bersifat *open source* sehingga dapat dimanfaatkan secara gratis. Pemrograman PHP juga sangat mendukung adanya *Database* MySQL. Dijelaskan oleh Rosa & Shalahuddin (2014), *My*SQL adalah salah satu bahasa yang dimanfaatkan untuk mengatur suatu data pada *Relational Database Management System* (RDBMS).

Raharjo dkk (2014) mendefinisikan bahwa “MySQL merupakan RDBMS (*server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menghimpun data dalam jumlah sangat banyak, dan dapat diakses oleh banyak *user*”.

Berikut ini beberapa karakteristik dari *MySQL*:

1. Portabilitas: *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga,* dan masih banyak lagi.
2. Perangkat lunak sumber terbuka: *MySQL* didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi *GPL* sehingga dapat digunakan secara gratis.
3. *Multi-user*: *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*: *MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.
5. Ragam tipe data: *MySQL* memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan fungsi: *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah *(query).*
7. Keamanan: *MySQL* memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan: *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas: *MySQL* dapat melakukan koneksi dengan klienmenggunakan protokol *TCP/IP, Unix soket (UNIX),* atau *Named Pipes (NP)*.
10. Lokalisasi: *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antarmuka: *MySQL* memiliki antarmuka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *Application Programming Interface (API*).
12. *Client* dan peralatan: *MySQL* dilengkapi dengan berbagai peralatan (*tools*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online.*
13. Struktur tabel: *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABEL*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.

Tujuan menggunakan *MySQL* adalah agar dapat mempermudah *developer* dalam mengembangkan dan mengelola suatu data. Dengan menggunakan *MySQL*, juga bisa diintegrasikan untuk *mobile apps* dan *desktop application* untuk mengembangkan sistem ke depannya.

**II.4.8 *Bootstrap***

Menurut Husein Alatas (2013), *Bootstrap* merupakan *framework* untuk membangun desain *web* secara *responsif*. Artinya, tampilan *web* yang dibuat oleh *bootstrap* akan menyesuaikan ukuran layar dan *browser* yang kita gunakan baik di *desktop*, tablet ataupun *mobile device*. Dengan *bootstrap* kita juga bisa membangun *web dinamis* ataupun *statis*.

**II.4.9 *Leaflet Js***

Menurut Fera Meliyanti (2015), *Leaflet Js* merupakan library *javascript* *open source* yang berguna untuk membangun aplikasi peta interaktif berbasis web. *Leaflet* support dengan *platform* *mobile* dan *platform* *desktop*, *HTML5* dan *CSS3,* serta *OpenLayer* dan *Google Maps API* yang merupakan library *javascript* untuk membangun aplikasi peta yang sangat popular saat ini.

Dengan memanfaatkan *leaflet*, *developer* yang tidak memiliki latar belakang GIS pun dapat dengan mudah menampilkan peta interaktif berbasis web pada *server*. *Leaflet* mampu menampilkan layer dari file *geojson*, memberi *style* dan membuat *layer* yang interaktif seperti menampilkan *marker* yang menampilkan *popup* informasi ketika di-klik. Pada intinya, *Leaflet JS* adalah sebuah *library* *javascript* yang dibangun untuk dapat membuat sebuah *platform* peta berbasis *open source*. Berikut ini yang dapat dilakukan dengan *Leaflet JS*:

* 1. *Leaflet JavaScript library* memungkinkan kita menggunakan lapisan/ *layer* seperti *Tile layers, WMS, Markers, Popups, Vector layers (polylines, polygons, circles, dll.), Image overlays* dan *GeoJSON.*
  2. *User* juga dapat berinteraksi dengan peta *Leaflet js* dengan menggeser peta, memperbesar (*zoom*) dengan mengklik dua kali atau menggulir mouse, menggunakan *user keyboard*, *event handling*, dan dengan menyeret *marker* (penanda).
  3. *Leaflet.js* mendukung browser seperti *Chrome, Firefox, Safari 5+, Opera 12+, IE 7–11 di desktop* dan browser *seperti Safari, Android, Chrome, Firefox* untuk ponsel.

1. ***Database Management System (DBMS****)*

Menurut Hesananda dalam Mulyawan (2018),*Database Management System (DBMS)* adalah kumpulan program yang digunakan untuk mendefinisikan, mengatur, dan memproses *database*. Ketika ingin mengoperasikan *DBMS*, maka *user* harus menggunakan bahasa komputer yang memang sesuai dengan sistem terkait. Terdapat dua jenis bahasa komputer yang disarankan, agar bisa berkomunikasi melalui *DBMS*, yaitu:

1. *Data Definition Language* (*DDL*)

Digunakan untuk menggambarkan desain dari basis data secara keseluruhan, mulai dari pembuatan tabel baru, memuat atau me-*load* *indeks*, dan juga mengubah tabel.

1. *Data Manipulation Language* (*DML*)

Digunakan untuk memanipulasi *database* seperti menambah data, mengubah data dan menghapus data

1. *Website*

Wibowo dkk (2015) mengemukakan bahwa *website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, suara, dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Dalam *website* dikenal sebutan *website* dinamis dan *website* statis.

1. *Website* statis

*Website* statis adalah *website* yang mempunyai halaman konten yang tidak berubah-ubah.

1. *Website* dinamis

*Website* dinamis merupakan *website* yang secara struktur ditujukan untuk *update* sesering mungkin.

1. *Hierarchical Model View Controller (HMVC)*

Hakim, (2021) mengemukakan bahwa *Hierarchical Model View Controller (HMVC)* merupakan konsep pengembangan suatu aplikasi berbasis modular dengan versi pengembangan dari *Design Patern MVC (Model View Controller)* yang diimplementasikan secara hirarkis. Metode *HMVC* umumnya membuat sebuah aplikasi dengan memisahkan data *(Model)* dari tampilan *(View)* dan cara bagaimana memprosesnya *(Controller).* Dalam hal ini, *HMVC* benar-benar hanya produk sampingan alami untuk meningkatkan modularitas kode program, dan *re-usability.*

Adapun beberapa manfaat utama menggunakan *HMVC* diantaranya:

1. Mengurangi ketergantungan antara bagian-bagian yang berbeda dari program.
2. Mendorong penggunaan kembali kode, komponen, dan modul.
3. mempermudah pemeliharaan aplikasi, karena script atau kode yang lebih rapi dan terstruktur sehingga mempermudah developer dalam proses pemeliharaan aplikasi
4. Rekayasa Perangkat Lunak

Sri, (2016) mengemukakan bahwa Rekayasa Perangkat Lunak adalah sebuah bidang yang dilakukan oleh seorang perekayasa perangkat lunak yang berkaitan dengan pembuatan dan pemeliharaan aplikasi perangkat lunak dengan mererapkan teknologi dan praktik dari ilmu komputer, manajemen proyek, dan bidang bidang lainnya.

Rekayasa perangkat lunak dilakukan untuk menghasilkan sistem yang baik, efektik, dan efisien untuk perangkat lunak. Maksud dari ini, sebelum membuat perangkat lunak harus menggunakan prinsip kaidah - kaidah proses rekayasa pada perangkat lunak.

Kriteria yang dapat digunakan sebagai prinsip dalam merekayasa perangkat lunak:

1. Dapat terus dirawat dan dipelihara (Maintainability)
2. Dapat mengikuti perkembangan teknologi (Dependability)
3. Dapat mengikuti keinginan pengguna (Robust)
4. Efektif dan efisien dalam menggunakan energi dan penggunaannya
5. Dapat memenuhi kebutuhan yang diinginkan (Usability)

Rekayasa perangkat lunak mencakup semua bagian siklus hidup pengembangan perangkat lunak, termasuk analisis, desain, pengembangan, pengujian, integrasi, implementasi, pemeliharaan, dan bahkan penghentian.

1. *Rapid Application Development (RAD)*

Menurut Sukamto & Shalahudin (2013), *Rapid Application Development* (RAD) adalah *model* proses pengembangan perangkat lunak *incremental* terutama untuk waktu pengerjaan yang pendek. Adapun skema *model* pengembangan *RAD* dapat dilihat pada gambar II.1.



Gambar II. 1 Skema metode pengembangan perangkat lunak RAD (Kendall, 2010).

Menurut Kendall (2010), dalam melakukan pengembangan perangkat lunak dengan menggunakan metode *RAD*, ada tiga fase yang harus dilalui yaitu:

* + - 1. *Requirements planning* (perencanaan syarat-syarat).

*User* dan *analyst* melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.

* + - 1. Proses Desain Sistem *(Design System)*.

Pada tahap ini keaktifan *user* yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Seorang *user* dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data dan yang lain.

* + - 1. *Implementation* (implementasi).

Tahapan ini adalah tahapan *programmer* yang mengembangkan desain suatu program yang telah disetujui oleh *user* dan *analyst*. Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini *user* biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut.

# BAB III PROSES BISNIS DAN ARSITEKTUR SISTEM

1. Proses Bisnis

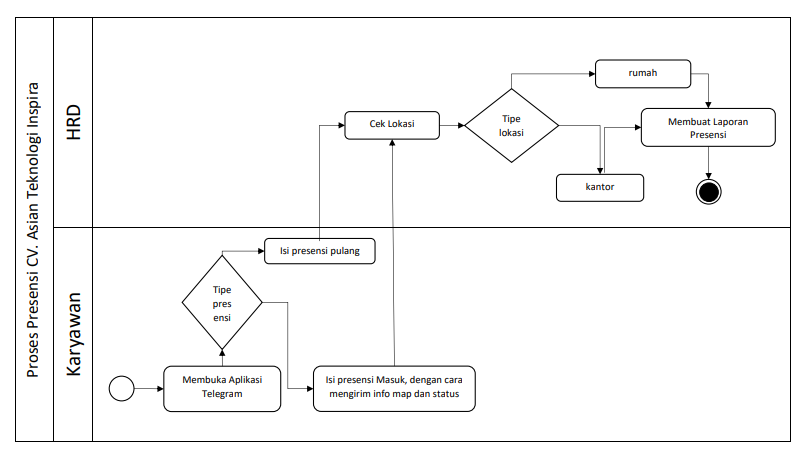
Pada saat ini industri 5.0 semakin canggih dan maju, di mana sebuah pertumbuhan sektor industri di Indonesia sedang gencar-gencarnya dalam mendongkrak perteknologian di era pandemi saat ini. Salah satu teknologi yang pada saat ini berkembang yaitu dunia *aplikasi online*. Aplikasi presensi berbasis *GPS* di perusahaan mulai dilirik dan sudah mulai dipakai perusahaan – perusahaan lainnya, karena aplikasi ini memiliki fungsi yang sangat baik dan akurat. Ini menjadi sebuah potensi dan peluang yang besar bagi perusahaan – perusahaan dalam meningkatkan kinarja yang berkualitas, dan *marketing* untuk memompa dan memberikan nilai jual lebih suatu aplikasi masing-masing produsen dalam negri.

Aplikasi Web Presensi berbasis *GPS* merupakan sebuah aplikasi yang dibutuhkan di perusahaan, turut meningkatkan kemajuan pada perusahaan. Pada dasarnya CV.Asian Technology Inspira melakukan presensi dengan menggunakan bantuan aplikasi *Telegram*, kemudian *HRD* mencatat laporan presensi karyawan ini secara manual. Oleh karena itu penulis membuat bagaimana cara proses presensi ini lebih efektif dan efisien. CV.Asian Technology Inspira yang berlokasi di Jalan Komplek PU Blok B1 No.10, Buah Batu (Cipagalo Bojongsoang, Kujangsari, Kec. Bandung Kidul, Kota Bandung, Jawa Barat 40287).

**III.1.1 Proses Bisnis Sistem Berjalan**

Proses bisnis yang berjalan pada presensi karyawan CV. Asian Teknologi Inspira secara manual saat ini adalah

1. Karyawan membuka aplikasi *Telegram.*
2. Melakukan presensi dengan mengirimkan lokasi menggunakan *maps* dan status kehadirannya.
3. *HRD* cek lokasi *maps*
4. Membuat laporan presensi karyawan.

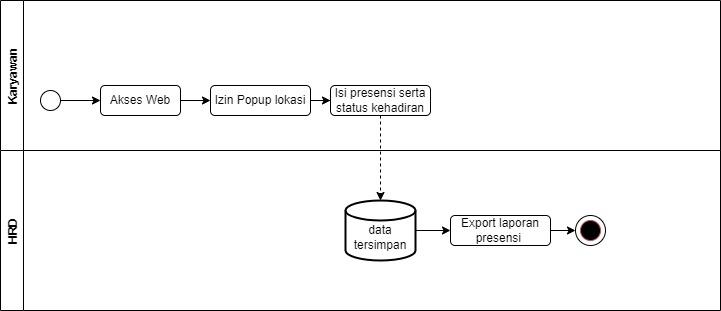


Gambar III. 1 Gambar Proses Bisnis Berjalan

**III.1.2 Proses Bisnis Usulan**

Proses Bisnis Sistem usulan yang akan berjalan pada CV. Asian Teknologi Inspira berbasis website yang akan datang adalah:

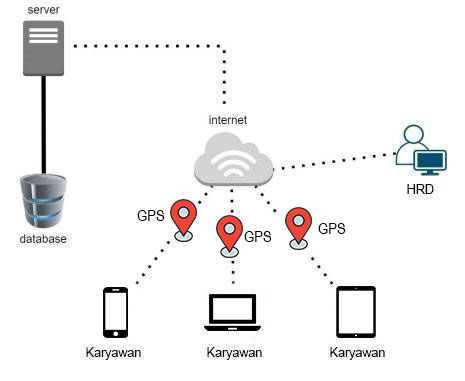
* + 1. Diawali dengan *HRD* *login* terlebih dahulu.
    2. *HRD* melakukan *create user* untuk karyawan.
    3. *HRD* melakukan *setting* jam masuk dan jam pulang.
    4. Karyawan *login* yang tadi sudah dibuatkan akunnya.
    5. Karyawan dan *HRD* melakukan presensi, dengan memilih opsi status kehadirannya dan mengambil lokasinya *Langtitude* dan *Longatitude.*
    6. Setelah mengisi presensi karyawan dapat mengakses menu data presensinya.
    7. *HRD* dapat mengakses menu presensi semua user.
    8. *HRD* bisa *mengexport* presensi ke *PDF* untuk data presensi setiap bulannya pada setiap masing-masing karyawan ataupun *export* semua data presensi karyawan.



*Gambar III. 2* Proses Bisnis Usulan

1. Arsitektur Sistem

*Arsitektur sistem* usulan di CV. Asian Teknologi Inspira adalah menggunakan jaringan internet sebagai media penghubung antara Karyawan dan HRD untuk meminta data ke server dan database kemudian dijelaskan pada gambar.



Gambar III. 3 Gambar Proses Arsitektur Sistem

Berikut tahapan proses interaksi dalam aplikasi:

1. HRD meng-*input* data karyawan dan akan tersimpan di *database*.
2. *HRD* meng-*setting* jam masuk dan jam pulang untuk presensi.
3. Karyawan *login* ke *website* yang tadi akunnya sudah di *input*-kan oleh *HRD.*
4. *HRD* dan karyawan mengisi presensi dan akan tersimpan di *database.*
5. HRD dapat mengakses data presensi semua *user* dan meng-*export* menjadi laporan.
6. *Web* presensi mendukung diberbagai device, *smartphone,* laptop, dll. Asalkan device tersebut memiliki *GPS.*

# BAB IV HASIL DAN PEMBAHASAN

1. Requirement Model
2. Identifikasi Hak Akses

Identifikasi hak akses adalah izin atau hak istimewa yang diberikan kepada pengguna untuk membuat, mengubah, menghapus atau melihat data dalam sebuah aplikasi sebagaimana ditetapkan oleh aturan yang dibuat oleh pemilik data dan sesuai kebijakan keamanan informasi. Identifikasi hak akses dijelaskan pada tabel IV.1

Tabel IV. 1 Identifikasi hak akses

|  |  |  |
| --- | --- | --- |
| ***Level Autorisasi*** | **Sumber Daya Manusia** | |
| **Tingkatan Pengguna** | **Hak Akses** |
| ***HRD*** | *HRD* | Melakukan presensi, melihat lokasi maps sendiri, mengelola data user, mengelola presensi user, melakukan export pdf presensi, setting jam untuk presensi, melihat *profile* |
| **Karyawan** | Karyawan | Melakukan presensi, melihat lokasi maps sendiri, melihat data kehadiran presensi , melihat *profile* |

1. Identifikasi Kebutuhan Fungsional

Identifikasi Kebutuhan Fungsional ini merupakan hal yang lebih rinci lagi daripada identifikasi pengguna perangkat lunak dan juga dapat dimasukan ke dalam kontrak sistem. Kebutuhan fungsional dikembangkan setelah kebutuhan pengguna telah ditetapkan di mana semua fitur dan kemampuan sistem dijelaskan. Berikut ini adalah identifikasi kebutuhan fungsioanl berdasarkan kebutuhan pengguna:

1. **Kebutuhan fungsional *HRD***

Tabel IV. 2 Kebutuhan fungsional HRD

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | Nomor Requirement | Nama Requirement | Keterangan |
| 1 | Req-01 | *Login* | *HRD* dapat *login* melalui website |
| 2 | Req-02 | *Logout* | *HRD* dapat melakukan *logout* pada website |
| 3 | Req-03 | Isi presensi | *HRD* dapat melakukan presensi kehadiran |
| 4 | Req-04 | Cek lokasi | *HRD* dapat melihat posisi lokasinya sendiri |
| 5 | Req-05 | Tampil data presensi sendiri | *HRD* dapat melihat data kehadirannya sendiri setelah melakukan presensi |
| 6 | Req-06 | Kelola data *user* | *HRD* dapat menambahkan, mengedit, dan menghapus data *user* |
| 7 | Req-07 | Kelola data presensi *user* | *HRD* dapat melihat, men-*filter*, dan menghapus data kehadiran *user* |
| 8 | Req-08 | *Setting* jam presensi | *HRD* dapat mengubah jam masuk dan jam pulang waktu presensi |
| 9 | Req-09 | *Export* data presensi ke *pdf* | *HRD* dapat meng-*export* kehadiran *user* berdasarkan nama *user*, bulan, dan tahun atau meng-export semua user ke *pdf* |
| 10 | Req-10 | *Profile* | *HRD* dapat melihat *profile* data sendiri |

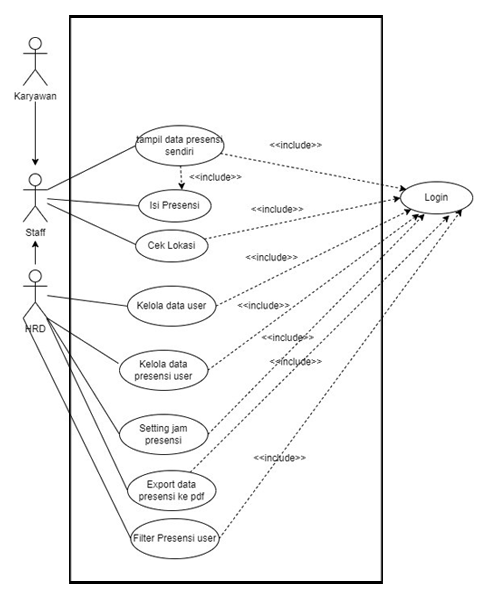
1. **Kebutuhan Fungsional Karyawan**

Tabel IV. 3 Kebutuhan fungsional Karyawan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | Nomor Requirement | Nama Requirement | Keterangan |
| 1 | Req-K01 | *Login* | Karyawandapat login melalui website |
| 2 | Req-K02 | *Logout* | Karyawandapat melakukan *logout* pada website |
| 3 | Req-K03 | Isi presensi | Karyawanmelakukan presensi kehadiran |
| 4 | Req-K04 | Cek lokasi | Karyawandapat melihat posisi lokasinya sendiri |
| 5 | Req-K05 | *Profile* | Karyawandapat melihat *profile* data sendiri |
| 6 | Req-K06 | Tampil data kehadiran sendiri | Karyawandapat melihat data kehadirannya sendiri setelah melakukan presensi |

1. *Use case Model*
2. **Diagram *Usecase***

*Use Case* diagram merupakan tindakan atau langkah – langkah peristiwa yang mendefinisikan interaksi antara aktor dan sistem untuk mencapai tujuan. Berikut merupakan diagram *Use Case* dari Aplikasi Web Presensi Berbasis *GPS.*



Gambar IV. 1 Diagram *use case*

1. **Skenario *Usecase***

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | *Login* |
| **Tujuan** | Akses *HRD* dan karyawan agar dapat mengakses *website* |
| **Deskripsi** | Sistem melakukan autentifikasi terhadap *user* yang masuk |
| ***Actor*** | *HRD,* karyawan |
| **Skenario** | |
| **Kondisi Awal** | Halaman *login* ditampilkan |
| **Aksi *Actor*** | masukan *username* dan *password* kemudia tekan tombol *login* |
| **Reaksi Sistem** | Melakukan pengecekan *username* dan *password* dan mencocokan dengan data *user* di *database* yang telah tersimpan  Jika *username* dan *password* valid maka *user* akan masuk ke halaman *website* utama.  Jika *username* dan *password* salah maka *user* akan dikembalikan ke halaman *login* |
| **Kondisi Akhir** | *HRD* dan karyawan dapat masuk ke dalam *website* |

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | Kelola Data *User* |
| **Tujuan** | *HRD* dapat mengelola data *user,* meliputi tambah *user*, edit *user*, dan *delete user* |
| **Deskripsi** | Akun *user* dibuatkan oleh *HRD* dan *HRD* juga dapat meng-edit dan meng-*delete* akun *user* |
| ***Actor*** | *HRD* |
| **Skenario** | |
| **Kondisi Awal** | *HRD* sudah melakukan *login*. |
| **Aksi *Actor*** | *HRD* mengakses halaman Data Pegawai, lalu menambahkan data *user* baru beserta role-nya *HRD* juga dapat mengedit dan menghapus data *user*. |
| **Reaksi Sistem** | Sistem menyimpan data *user* yang telah *HRD* lakukan (*CRUD*). |
| **Kondisi Akhir** | HRD dapat melihat, mengedit, menghapus daftar *user.* |

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | *Setting* jam presensi |
| **Tujuan** | *HRD* dapat mengatur waktu presensi masuk dan presensi pulang |
| **Deskripsi** | Waktu presensi masuk dan pulang di-set oleh HRD pada waktu sistem berjalan pertama kali, atau ketika ada perubahan kebijakan. Karyawan harus melakukan presensi masuk sebelum waktu awal dan melakukan presensi pulang setelah waktu akhir yang ditetapkan |
| ***Actor*** | *HRD* |
| **Skenario** | |
| **Kondisi Awal** | *HRD* sudah melakukan *login* pada *website* |
| **Aksi *Actor*** | *HRD* mengakses halaman *setting* jam |
| **Reaksi Sistem** | Sistem menampilkan halaman *setting* jam |
| **Kondisi Akhir** | HRD dapat mengedit jam masuk dan jam pulang untuk presensi*.* |

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | Isi presensi |
| **Tujuan** | *Staff* mengisi presensi |
| **Deskripsi** | *HRD* dan karyawan (*staff)* mengisi presensi waktu masuk dan waktu pulang sesuai waktu yang ditetapkan. |
| ***Actor*** | *Staff* |
| **Skenario** | |
| **Kondisi Awal** | *Actor* sudah melakukan *login* pada *website* |
| **Aksi *Actor*** | *Actor* mengakses halaman d*ashboard* |
| **Reaksi Sistem** | Sistem menampilkan halaman *dashboard* |
| **Kondisi Akhir** | *Actor* dapat melakukan isi presensi masuk dan presensi pulang dengan cara mengizinkan dulu lokasinya agar *latitude* dan *longatitude*-nya terdeteksi dan tersimpan ke *database* |

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | Cek lokasi |
| **Tujuan** | *Staff* mengecek lokasi |
| **Deskripsi** | *Staff* melakukan pengecekan lokasinya berada, sebagai acuan saat meng-isi presensi lokasinya sesuai dengan cek lokasi tersebut. |
| ***Actor*** | *Staff* |
| **Skenario** | |
| **Kondisi Awal** | *Actor* sudah melakukan *login* pada *website.* |
| **Aksi *Actor*** | *Actor* mengakses halaman d*ashboard*, lalu akses halaman cek lokasi. |
| **Reaksi Sistem** | Sistem menampilkan halaman cek lokasi. |
| **Kondisi Akhir** | *Actor* dapat melihat posisi lokasinya berada, di mana lokasi tersebut sebagai acuan untuk posisi *maps* saat melakukan presensi. |

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | Tampil data presensi sendiri |
| **Tujuan** | *Staff* dapat melihat data presensi sendiri |
| **Deskripsi** | Setelah mengisi presensi, *staff* dapat melihat semua data kehadirannya sesuai data isi presensi. |
| ***Actor*** | *Staff* |
| **Skenario** | |
| **Kondisi Awal** | *Actor* sudah melakukan isi presensi pada website |
| **Aksi *Actor*** | *Actor* mengakses halaman data kehadiran |
| **Reaksi Sistem** | Sistem menampilkan halaman data kehadiran |
| **Kondisi Akhir** | *Actor* dapat melihat data presensinya sendiri dengan lokasi mapsnya |

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | Kelola data presensi *user* |
| **Tujuan** | *HRD* dapat melihat data presensi *user* |
| **Deskripsi** | Semua presensi data *user,* dapat dilihat oleh *HRD.* |
| ***Actor*** | *HRD* |
| **Skenario** | |
| **Kondisi Awal** | *HRD* sudah melakukan *login* pada website |
| **Aksi *Actor*** | *Actor* mengakses halaman presensi pegawai |
| **Reaksi Sistem** | Sistem menampilkan halaman presensi pegawai |
| **Kondisi Akhir** | *HRD* dapat melihat data presensi semua *user* dengan detailnya dan *HRD* juga dapat menghapus data presensi *user* yang dipilih. |

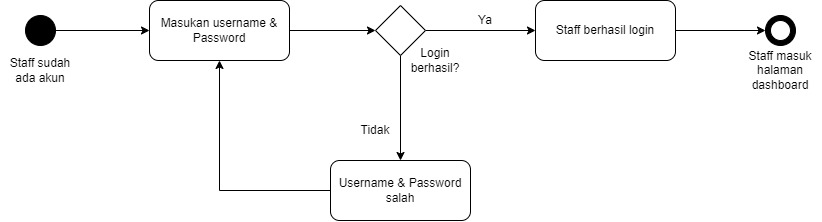
|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | *Export* data presensi ke *pdf* |
| **Tujuan** | *HRD* dapat meng-*export* data presensi ke *pdf* |
| **Deskripsi** | Semua data presensi *user* dapat di *export* ke *pdf* oleh *HRD* |
| ***Actor*** | *HRD* |
| **Skenario** | |
| **Kondisi Awal** | *HRD* sudah melakukan *login* pada *website* |
| **Aksi *Actor*** | *HRD* mengakses halaman presensi pegawai, dan klik tombol *export to pdf* |
| **Reaksi Sistem** | Sistem menampilkan halaman *export pdf* |
| **Kondisi Akhir** | *HRD* mengisi *form* nama pegawai dan periode waktu setelah itu *export* data presensi berhasil. |

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nama** | *Filter* presensi *user* |
| **Tujuan** | *HRD* dapat meng-*filter* presensi *user* |
| **Deskripsi** | Semua data presensi dapat dilakukan *filter* oleh *HRD* |
| ***Actor*** | *HRD* |
| **Skenario** | |
| **Kondisi Awal** | *HRD* sudah melakukan *login* pada *website* |
| **Aksi *Actor*** | *HRD* mengakses halaman presensi pegawai |
| **Reaksi Sistem** | Sistem menampilkan halaman presensi pegawai |
| **Kondisi Akhir** | *HRD* dapat meng-*filter* presensi *user* berdasarkan nama pegawai, bulan dan tahun presensi. |

1. Design Model
2. *Diagram Activity*

*Activity* diagram pada aplikasi akan dijelaskan pada gambar di bawah ini:

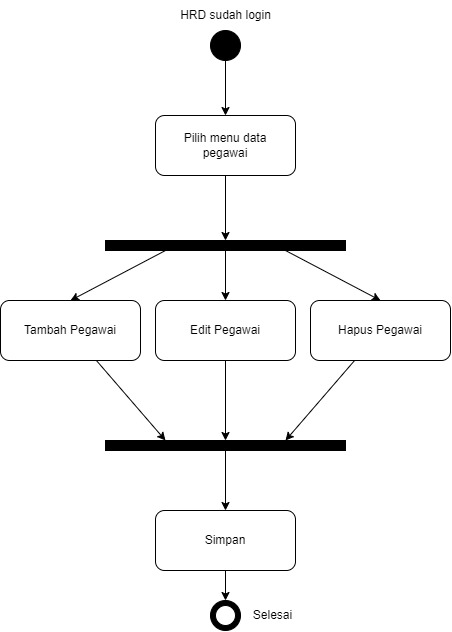
* + - 1. **Diagram *Activity Login***

****

Gambar IV. 2 Diagram activity login

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *Staff* | * Staff sudah ada akun dan mengakses halaman *login.* * Masukan *username* dan *password* pada *form login.* * Validasi, jika *username* dan password benar makan *staff* akan diarahkan ke halaman *dashboard,* dan jika *username* dan *password* salah maka akan dikembalikan ke halaman *login* lagi. |

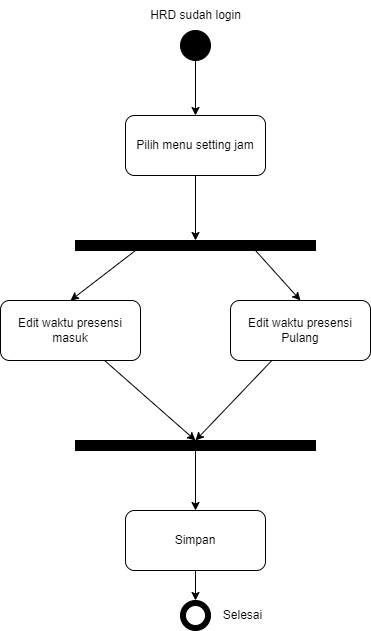
* + - 1. Diagram *Activity* Kelola Data *User*

****

Gambar IV. 3 Diagram *activity* kelola data *user*

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *HRD* | * *HRD* sudah melakukan *login.* * Pilih menu data pegawai*.* * Pilih fungsi yang akan dilakukan, tambah pegawai, edit pegawai, atau hapus pegawai. * Fungsi berhasil disimpan |

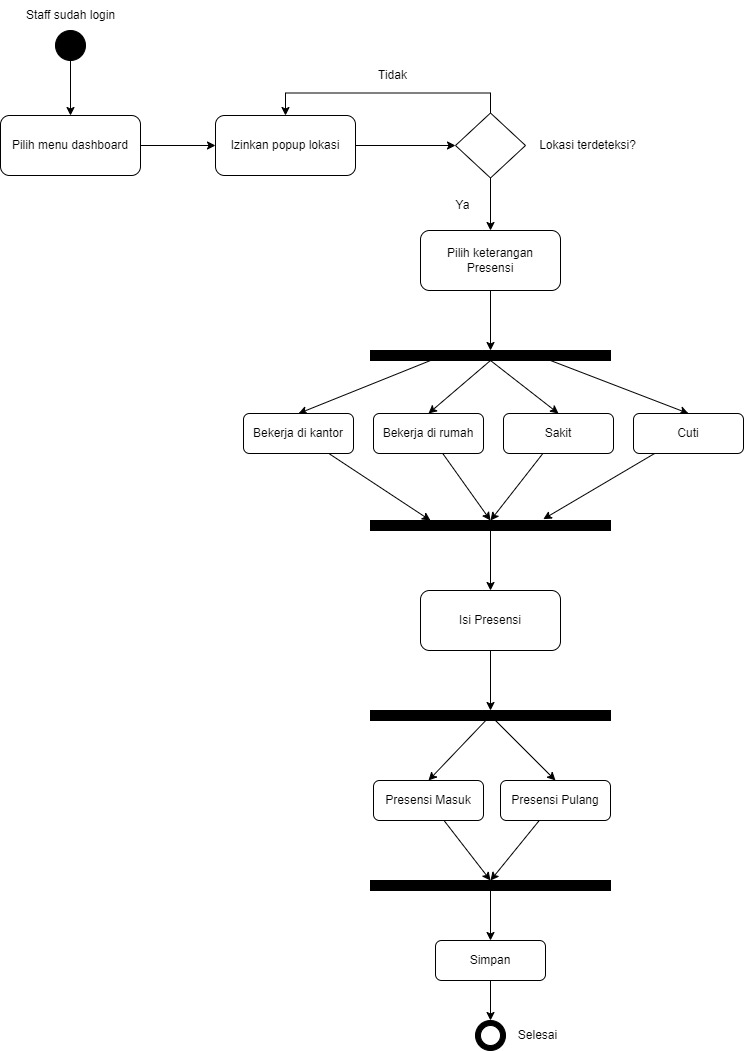
* + - 1. Diagram *Activity Setting* Jam

****

Gambar IV. 4 Diagram *activity setting* jam

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *HRD* | * *HRD* sudah melakukan *login.* * Pilih menu *setting* jam*.* * Pilih fungsi yang akan dilakukan, edit waktu presensi masuk atau edit waktu presensi pulang. * Fungsi berhasil disimpan. |

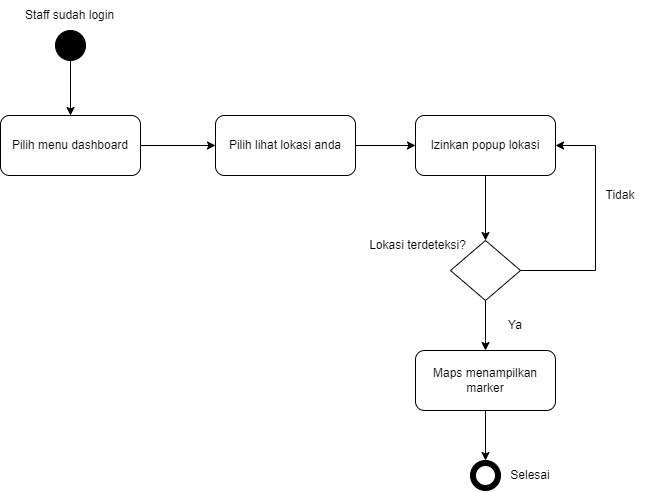
* + - 1. Diagram *Activity* Isi Presensi

****

Gambar IV. 5 Diagram *activity* isi presensi

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *Staff* | * *Staff* sudah melakukan *login.* * Pilih menu *dashboard.* * Izinkan *popup* lokasi untuk mendeteksi *latitude* dan *longitude-*nya, fungsinya agar saat mengisi presensi lokasinya dapat terlihat. * Pilih keterangan presensi diantaranya bekerja di kantor, bekerja di rumah, sakit, dan cuti. * Pilih tipe isi presensi, presensi masuk atau pulang * Presensi berhasil tersimpan |

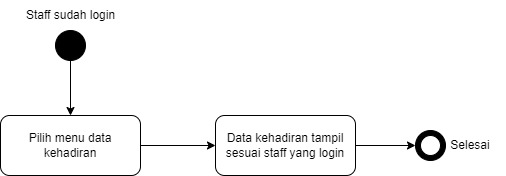
* + - 1. Diagram *Activity* Cek Lokasi

****

Gambar IV. 6 Diagram *activity* cek lokasi

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *Staff* | * *Staff* sudah melakukan *login.* * Pilih menu *dashboard.* * Pilih tombol lihat lokasi anda. * Tampil halaman cek lokasi. * Izinkan *popup* lokasi untuk mendeteksi *latitude* dan *longitude-*nya, fungsinya agar *maps* dapat mendeteksi lokasi. * *Maps* berhasil mendeteksi lokasi. |

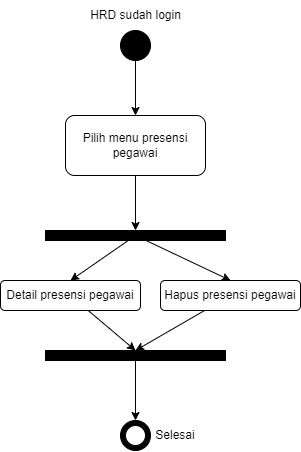
* + - 1. Diagram *Activity* Tampil Data Presensi Sendiri

****

Gambar IV. 7 Diagram *activity* tampil data presensi sendiri

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *Staff* | * *Staff* sudah melakukan *login.* * Pilih menu data kehadiran*.* * Data kehadiran tampil sesuai *staff* yang sedang *login.* |

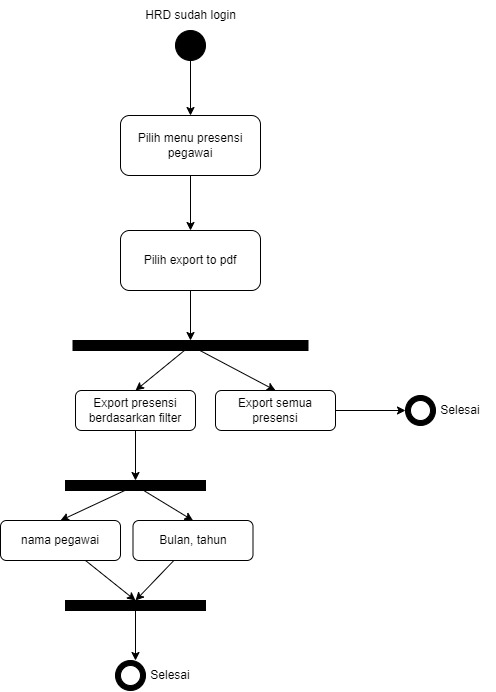
* + - 1. Diagram *Activity* Kelola Data Presensi *User*

****

Gambar IV. 8 Diagram *activity* kelola data presensi *user*

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *HRD* | * *HRD* sudah melakukan *login.* * Pilih menu data presensi pegawai*.* * Halaman presensi pegawai tampil*.* * *HRD* dapat melakukan fungsi tampil detail presensi pegawai yang dipilih dan hapus presensi pegawai yang dipilih. |

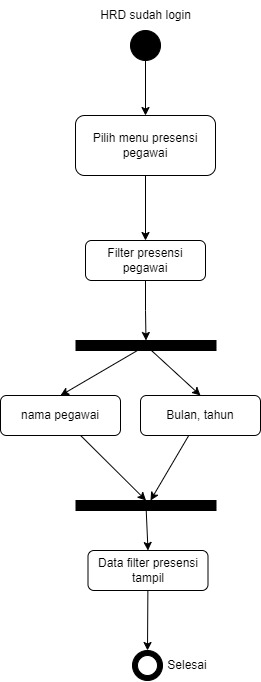
* + - 1. Diagram *Activity Export* Data Presensi Ke *Pdf*

****

Gambar IV. 9 Diagram *activity export* data presensi ke *pdf*

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *HRD* | * *HRD* sudah melakukan *login.* * Pilih menu presensi pegawai*.* * Halaman presensi pegawai tampil*.* * Pilih tombol *export to pdf.* * Halaman *export to pdf* tampil. * *HRD* dapat melakukan *export* data presensi berdasarkan *filter* atau *export* semua presensi. * *Export* presensi berdasarkan *filter* mengisi *form* nama pegawai dan bulan, tahun. * Data presensi berhasil ter-*export* ke *pdf.* |

* + - 1. Diagram *Activity Filter* Presensi *User*

****

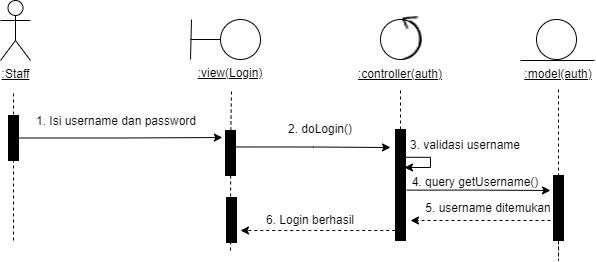
Gambar IV. 10 Keterangan diagram *activity export* data presensi ke *pdf*

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| *HRD* | * *HRD* sudah melakukan *login.* * Pilih menu presensi pegawai*.* * Halaman presensi pegawai tampil*.* * *Filter* presensi pegawai dengan mengisikan nama dan bulan*.* * Data *filter* presensi ditampilkan. |

1. *Diagram Sequence*

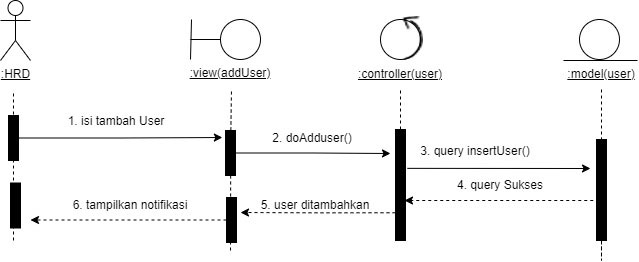
Diagram *Sequence* pada aplikasi akan dijelaskan pada gambar di bawah ini:

* + - 1. **Diagram *Sequence* *Login***



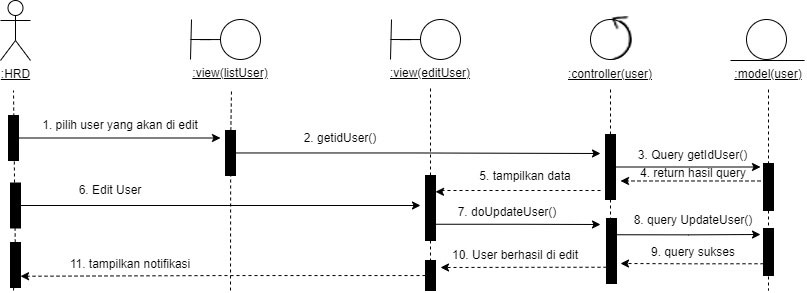
Gambar IV. 11 *Sequence* diagram *login*

* + - 1. Diagram *Sequence* Tambah *User*



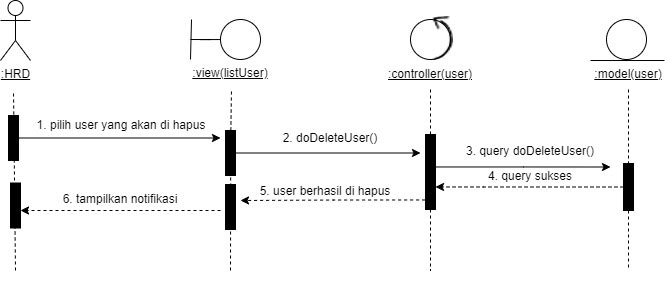
Gambar IV. 12 *Sequence* diagram tambah *user*

* + - 1. Diagram *Sequence* Edit *User*



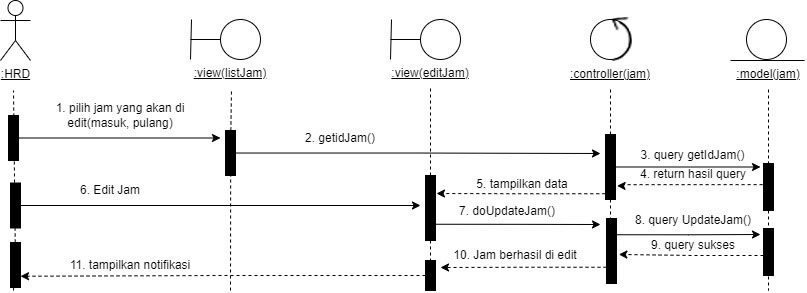
Gambar IV. 13 *Sequence* diagram edit *user*

* + - 1. Diagram *Sequence Delete User*



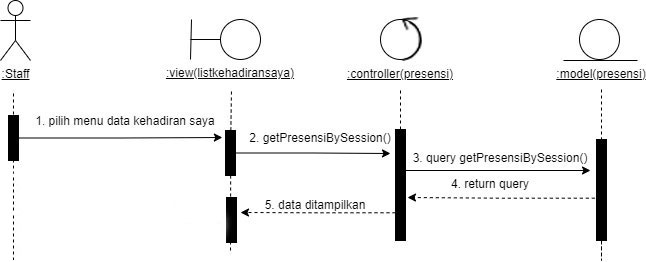
Gambar IV. 14 *Sequence* diagram *delete user*

* + - 1. Diagram Sequence Setting Jam

****

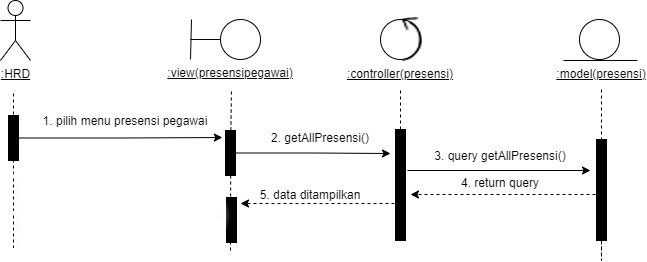
Gambar IV. 15 *Sequence* diagram *setting* jam.

* + - 1. Diagram Sequence Tampil Data Presensi Sendiri

****

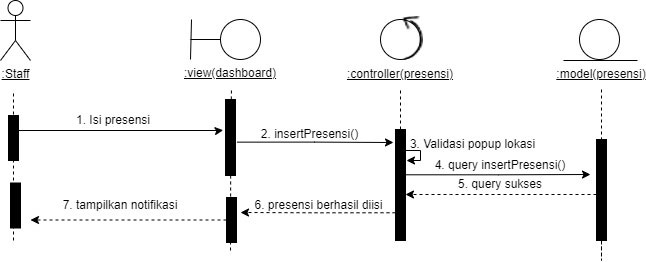
Gambar IV. 16 *Sequence* diagram tampil data presensi sendiri.

* + - 1. Diagram Sequence Kelola Data Presensi User

****

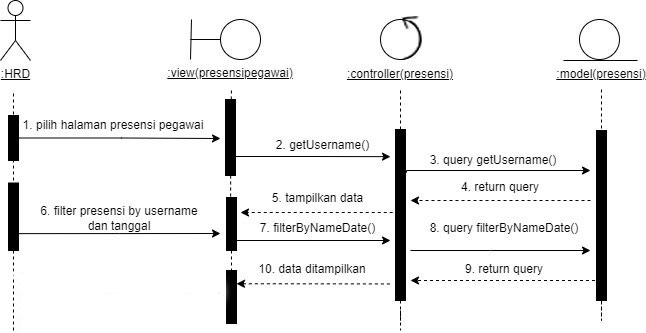
Gambar IV. 17 *Sequence* diagram kelola data presensi *user*.

* + - 1. Diagram Sequence Isi Presensi

****

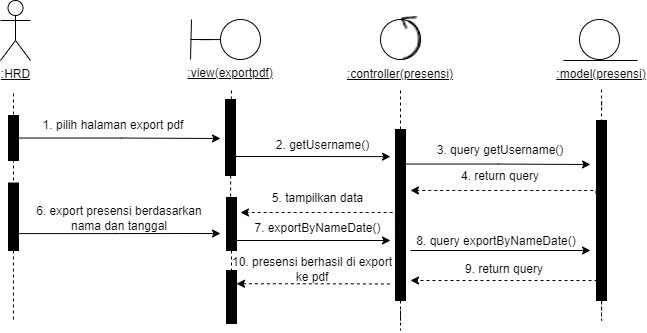
Gambar IV. 18 *Sequence* diagram isi presensi.

* + - 1. Diagram Sequence Filter Presensi

****

Gambar IV. 19 *Sequence* diagram *filter* presensi.

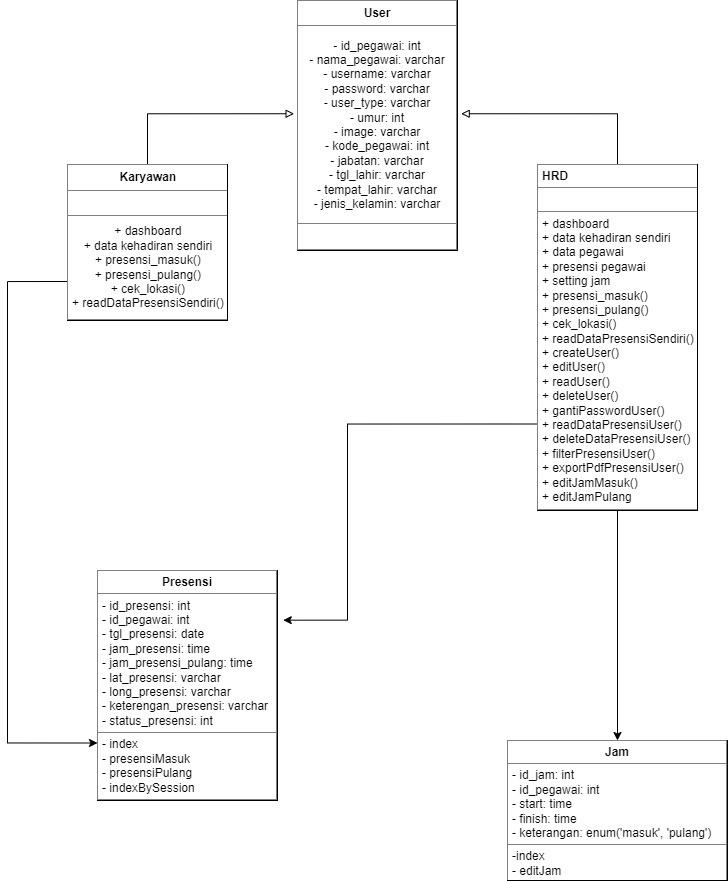
* + - 1. Diagram Sequence Export Presensi To PDF

****

Gambar IV. 20 *Sequence* diagram *export* presensi *to pdf*.

1. *Class* Diagram

Diagram Class ialah alat untuk membantu menggambarkan serta memvisualisasikan struktur kelas yang terdapat dalam suatu sistem

****

Gambar IV. 21 *Class* diagram.

1. Perancangan Basis Data
   * + 1. **Tabel *User***

Tabel IV. 22 Peracangan Tabel *User*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | ***Field*** | **Tipe Data** | **Keterangan** |
| **1** | Id\_pegawai | *Integer* | Sebagai *Primary* *Key* *user* |
| **2** | Nama\_lengkap | *Varchar(125)* | Sebagai tanda pengenal *user* |
| **3** | *Username* | *Varchar(125)* | Sebagai *account* yang digunakan untuk masuk ke dalam sistem |
| **4** | *Password* | *Varchar(125)* | Sebagai kata sandi *account* yang digunakan untuk masuk ke dalam sistem |
| **5** | *User\_type* | *Varchar(125)* | Sebagai *role* akun untuk login ke dalam sistem |
| **6** | Umur | *int* | Sebagai data diri umur *user* |
| **7** | *Image* | *Varchar(125)* | Sebagai data diri gambar *profile* *user* |
| **8** | Kode\_pegawai | *Varchar(125)* | Sebagai data diri kode pegawai *user* |
| **9** | Jabatan | *Varchar(125)* | Sebagai data diri jabatan *user* |
| **10** | Tgl\_lahir | *Varchar(25)* | Sebagai data diri tanggal lahir *user* |
| **11** | Tempat\_lahir | *Varchar(25)* | Sebagai data diri tempat lahir *user* |
| **12** | Jenis\_kelamin | *Varchar(25)* | Sebagai data diri jenis kelamin *user* |

* + - 1. **Tabel Presensi**

Tabel IV. 23 Peracangan Tabel *User*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | ***Field*** | **Tipe Data** | **Keterangan** |
| **1** | Id\_presensi | *Integer* | Sebagai *Primary* *Key* presensi |
| **2** | Id\_pegawai | *Integer* | Sebagai *Foreign key* dari *user* |
| **3** | Tgl\_presensi | *Date* | Sebagai tanggal presensi |
| **4** | Jam\_presensi | *Time* | Sebagai jam masuk presensi |
| **5** | Jam\_presensi Pulang | *Time* | Sebagai jam pulang presensi |
| **6** | *Lat*\_presensi | *Varchar(250)* | Sebagai *latitude maps* presensi |
| **7** | *Long\_*presensi | *Varchar(250)* | Sebagai *longitude maps* presensi |
| **8** | Keterangan\_ presensi | *Varchar(100)* | Sebagai keterangan saat melakukan presensi |
| **9** | Status\_presensi | *Integer* | Sebagai ciri apakah *user* sudah presensi masuk atau presensi pulang |

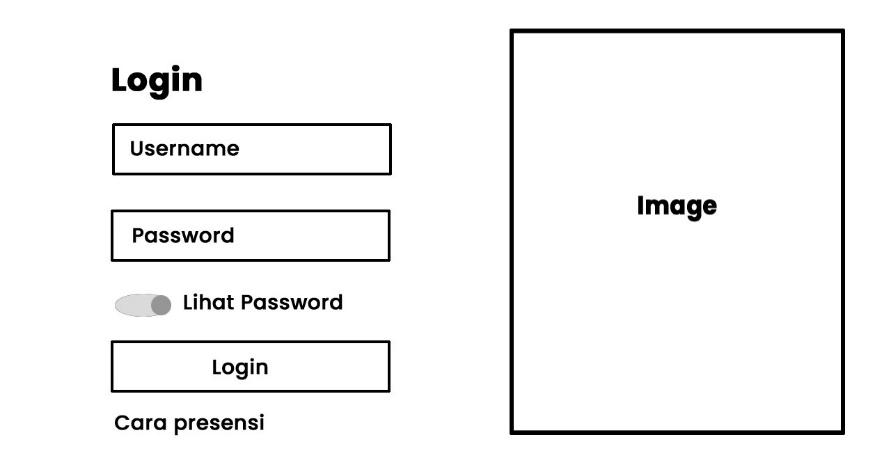
* + - 1. **Tabel Jam**

Tabel IV. 24 Peracangan Tabel Jam

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | ***Field*** | **Tipe Data** | **Keterangan** |
| **1** | Id\_jam | *Integer* | Sebagai *Primary* *Key* jam |
| **2** | *Start* | *Time* | Sebagai awal waktu untuk presensi |
| **3** | *Finish* | *Time* | Sebagai akhir waktu untuk presensi |
| **4** | keterangan | *Enum( ‘*masuk*’, ‘* pulang*’)* | Sebagai keterangan waktu masuk atau pulang presensi |

1. Perancangan Tampilan
   * + 1. **Perancangan Antarmuka Halaman Login**

Pada halaman ini, merupakan tampilan untuk melakukan aktifitas *login* yang didalamnya terdapat *input text field* untuk mengisi *username, input password field* untuk mengisi kata sandi dari akun, dan terdapat tombol untuk melakukan *login,* perancangan antarmuka ini dapat diakses oleh *user* *HRD* dan karyawan. Rancangan antarmuka halaman *login* ditampilkan pada Gambar IV.22 dibawah ini.



Gambar IV. 22 Perancangan antarmuka halaman *login*.

* + - 1. **Perancangan Antarmuka Halaman Dashboard HRD**

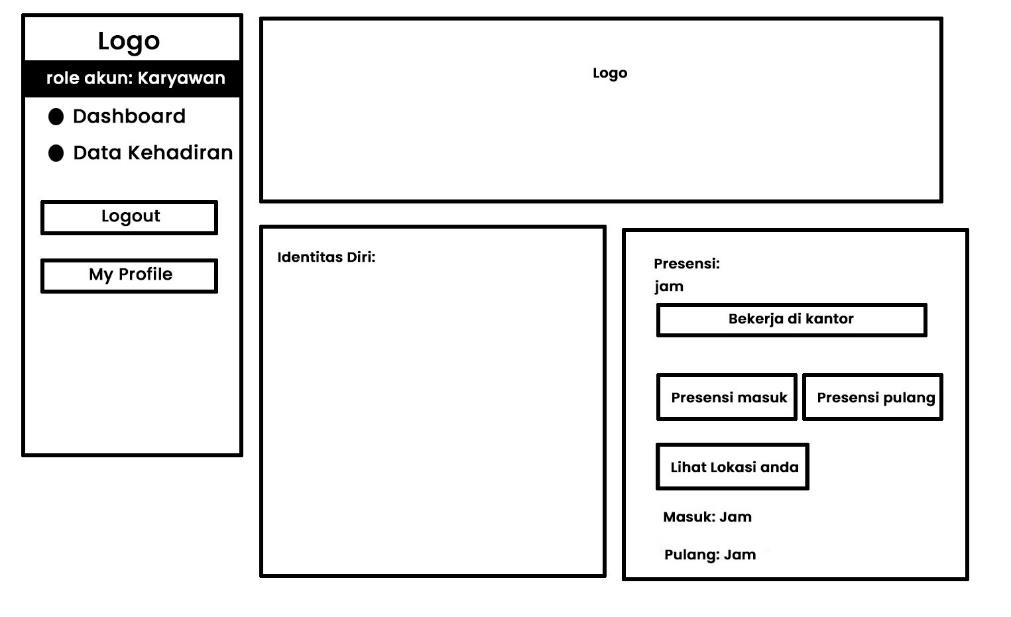
Pada halaman *dashboard HRD*, merupakan tampilan untuk melihat data jumlah pegawai yang sudah melakukan prensesi masuk, presensi pulang, sakit dan cuti. Di halaman ini juga pegawai melakukan cek lokasi, presensi masuk, presensi pulang beserta keterangan presensinya, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman *Dashboard* *HRD* ditampilkan pada gambar IV.23 di bawah ini.



Gambar IV. 23 Perancangan antarmuka halaman *dashboard HRD.*

* + - 1. **Perancangan Antarmuka Halaman Dashboard Karyawan**

Pada halaman *dashboard* karyawan, merupakan tampilan untuk melakukan cek lokasi, presensi masuk, presensi pulang beserta keterangan presensinya, perancangan antarmuka ini dapat diakses oleh *user* Karyawansaja. Rancangan antarmuka halaman *Dashboard* Karyawan ditampilkan pada gambar IV.24 di bawah ini.



Gambar IV. 24 Perancangan antarmuka halaman *dashboard* Karyawan.

* + - 1. **Perancangan Antarmuka Halaman Cek Lokasi**

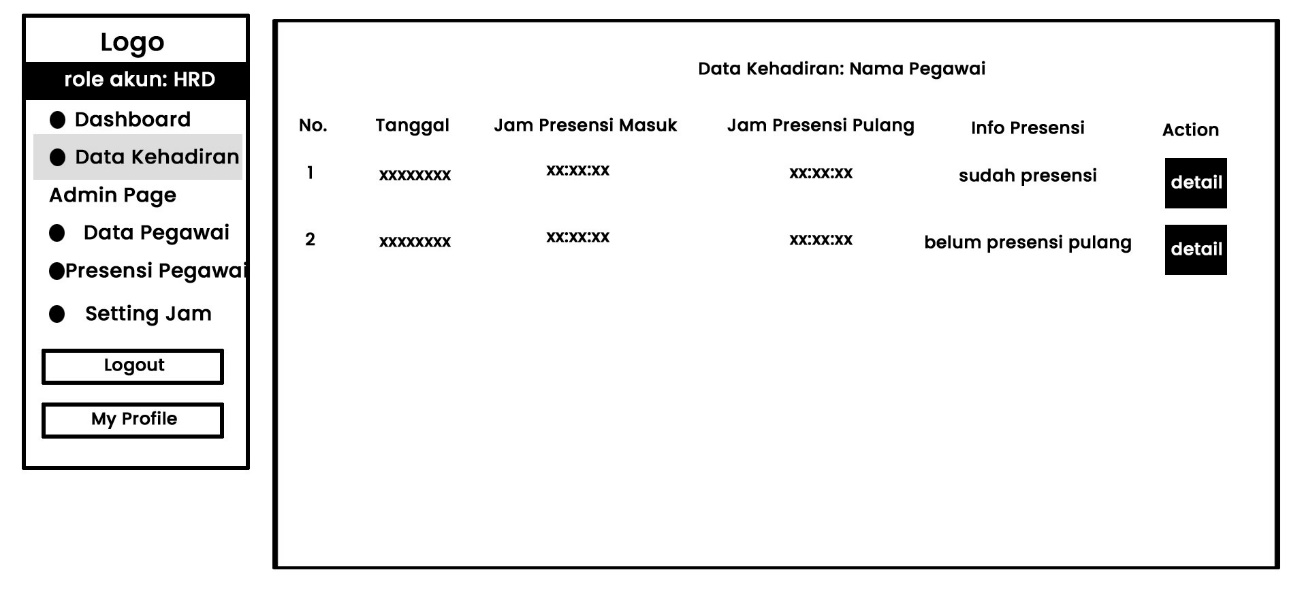
Pada halaman ini, merupakan tampilan detail dari cek lokasi dimana nanti akan muncul *maps* beserta *marker*-nya untuk mengetahui lokasi tersebut sebagai acuan lokasi *maps* saat melakukan presensi. *HRD* dan karyawan diberikan hak otorisasi untuk mengakses halaman ini. Rancangan antarmuka halaman cek lokasi ditampilkan pada gambar IV.25 di bawah ini.



Gambar IV. 25 Perancangan antarmuka halaman cek lokasi.

* + - 1. **Perancangan Antarmuka Halaman Data Kehadiran Sendiri**

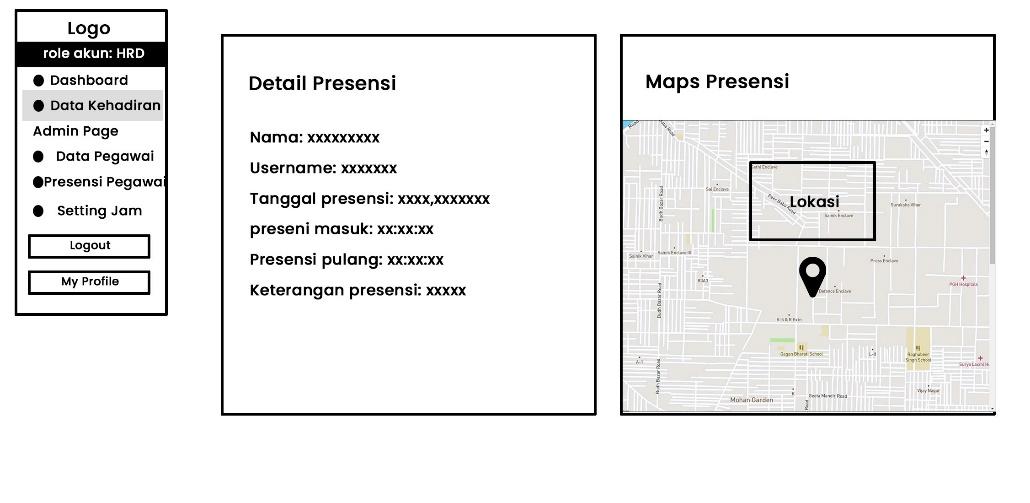
Pada halaman ini, merupakan tampilan untuk melihat data presensi sesuai *login*. Halaman ini terdapat tabel data kehadiran yang berisikan tanggal, nama, jam presensi masuk, jam presensi pulang, info presensi yang berisi sudah presensi, belum presensi pulang, sakit serta cuti dan yang terakhir ada tombol detail untuk melihat detail presensi, perancangan antarmuka ini dapat diakses oleh *user* *HRD* dan karyawan. Rancangan antarmuka halaman data kehadiran sendiri ditampilkan pada gambar IV.26 di bawah ini.



Gambar IV. 26 Perancangan antarmuka halaman data kehadiran sendiri.

* + - 1. **Perancangan Antarmuka Halaman Detail Presensi Sendiri**

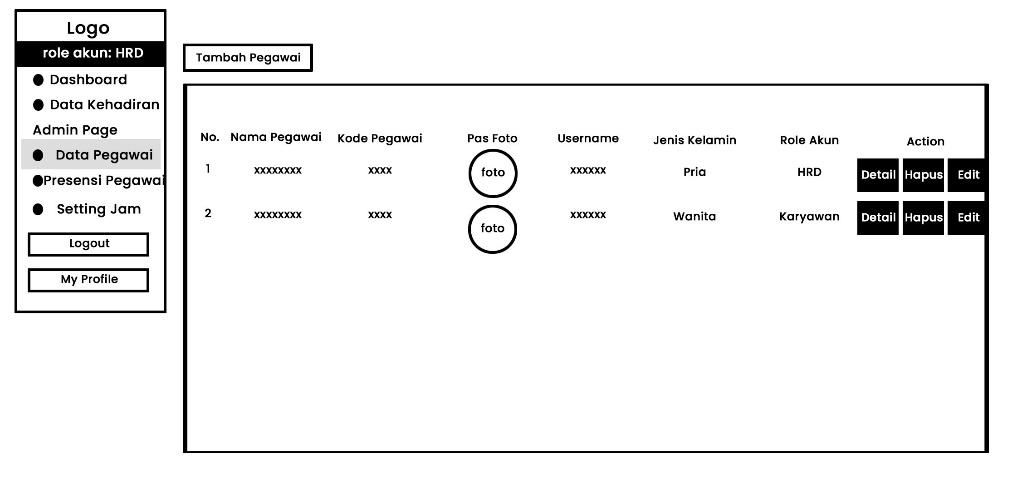
Pada halaman ini, merupakan tampilan untuk melihat data detail presensi. Halaman ini terdapat data detail presensi user yang dipilih yang terdiri dari nama, *username,* tanggal presensi, presensi masuk, presensi pulang serta keterangan presensi dan ada *maps* beserta *marker-*nya saat *user* melakukan presensi, perancangan antarmuka ini dapat diakses oleh *user* *HRD* dan karyawan. Rancangan antarmuka halaman detail kehadiran sendiri ditampilkan pada gambar IV.27 di bawah ini.



Gambar IV. 27 Perancangan antarmuka halaman data detail presensi.

* + - 1. **Perancangan Antarmuka Halaman Data Pegawai**

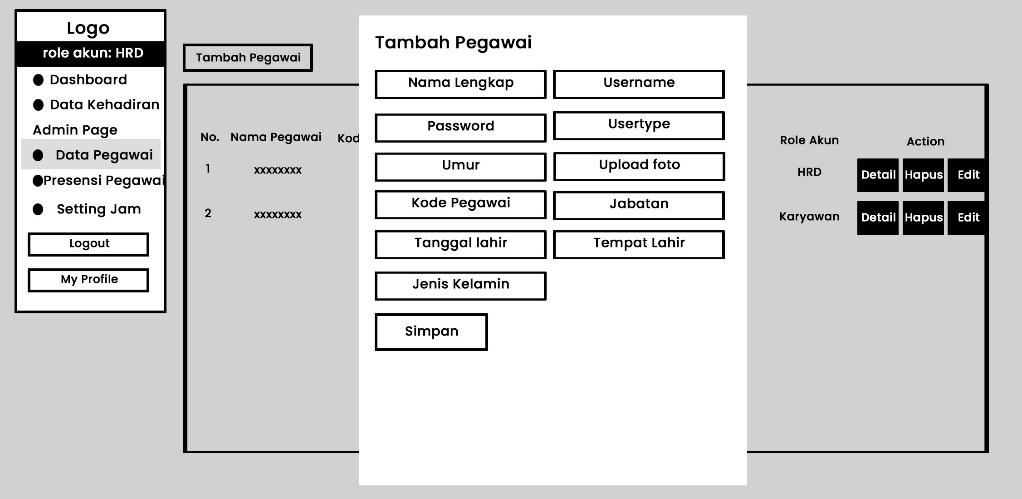
Pada halaman ini, merupakan tampilan untuk melihat data pegawai. Halaman ini terdapat tabel pegawai yang berisikan identitas pegawai, tombol tambah pegawai, detail, hapus, dan edit pegawai, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman data kehadiran sendiri ditampilkan pada gambar IV.28 di bawah ini.



Gambar IV. 28 Perancangan antarmuka halaman data pegawai.

* + - 1. **Perancangan Antarmuka Halaman Tambah Pegawai**

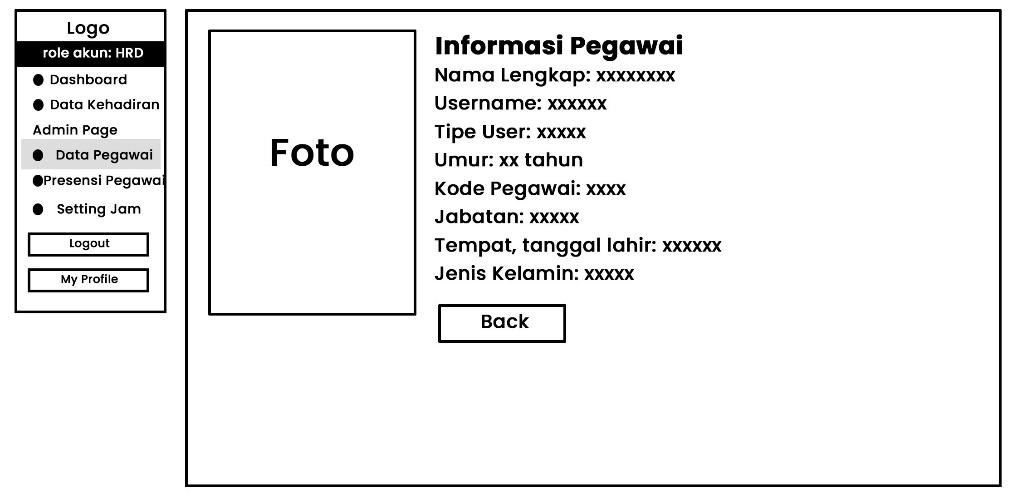
Pada halaman ini, merupakan tampilan rancangan tambah pegawai. Tambah pegawai tampilannya akan seperti *popup* yang muncul apabila tombol tambah pegawai diklik, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka tambah pegawai ditampilkan pada gambar IV.29 di bawah ini.



Gambar IV. 29 Perancangan antarmuka tambah pegawai.

* + - 1. **Perancangan Antarmuka Halaman Detail Pegawai**

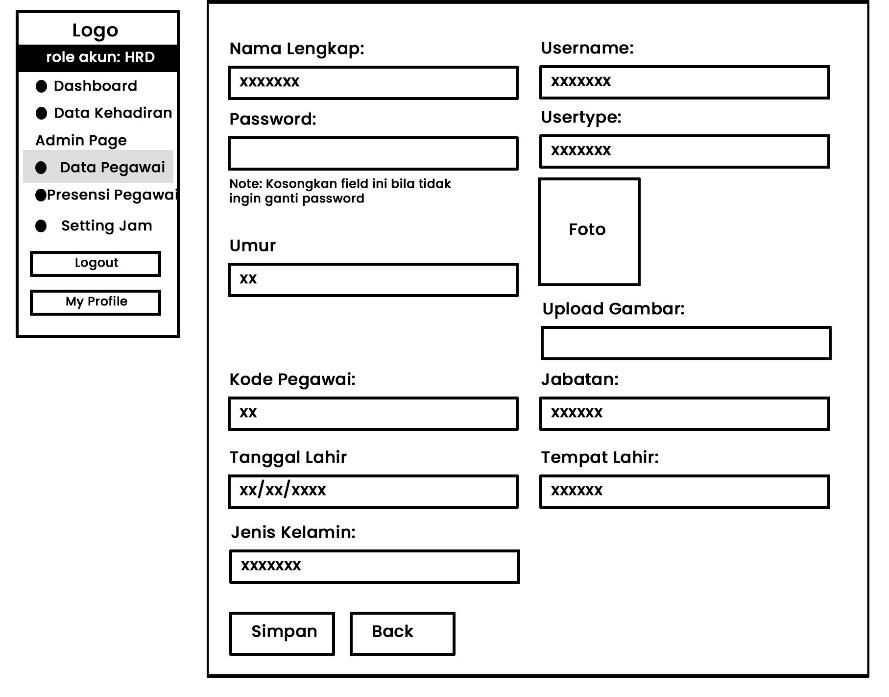
Pada halaman ini, merupakan tampilan rancangan detail pegawai. Halaman detail pegawai akan berisikan foto pegawai serta identitas dari pegawai tersebut, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman detail pegawai ditampilkan pada gambar IV.30 di bawah ini.



Gambar IV. 30 Perancangan antarmuka halaman detail pegawai.

* + - 1. **Perancangan Antarmuka Halaman Edit Pegawai**

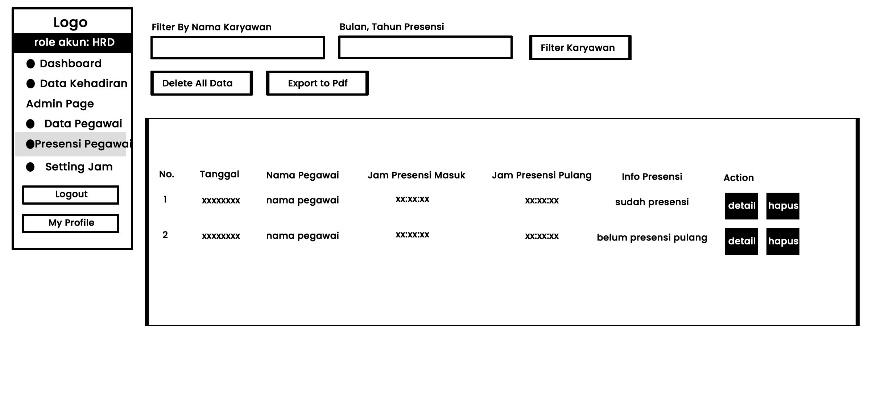
Pada halaman ini, merupakan tampilan rancangan halaman *edit* pegawai. Halaman *edit* pegawai akan berisikan foto pegawai serta identitas dari pegawai tersebut yang bisa diubah, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman *edit* pegawai ditampilkan pada gambar IV.31 di bawah ini.



Gambar IV. 31 Perancangan antarmuka halaman *edit* pegawai.

* + - 1. **Perancangan Antarmuka Halaman Presensi Pegawai**

Pada halaman ini, merupakan tampilan rancangan halaman presensi pegawai. Halaman presensi pegawai akan berisikan data presensi semua *user* dan ada *form filter* presensi serta ada tombol *export to pdf*, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman presensipegawai ditampilkan pada gambar IV.32 di bawah ini.



Gambar IV. 32 Perancangan antarmuka halaman presensi pegawai.

* + - 1. **Perancangan Antarmuka Halaman Filter Presensi Pegawai**

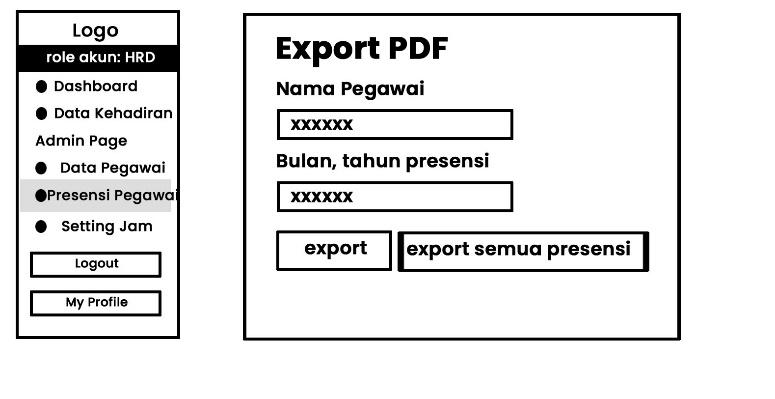
Pada halaman ini, merupakan tampilan rancangan halaman *filter* presensi pegawai. Halaman *filter* presensi pegawai akan berisikan data presensi *user* yang telah di *filter*, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman *filter* presensipegawai ditampilkan pada gambar IV.33 di bawah ini.



Gambar IV. 33 Perancangan antarmuka halaman filter presensi pegawai.

* + - 1. **Perancangan Antarmuka Halaman Export Presensi Pegawai**

Pada halaman ini, merupakan tampilan rancangan halaman *export* presensi pegawai. Halaman *export* presensi pegawai akan berisikan *form* nama pegawai dan *form* bulan tahun yang akan di *export* ke *pdf* serta ada tombol untuk *export* semua data presensi, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman *export* presensipegawai ditampilkan pada gambar IV.27 di bawah ini.



Gambar IV. 34 Perancangan antarmuka halaman *export* presensi pegawai

* + - 1. **Perancangan Antarmuka Halaman Atur Jam Kerja**

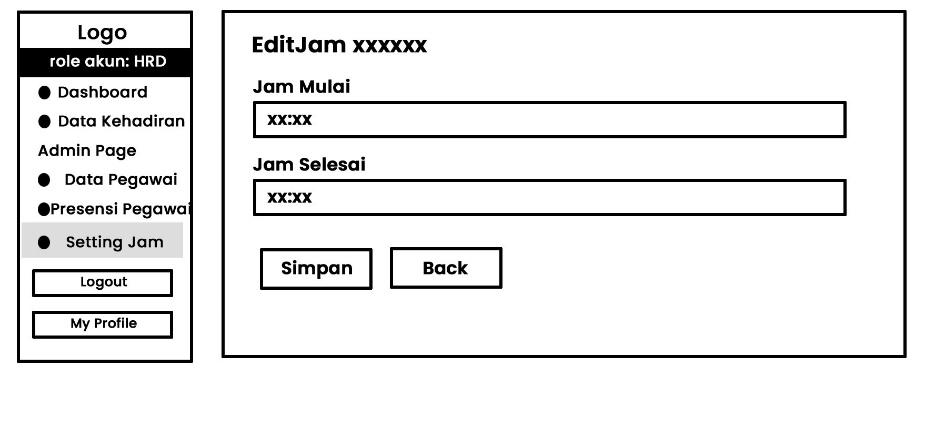
Pada halaman ini, merupakan tampilan rancangan halaman atur jam kerja. Halaman atur jam kerja akan berisikan tabel jam presensi dan button *edit*, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman atur jam kerja ditampilkan pada gambar IV.35 di bawah ini.



Gambar IV. 35 Perancangan antarmuka halaman atur jam kerja.

* + - 1. **Perancangan Antarmuka Halaman Edit Jam Kerja**

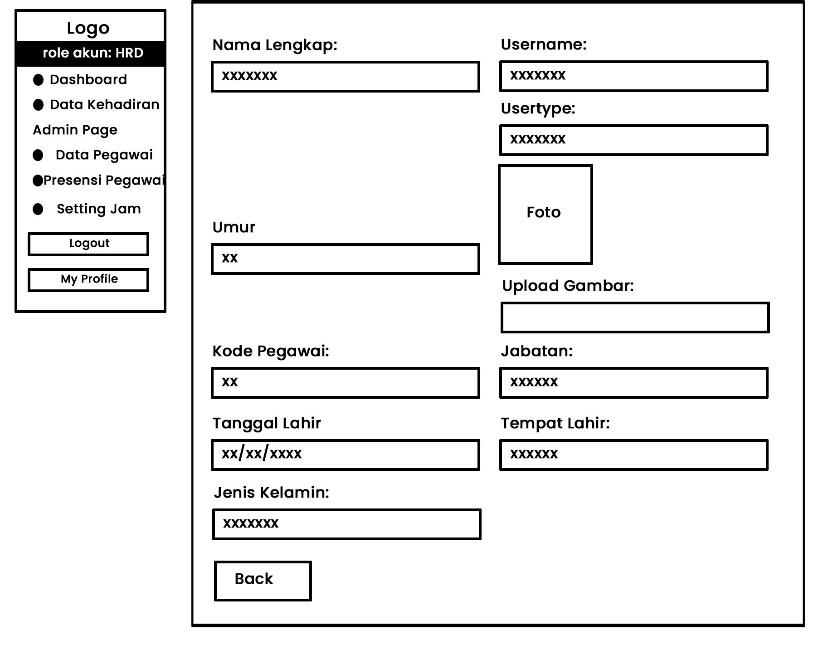
Pada halaman ini, merupakan tampilan rancangan halaman edit jam kerja. Halaman atur jam kerja akan berisikan *form* jam mulai dan jam selesai, yang nantinya bisa diubah jamnya sesuai keterangan masuk atau pulang, perancangan antarmuka ini dapat diakses oleh *user* *HRD* saja. Rancangan antarmuka halaman edit jam kerja ditampilkan pada gambar IV.36 di bawah ini.



Gambar IV. 36 Perancangan antarmuka halaman edit jam kerja

* + - 1. **Perancangan Antarmuka Halaman Profile**

Pada halaman ini, merupakan tampilan rancangan halaman *profile*. Halaman *profile* akan berisikan *form* yang di-*disable* dalam *form* tersebut akan muncul profile diri sesuai *user* yang login, perancangan antarmuka ini dapat diakses oleh *user* *HRD* dan karyawan. Rancangan antarmuka halaman *profile* ditampilkan pada gambar IV.37 di bawah ini.

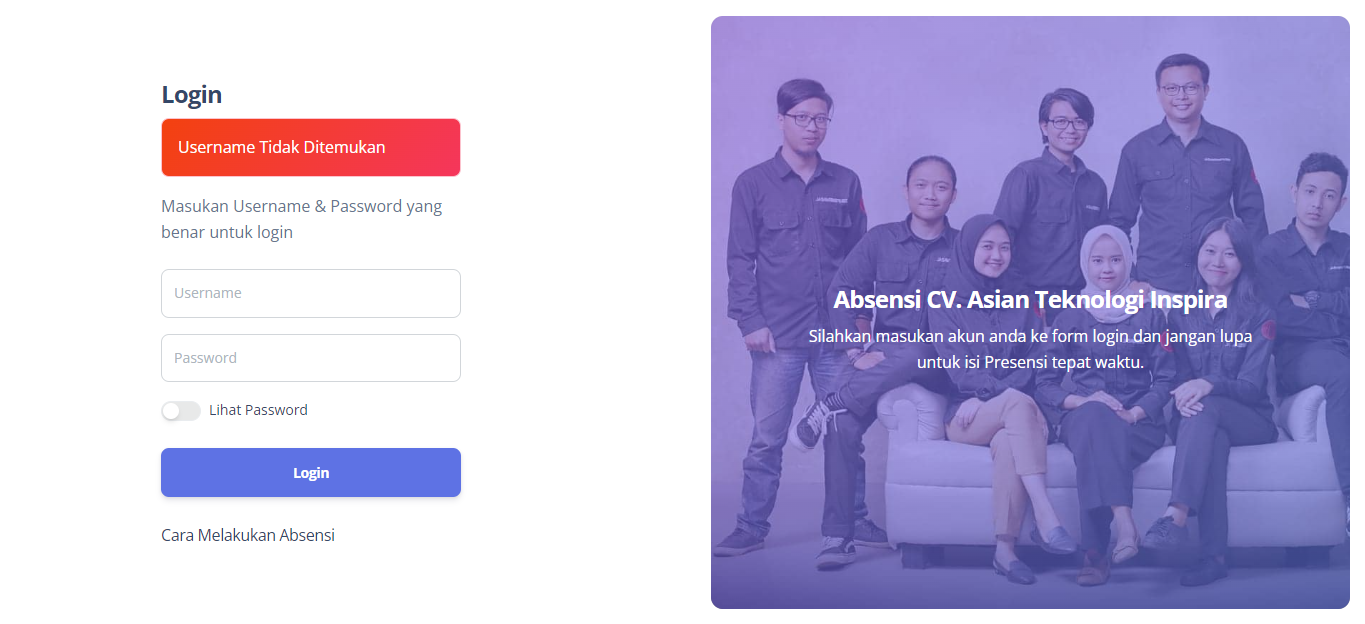


Gambar IV. 37 Perancangan antarmuka halaman *profile*.

1. Implementasi Antarmuka

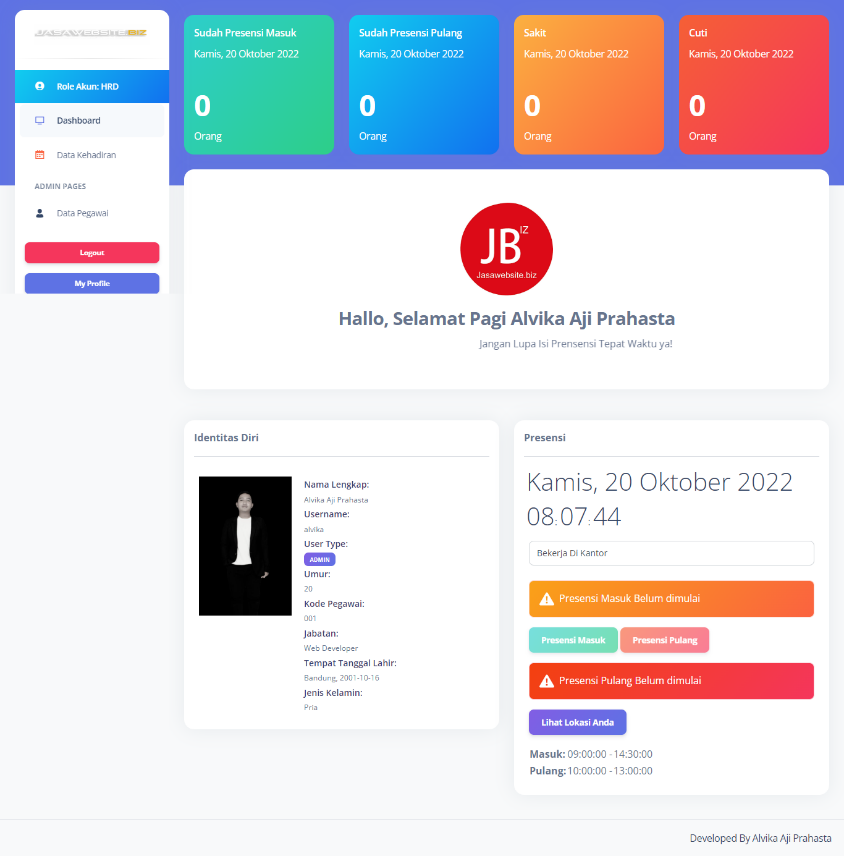
Implementasi antarmuka aplikasi *web* presensi berbasis *GPS* akan dijelaskan pada gambar dibawah ini:

* + - 1. **Implementasi Halaman *Login***



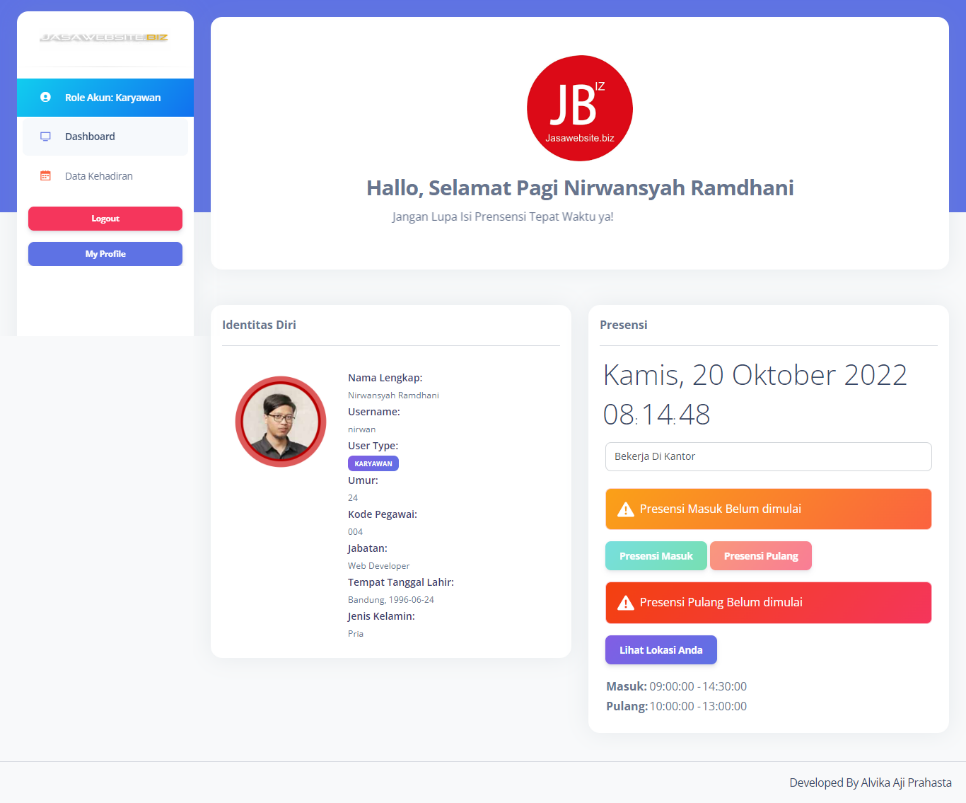
Gambar IV. 38 Implementasi halaman *login.*

* + - 1. **Implementasi Halaman Dashboard HRD**



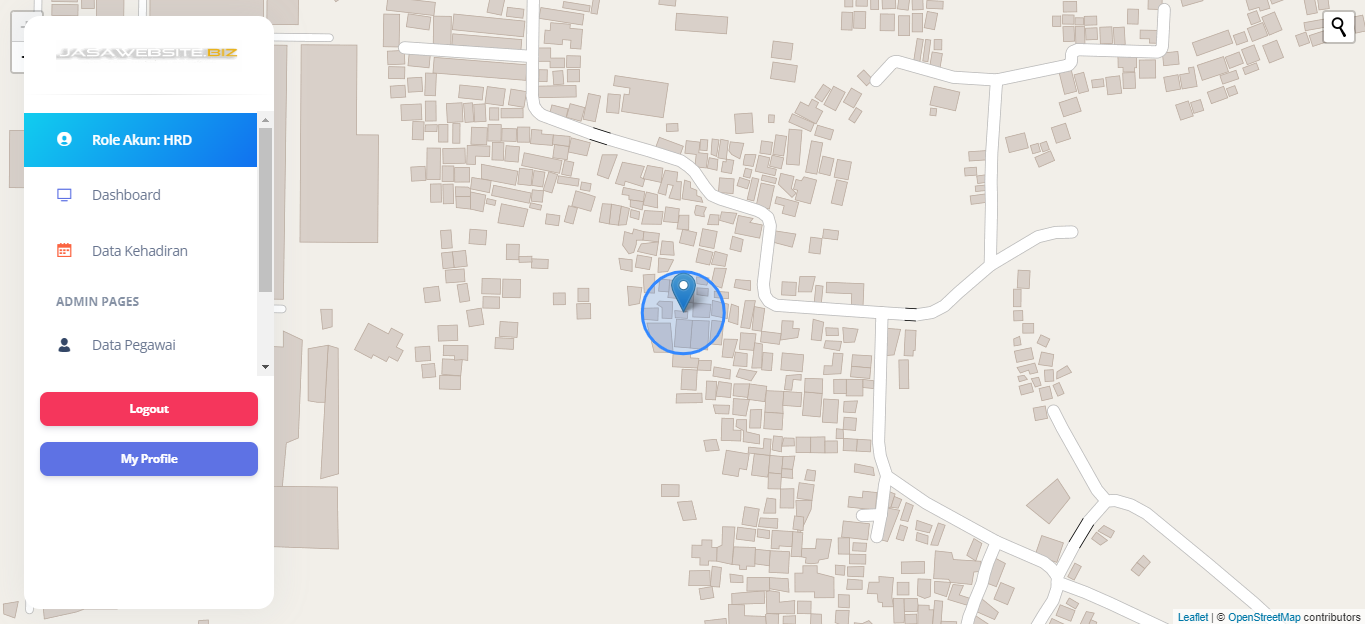
Gambar IV. 39 Implementasi halaman *dashboard HRD*

* + - 1. **Implementasi Halaman Dashboard Karyawan**



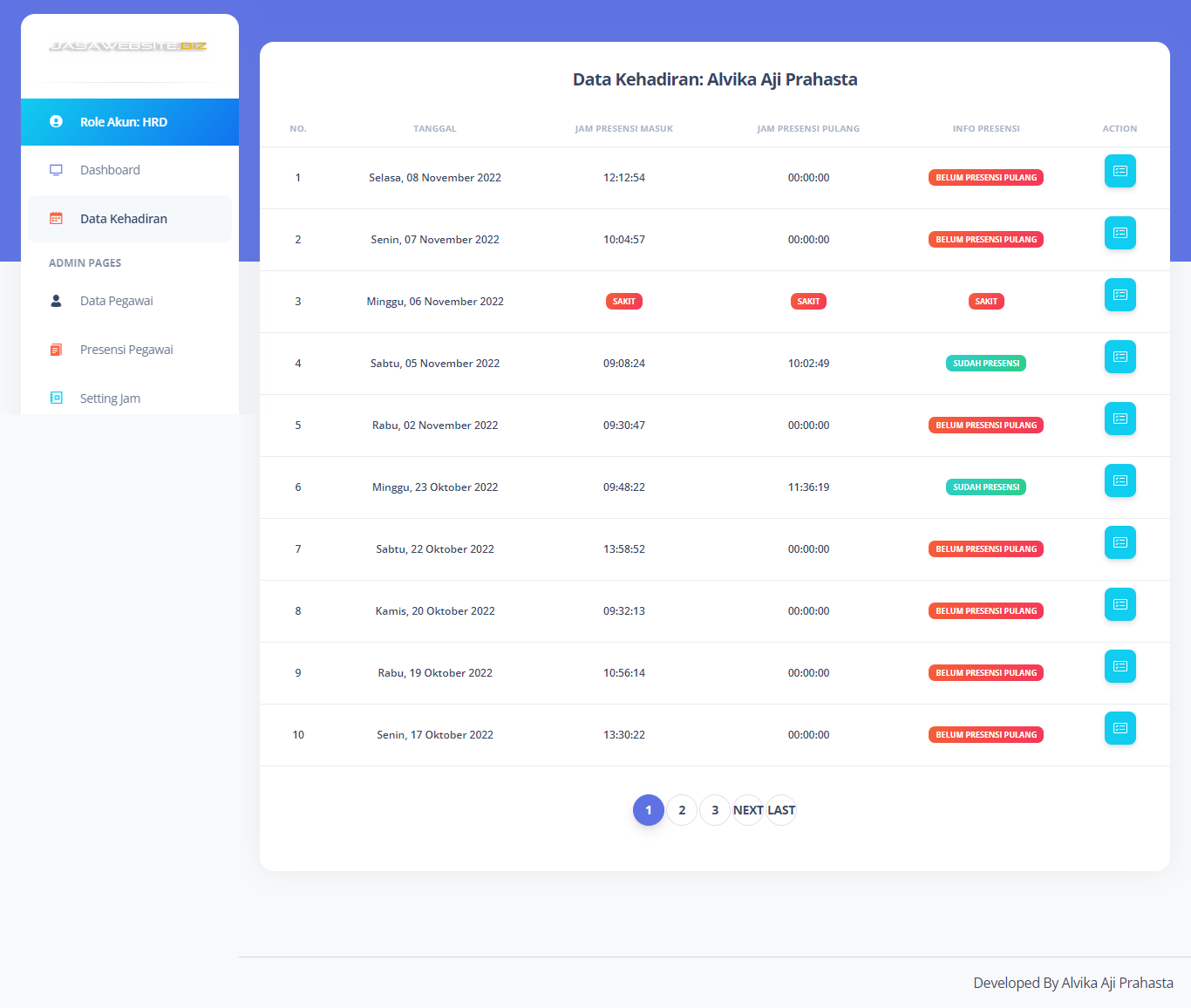
Gambar IV. 40 Implementasi halaman *dashboard* karyawan

* + - 1. **Implementasi Halaman Cek Lokasi**



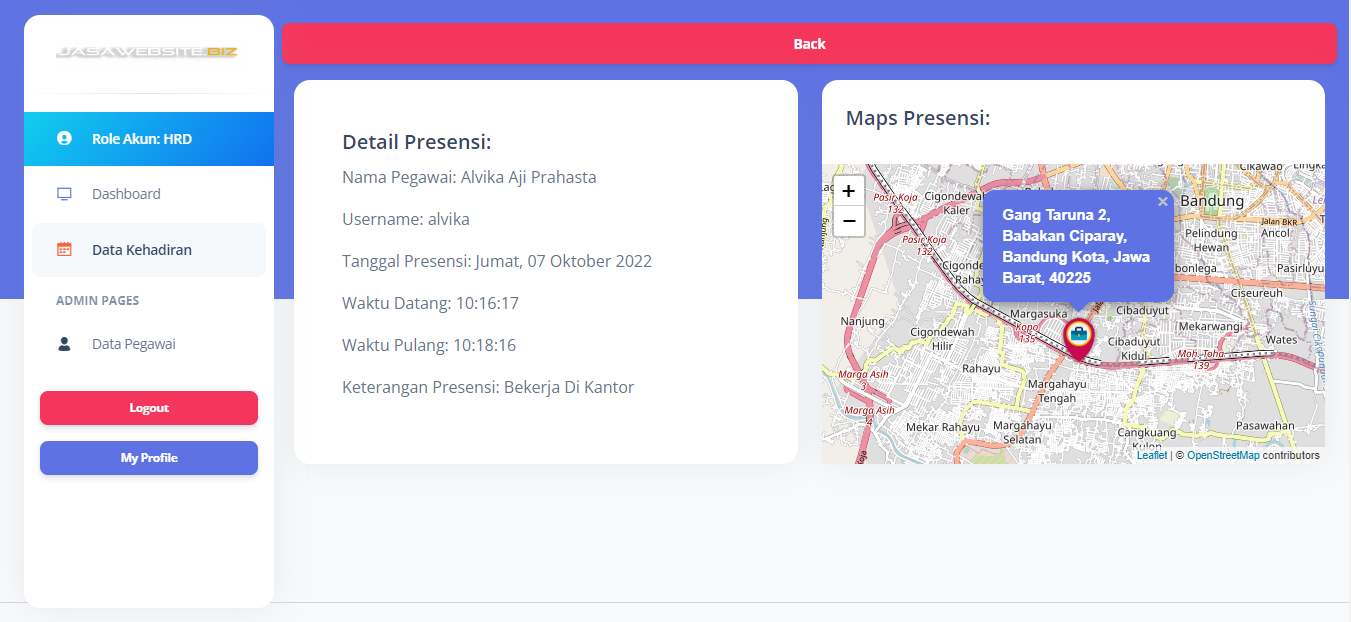
Gambar IV. 41 Implementasi halaman cek lokasi

* + - 1. **Implementasi Halaman Data Kehadiran Sendiri**



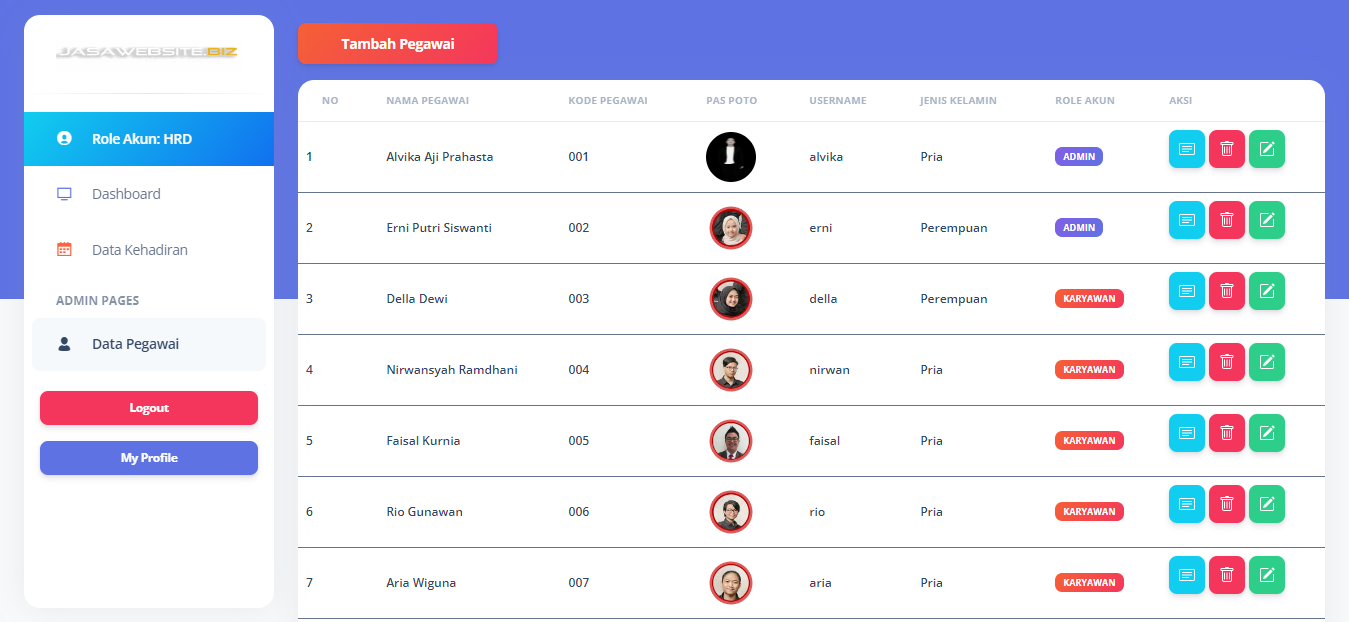
Gambar IV. 42 Implementasi halaman data kehadiran sendiri

* + - 1. **Implementasi Halaman Detail Presensi**



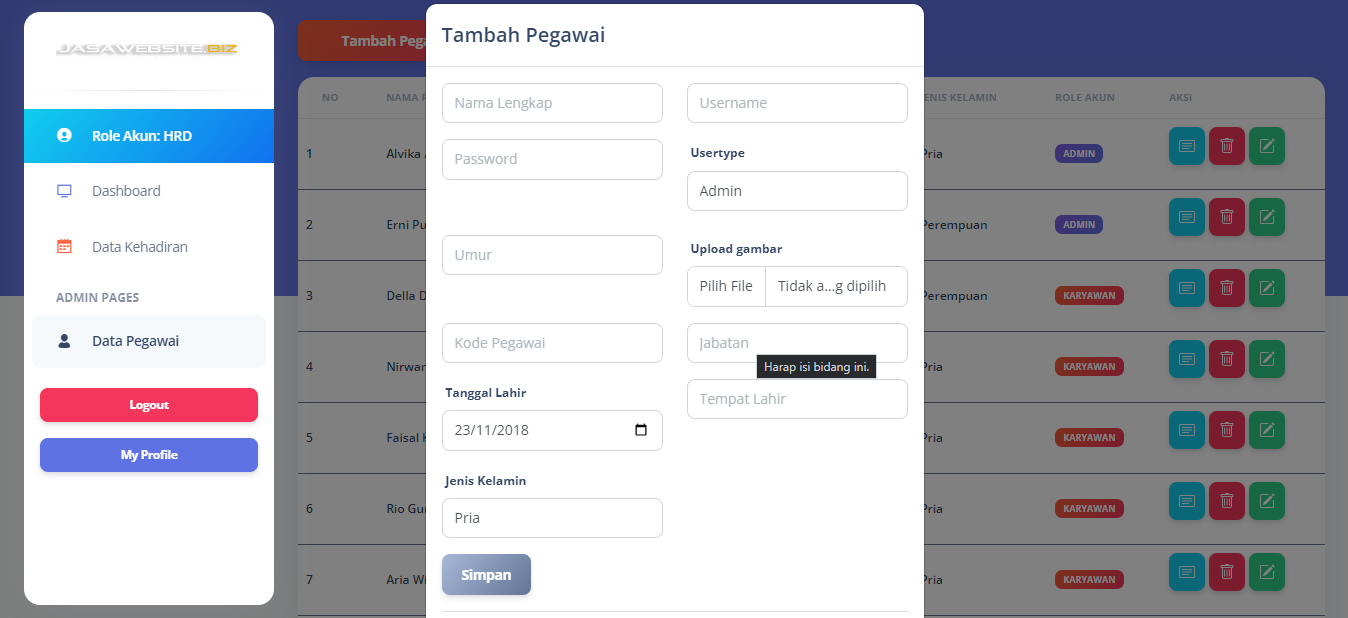
Gambar IV. 43 Implementasi halaman detail presensi

* + - 1. **Implementasi Halaman Data Pegawai**



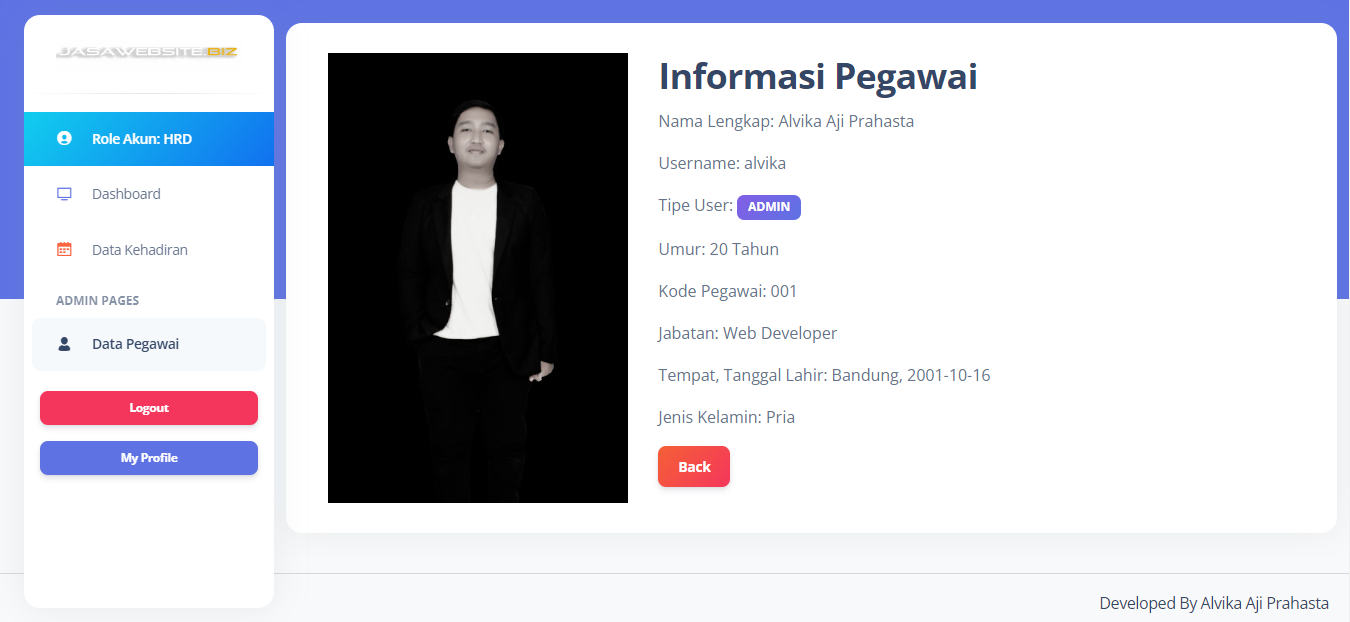
Gambar IV. 44 Implementasi halaman data pegawai

* + - 1. **Implementasi Halaman Tambah Pegawai**



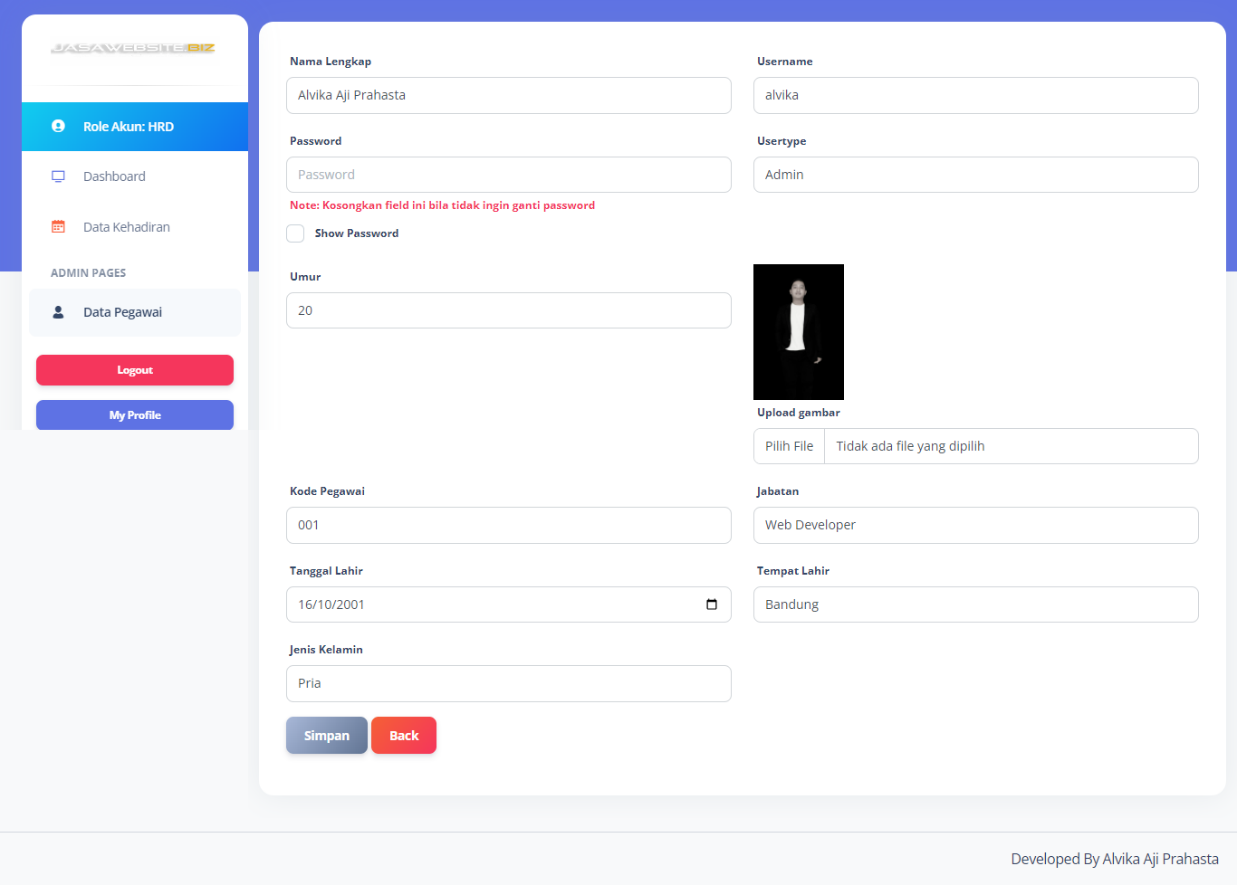
Gambar IV. 45 Implementasi halaman tambah pegawai

* + - 1. **Implementasi Halaman Detail Pegawai**



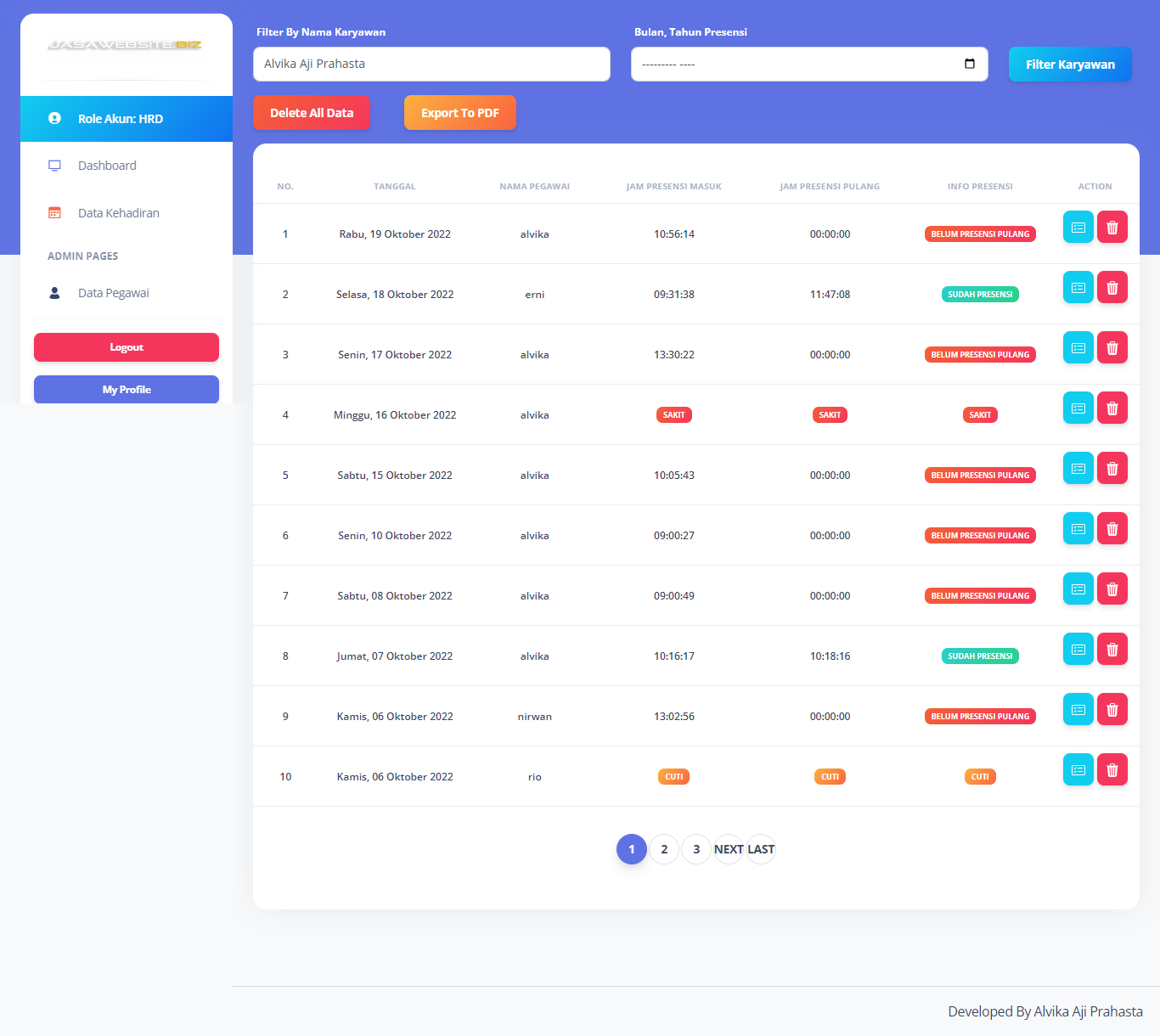
Gambar IV. 46 Implementasi halaman detail pegawai

* + - 1. **Implementasi Halaman Edit Pegawai**



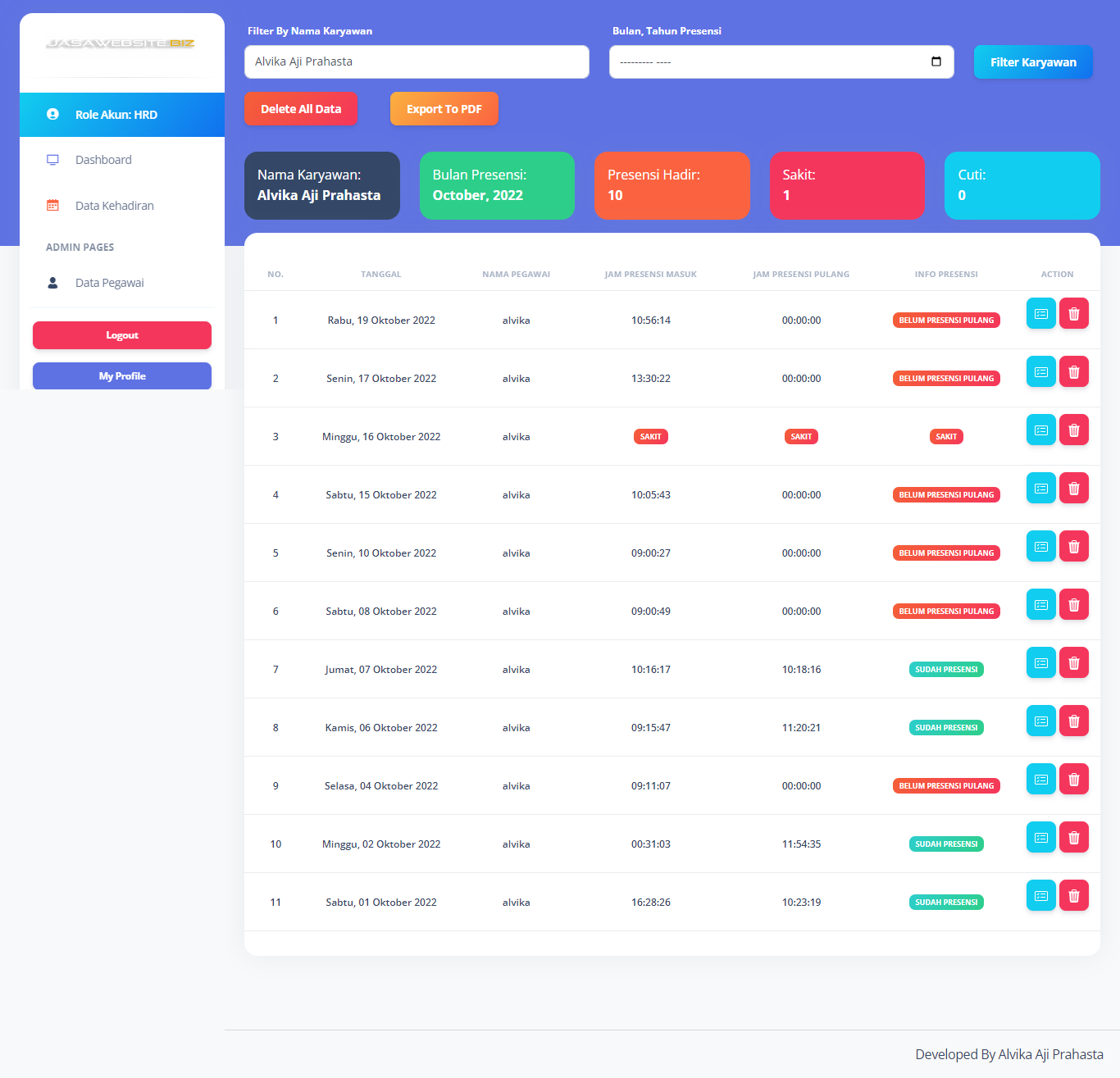
Gambar IV. 47 Implementasi halaman edit pegawai

* + - 1. **Implementasi Halaman Presensi Pegawai**



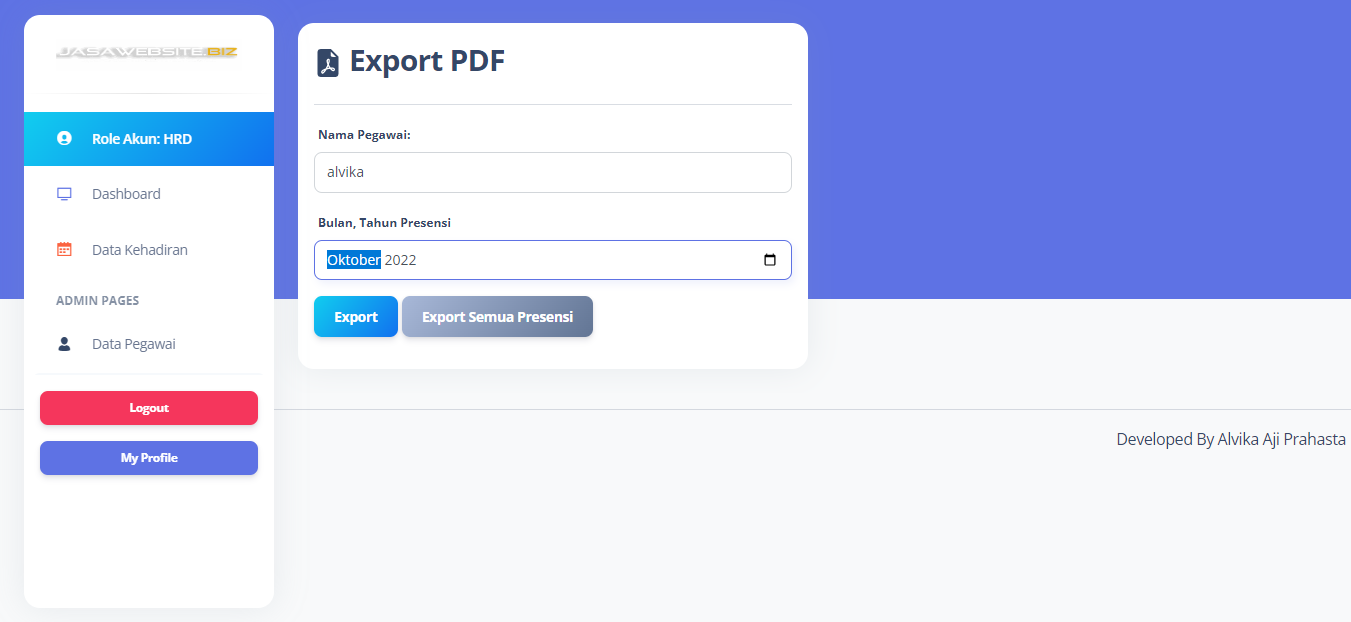
Gambar IV. 48 Implementasi halaman presensi pegawai

* + - 1. **Implementasi Halaman Filter Presensi Pegawai**



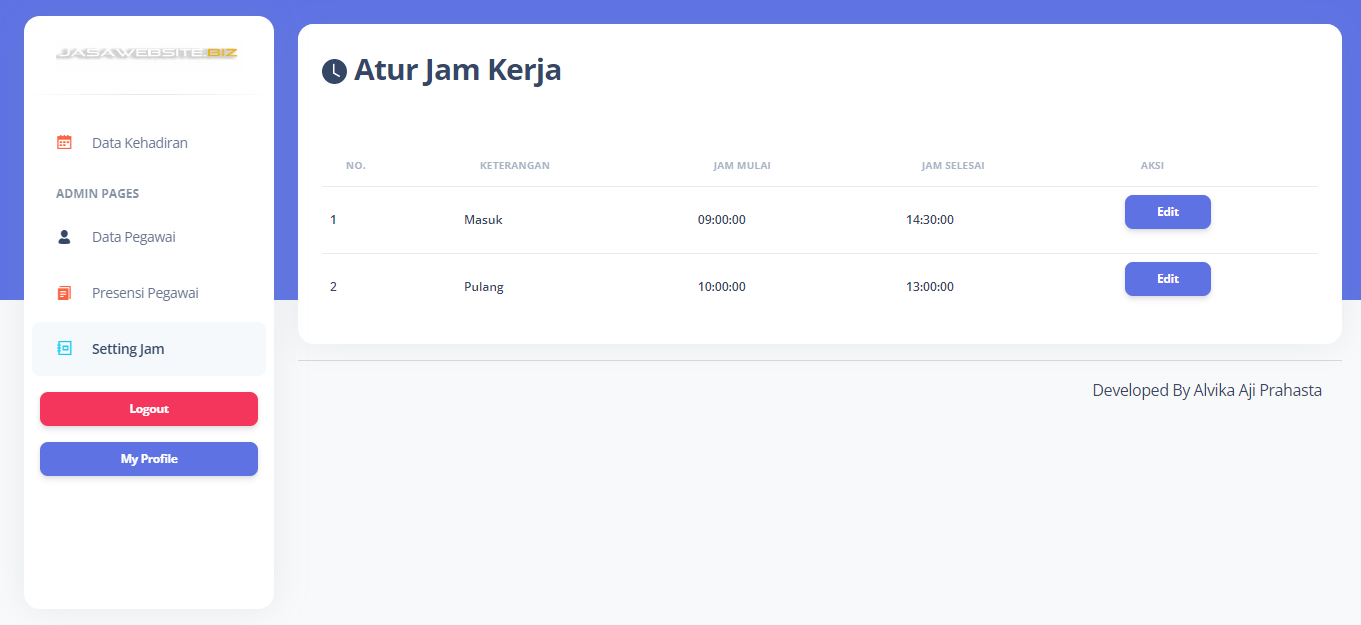
Gambar IV. 49 Implementasi halaman *filter* presensi pegawai

* + - 1. **Implementasi Halaman Export Presensi Pegawai**



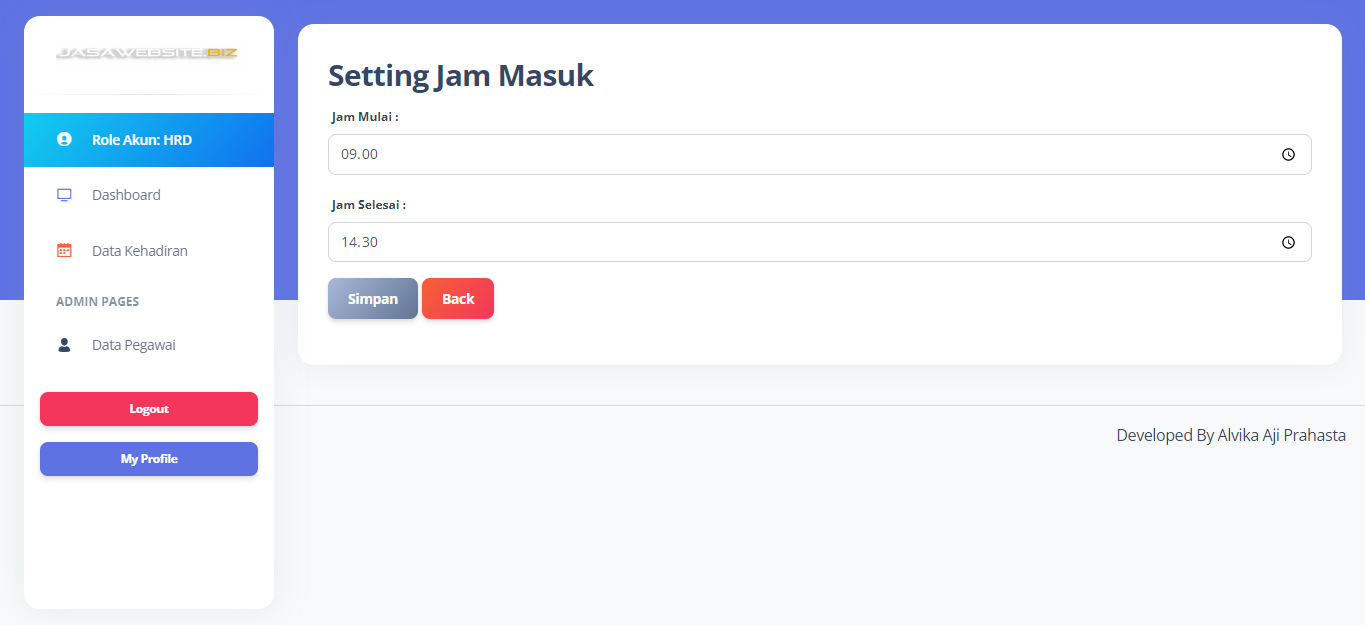
Gambar IV. 50 Implementasi halaman *export* presensi pegawai

* + - 1. **Implementasi Halaman Atur Jam Kerja**



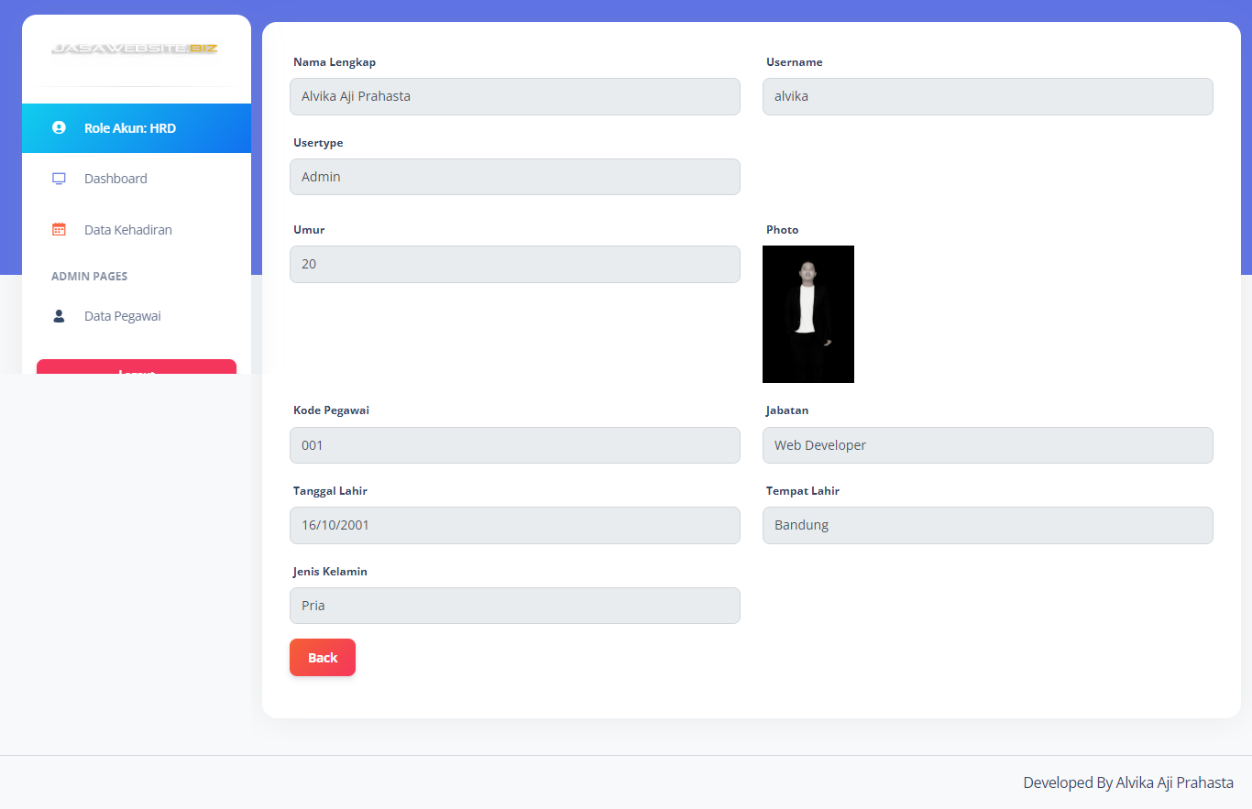
Gambar IV. 51 Implementasi halaman atur jam kerja

* + - 1. **Implementasi Halaman Edit Jam Kerja**



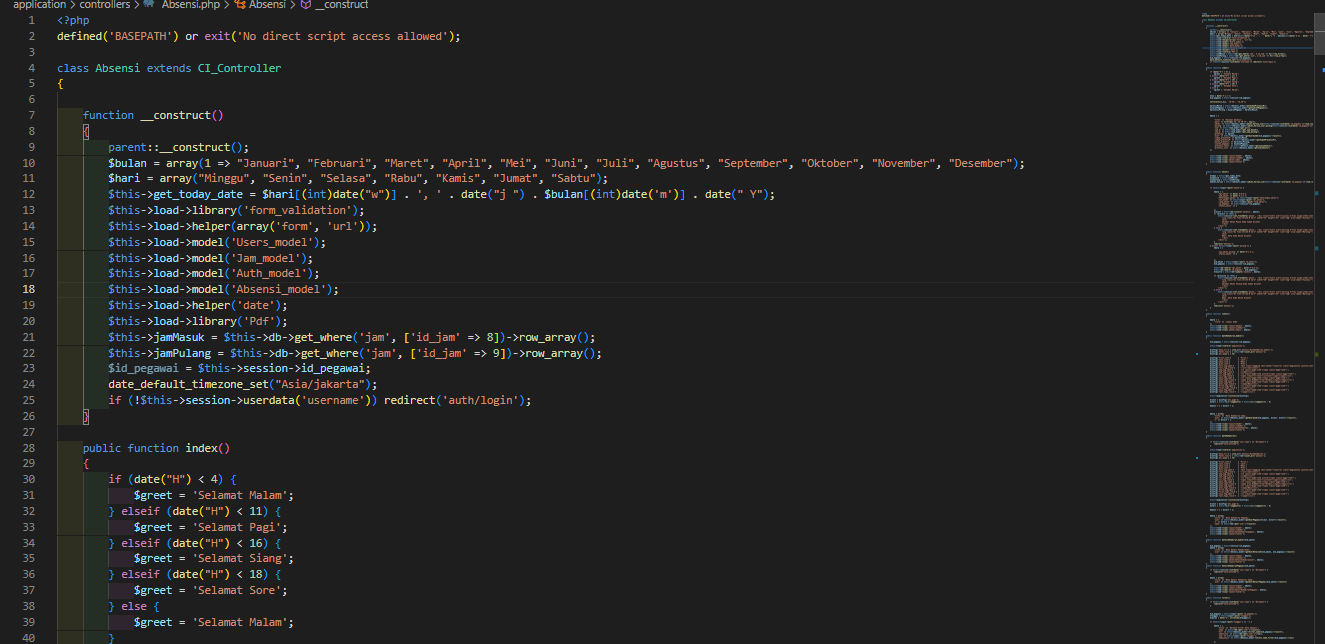
Gambar IV. 52 Implementasi halaman edit jam kerja

* + - 1. **Implementasi Halaman Profile**



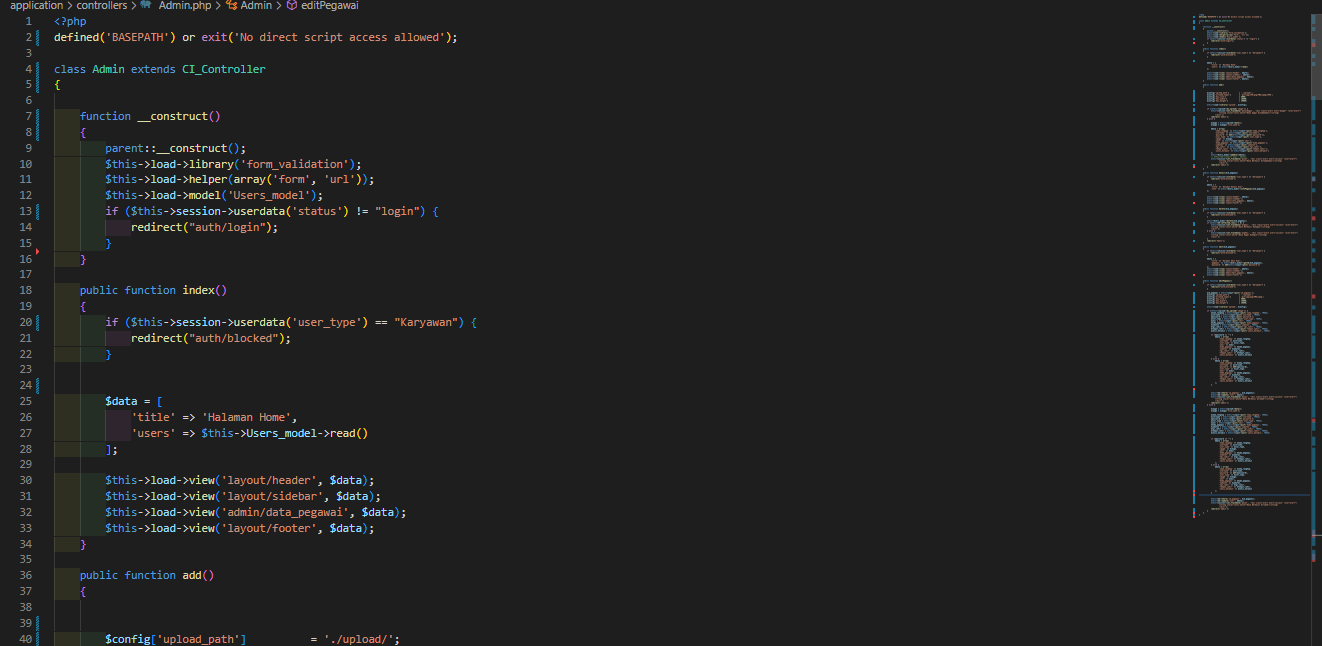
Gambar IV. 53 Implementasi halaman *profile*

1. Implementasi *Code*
   * + 1. **Implementasi *Code Controller* Presensi**



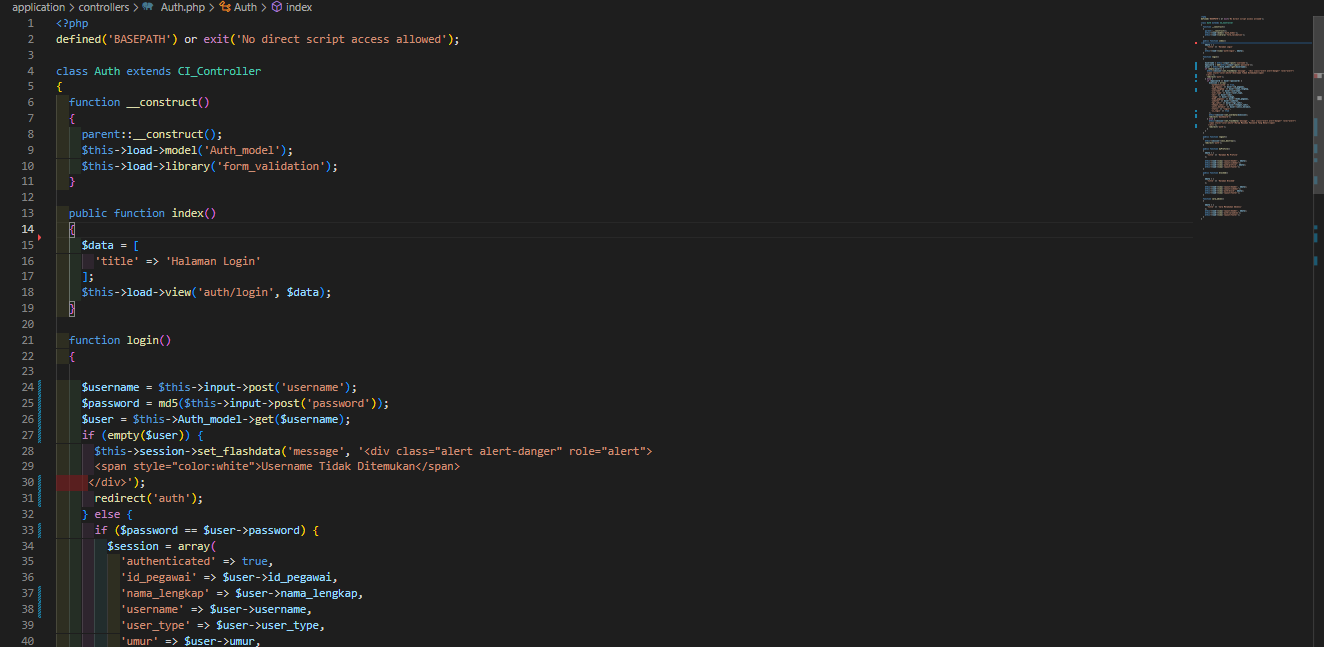
Gambar IV. 54 Implementasi *code controller* presensi

* + - 1. **Implementasi Code Controller Admin**



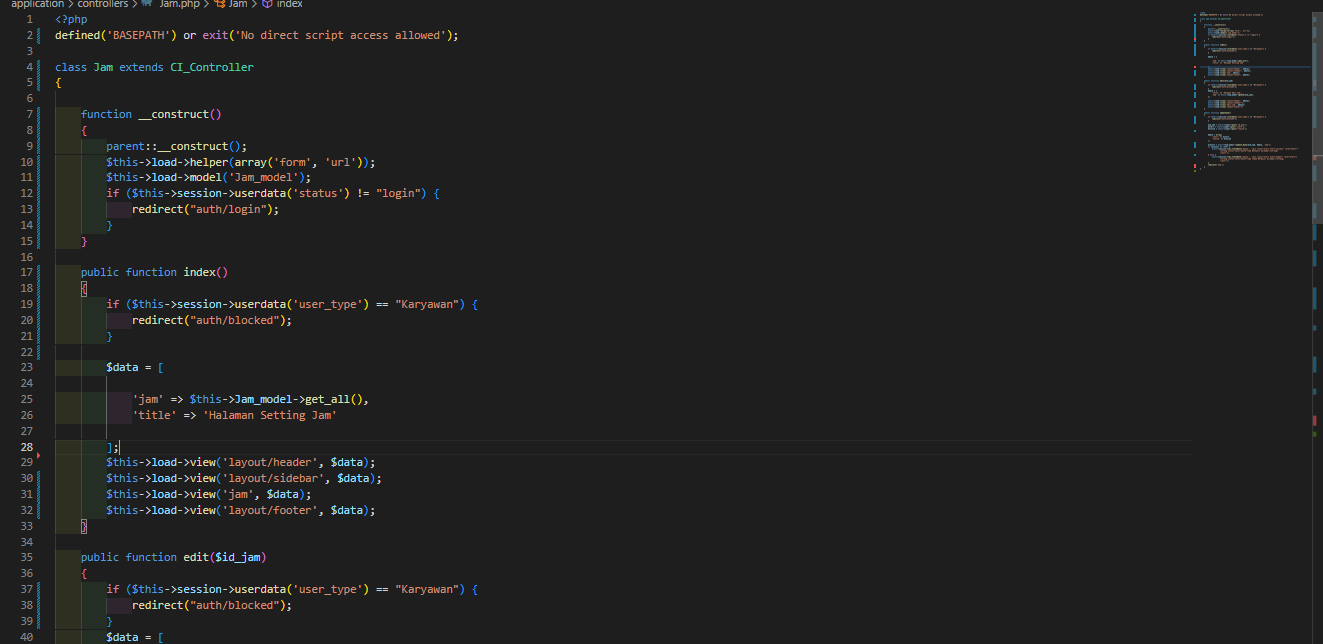
Gambar IV. 55 Implementasi *code controller admin*

* + - 1. **Implementasi Code Controller Auth**



Gambar IV. 56 Implementasi *code controller auth*

* + - 1. **Implementasi Code Controller Jam**



Gambar IV. 57 Implementasi *code controller* jam

* + - 1. **Implementasi Code Model Presensi**



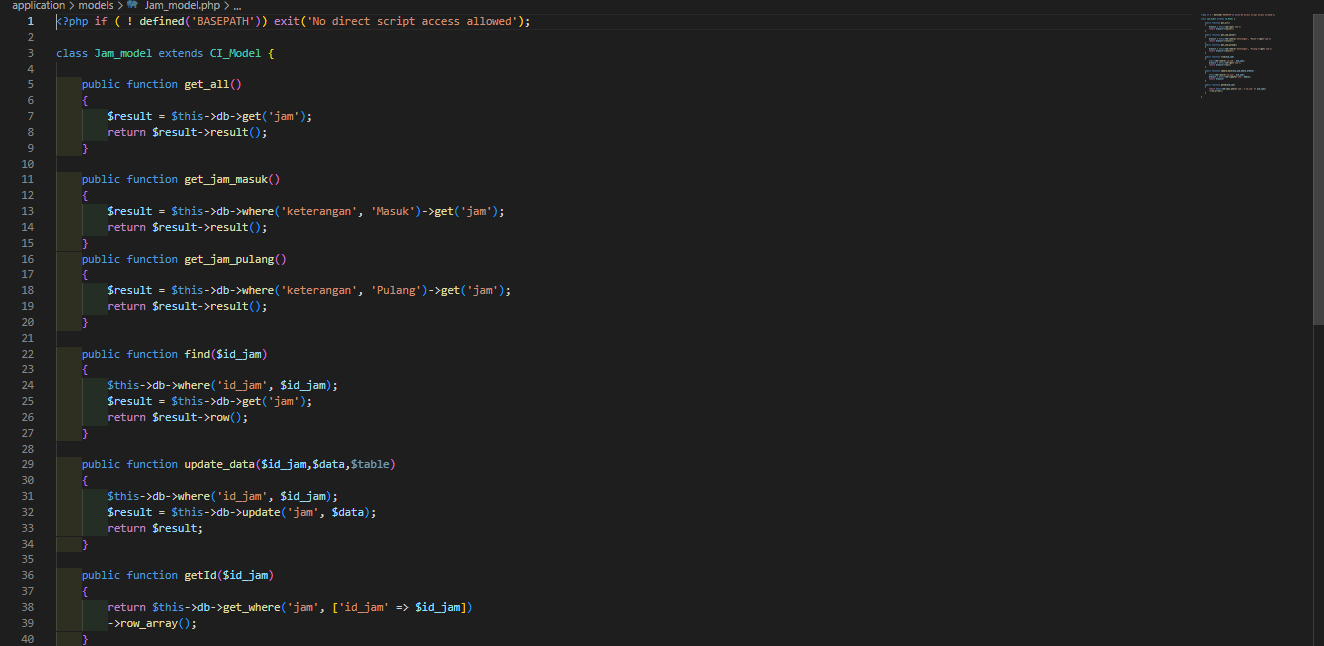
Gambar IV. 58 Implementasi *code model* presensi

* + - 1. **Implementasi Code Model Auth**



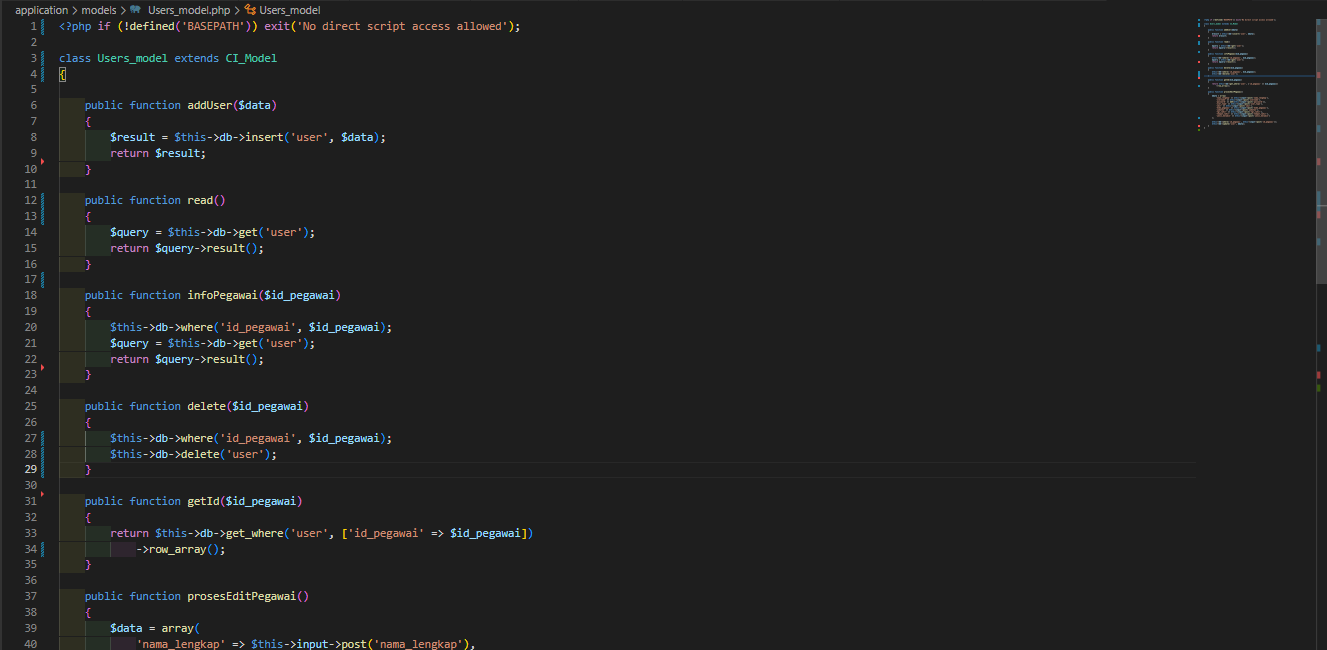
Gambar IV. 59 Implementasi *code model auth*

* + - 1. **Implementasi Code Model Jam**



Gambar IV. 60 Implementasi *code model* jam

* + - 1. **Implementasi Code Model User**



Gambar IV. 61 Implementasi *code model user*

1. Test Model
2. Rencana Pengujian

Pengujian dilakukan pada semua halaman antarmuka untuk mengetahui apakah fungsi-fungsi yang ada pada setiap halaman berfungsi sebagaimana mestinya dan sesuai dengan desain yang dibuat. Penjelasan tentang pengujian dijelaskan pada tabel dibawah ini.

Tabel IV. 25 Pengujian Sistem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Requirement*** | **Nama Pengujian** | **Deskripsi Pengujian** | **Pengguna** |
| **U-01** | *Login* | Mengisi *form login* dengan benar | Mengisi *username* dan *password* yang valid | *HRD dan* Karyawan |
| *Form login* belum di isi atau salah | Tidak mengisi *Form login* atau mengisi dengan *Username* dan *Password* yang tidak valid |
| **U-02** | Isi Presensi | Mengisi presensi kehadiran | 1. *User* mengakses halaman *dashboard* 2. *User* mengizinkan *popup* lokasi dan mengisi presensi beserta keterangan presensi-nya | *HRD* dan Karyawan |
| **U-03** | Cek Lokasi | Cek *marker* lokasi *maps* | 1. *User* mengakses halaman *dashboard* dan klik tombol cek lokasi anda 2. *User* mengizinkan *popup* lokasi setelah itu *marker* lokasi akan terdeteksi | *HRD* dan Karyawan |
| **U-04** | Data Kehadiran Sendiri | Melihat data kehadiran sendiri | 1. *User* mengakses halaman data kehadiran 2. Akan tampil tabel data kehadiran beserta tombol detailnya berdasarkan *user* yang sedang *login* | *HRD* dan Karyawan |
| **U-05** | Kelola Data Pegawai | Melakukan *CRUD* pada halaman Data Pegawai | 1. *HRD* mengakses halaman Data Pegawai 2. *HRD* dapat melakukan *create, read, update,* dan delete data pegawai | *HRD* |
| **U-06** | Kelola Data Presensi *User* | Melakukan *Read* dan *Delete* data presensi *user* | 1. *HRD* mengakses halaman Presensi Peagawai 2. *HRD* dapat melakukan *read* dan *delete* data presensi semua *user* | *HRD* |
| **U-07** | *Filter* Presensi *User* | Melakukan *Filter* pada data presensi *user* | 1. *HRD* mengakses halaman Presensi Pegawai 2. *HRD* mem-*filter* data presensi pegawai berdasarkan *form* nama karyawan dan *form* bulan tahun presensi. | *HRD* |
| **U-08** | *Export* Data Presensi ke *PDF* | Melakukan *Export* data presensi ke *pdf* | 1. *HRD* mengakses halaman Presensi Pegawai dan klik tombol *Export To PDF* 2. *HRD* dapat melakukan *Export* data presensi dengan cara *filter form* nama pegawai, dan *form* bulan tahun presensi atau *Export* semua data presensi. | *HRD* |
| **U-09** | *Setting* Jam Presensi | Melakukan *Setting* Jam Presensi Masuk dan Pulang | 1. *HRD* mengakses halaman *Setting* Jam 2. *HRD* dapat melakukan edit data jam masuk dan jam pulang untuk *user* yang akan melakukan presensi | *HRD* |
| **U-10** | *Profile* | Mengakses Halaman *Profile* | 1. *User* mengakses halaman *My Profile.* 2. *User* dapat melihat data *profile* sesuai data yang sedang *login.* | *HRD* dan Karyawan |

1. Implementasi *Code Controller*

Pengujian dilakukan pada fungsi-fungsi yang ada. Penjelasan tentang kasus dan hasil pengujian dijelaskan pada tabel dibawah ini.

Tabel IV. 26 Hasil pengujian *login*

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (*Login*) U-01** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| Mengisi *username* dan *password* secara valid | Jika data *login* valid, maka *user* akan masuk ke halaman *dashboard* | Berhasil |
| Tidak mengisi *form login* atau mengisi dengan *username* dan *password* yang tidak valid | Sistem menampilkan pesan kesalahan *login* gagal | Berhasil |

Tabel IV. 27 Hasil pengujian isi presensi

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (Isi Presensi) U-02** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *User* mengisi presensi dan keterangan presensi dengan mengizinkan *popup* lokasi | Presensi berhasil terisi dan disimpan ke *database* | Berhasil |
| *User* mengisi presensi dan keterangan presensi dengan tidak mengizinkan *popup* lokasi | Sistem menampilkan pesan gagal isi presensi | Berhasil |

Tabel IV. 28 Hasil pengujian cek lokasi

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (Cek Lokasi) U-03** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *User* melakukan cek lokasi dengan cara mengizinkan *popup* lokasi | Sistem berhasil mendeteksi lokasi dan memunculkan *marker* lokasi berada | Berhasil |
| *User* melakukan cek lokasi dan tidak mengizinkan *popup* lokasi | Sistem tidak mendeteksi lokasi | Berhasil |

Tabel IV. 29 Hasil pengujian data kehadiran sendiri

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (Data Kehadiran Sendiri) U-04** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *User* melihat data presensi sendiri | Sistem berhasil memunculkan data presensi *user* yang sedang *login* | Berhasil |

Tabel IV. 30 Hasil pengujian kelola data pegawai

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (Kelola Data Pegawai) U-05** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *HRD* mengakses halaman kelola data pegawai lalu melakukan *create, read, update* dan *delete* di halaman data pegawai | Sistem berhasil meng-*update* data pegawai | Berhasil |
| *HRD* tidak melakukan *create, read, update* dan *delete* di halaman data pegawai | Sistem tidak berhasil meng-*update* data pegawai | Berhasil |

Tabel IV. 31 Hasil pengujian kelola data presensi *user*

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (Kelola Data Presensi *User*) U-06** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *HRD* mengakses halaman presensi pegawaidan melakukan *read* dan *delete* | Sistem berhasil menampilkan data presensi *user* dan berhasil meng-*update* data presensi | Berhasil |
| *HRD* tidak melakukan *delete* data presensi *user* | Sistem tidak berhasil meng-*update* data presensi *user* | Berhasil |

Tabel IV. 32 Hasil pengujian *filter* presensi *user*

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (*Filter* Presensi *User*) U-07** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *HRD* melakukan *filter* presensi dengan cara mengisi *form* nama karyawan dan *form* bulan, tahun presensi | Sistem berhasil menampilkan data *filter* presensi *user* | Berhasil |
| *HRD* melakukan *filter* presensi tapi tidak mengisi *form* nama karyawan dan *form* bulan, tahun presensi | Sistem tidak berhasil menampilkan data *filter* presensi *user* | Berhasil |

Tabel IV. 33 Hasil pengujian *export* data presensi ke *pdf*

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (*Export* Data Presensi ke *PDF*) U-08** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *HRD* melakukan *export* data presensi dengan mengisi nama pegawai dan bulan tahun presensi | Sistem berhasil meng-*export* data presensi | Berhasil |
| *HRD* melakukan *export* semua data presensi | Sistem berhasil meng-*export* semua data presensi | Berhasil |
| *HRD* tidak melakukan *export* data presensi | Sistem tidak berhasil meng-*export* data presensi | Berhasil |

Tabel IV. 34 Hasil pengujian *setting* jam presensi

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (*Setting* Jam Presensi) U-09** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *HRD* melakukan edit pada jam presensi masuk atau pulang | Sistem berhasil meng-*update* jam presensi | Berhasil |
| *HRD* tidak melakukan edit pada jam presensi masuk atau pulang | Sistem tidak berhasil meng-*update* jam presensi | Berhasil |

Tabel IV. 35 Hasil pengujian *profile*

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji (*Profile*) U-10** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| *User* mengakses halaman *profile* | Sistem berhasil menampilkan data *profile user* yang sedang *login* | Berhasil |

# BAB V SIMPULAN DAN SARAN

1. **Simpulan**

Berdasarkan hasil penelitian ini, dapat disimpulkan beberapa hal mengenai Aplikasi Web Presensi Berbasis *GPS* dari kasus CV. Asian Technology Inspira yaitu:

1. Aplikasi Web Presensi Berbasis *GPS* merupakan sistem presensi yang dikembangkan dari sistem presensi CV. Asian Technology Inspira yang mempunya fitur isi presensi, cek lokasi, tampil data presensi sendiri, kelola data *user,* kelola data presensi *user, setting* jam presensi, meng-*export* data presensi ke *pdf,* dan meng-*filter* presensi *user.* *Role* pengguna pada aplikasi ada dua terdiri dari *HRD,* dan karyawan. Adanya berbagai macam fitur dapat membantu meningkatkan kualitas manajemen presensi dari CV. Asian Technologi Inspira.
2. Pembuatan Aplikasi Web Presensi Berbasis *GPS* ini dimulai dari pengumpulan data – data yang dibutuhkan dengan menggunakan metode observasi dan wawancara, pengembangan perangkat lunak menggunakan metode *Rapid Application Development* (RAD), perancangan *model* aplikasi berupa *Unified Modeling Language* (UML).
3. Implementasi aplikasi presensi ini menggunakan *Framework Code Igniter* dengan Bahasa pemograman PHP dan menggunakan *database MySQL* serta untuk *maps-*nya menggunakan *library javascript Leaflet* dengan *style CSS* menggunakan *framework Bootstrap.* Ketika aplikasi selesai dibuat, diuji dengan menggunakan metode *black box testing.*
4. Dengan adanya fitur *export* data presensi ke *pdf* diharapkan dapat mempermudah dan menghemat waktu *HRD* untuk membuat laporan presensi secara manual.
5. Dengan adanya fitur cek lokasi diharapkan dapat mempermudah staff untuk mendeteksi lokasinya sebelum nanti mengisi presensi.
6. Perancangan aplikasi ini hanya untuk perangkat berbasis website.
7. **Saran**

Sistem yang dibangun masih memiliki beberapa kekurangan, oleh karena itu ada beberapa hal yang bisa dikembangkan, antara lain:

1. Dapat ditambahkan fitur – fitur penunjang lainnya seperti penambahan perhitungan gaji serta diagram *chart* untuk presensi bulanan
2. Dibuatkan *mobile apps*-nyauntuk CV. Asian Technology Inspira agar lebih professional.
3. Dapat terintegrasi dengan *Api Google Calender* supaya dapat menyesuaikan dengan hari libur dan tanggal merah.

# DAFTAR PUSTAKA

Abbas, D. S., Hakim, M. Z & Rustandi, R. (2019). Pengaruh Profitabilitas, Solvabilitas, Opini Audit dan Reputasi Kantor Akuntan Publik Terhadap Audit Report Lag (Pada Perusahaan Manufaktur yang Terdaftar di Bursa Efek Indonesia Pada Tahun 2012-2015). COMPETITIVE, 3(3), 21–39.

Adhi Prasetio., 2014, Buku Sakti Webmaster (PHP Dan Mysql, HTML Dan CSS, HTML5 Dan CSS3, Javascipt). Jakarta: PT. Transmedia

Alatas, Husein. (2013). Responsive Web Design dengan PHP & Bootstrap. Yogyakarta:Lokomedia

B. Romney, S. & Steinbart, P. J. (2015). Sistem Informasi Akuntansi. Jakarta Selatan

Bastian, Indra. (2006): Akuntansi Sektor Publik. Jakarta: Erlangga.

Iskandaria (2012), blackbox (blackbox testing), metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas.

Kustiyahningsih, Yeni., Devie Rosa Anamisa. 2011. Pemrograman Basis Data Berbasis Web Menggunakan PHP & MySQL. Yogyakarta: Graha Ilmu.

Milner, G. (2016): What Is GPS. Journal of Technology In Human Services, Vol. 34, N.

Meliyanti, Fera. (2015): Efektivitas Penggunaan Leaflet Terhadap Peningkatan Pengetahuan Remaja Kelas Viii Tentang Hiv/aids Di Smp Negeri 2 Ogan Komering Ulu, Vol. 4, No. 2

Mulyawan, r. (2018, Desember 17). Mengenal Pengertian Database: Sejarah, Fungsi, Contoh dan Perbedaannya dengan Spreadsheets!

Muslimah, E., & Supriyono. (2018): Perancangan Sistem Informasi Manajemen Kas Berbasis Web Studi Kasus : RS dr . Etty Asharto Batu. 10(1), 21–25.

Nugroho, Bunafit. (2013): Dasar Pemrograman Web *PHP-MySql* dengan Dreamweaver. Yogyakarta: Gava Media.

Perkasa, P. (2019): Penggunaan Global Positioning System (Gps) Untuk Dasar Survey Pada Mahasiswa. Jurnal Pendidikan Teknologi Dan Kejuruan Balanga, Vol. 7 No. 1.

Raharjo, Budi dkk. (2014). Modul Pemrograman Web HTML, PHP, dan MySQL. Bandung: Modula.

Raharjo, Budi. 2015. Belajar Otodidak MySql. Bandung: Informatika.

Rosa A.S dan M.Shalahudin .2014. Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Informatika

Sri Mulyani. 2016, "Metode Analisis dan Perancangan Sistem Edisi Kedua". Bandung, Indonesia: Abdi Sistematika.

Sukamto, & Shalahuddin. (2013). Analisa dan Desain Sistem Informasi. Yogyakarta: Andi Offset

Wahyudi, Dian Pebri. 2014. Dasar-Dasar Teknik Komputer dan Informatika. Jakarta: Inti Prima Promosindo

Wibowo, Agus. 2015. Pendidikan Karakter Berbasis Kearifan Lokal di Sekolah. Yogyakarta: Pustaka Pelajar.

Yuniati et al (2017). Implementasi Modul *Global Positioning System (GPS)* pada *Sistem Tracking Bus Rapid Transit (BRT)* Lampung Menuju *Smart Transportation.* Lampung

# LAMPIRAN A: KODE PROGRAM CONTROLLER PRESENSI MASUK DAN PULANG

public function absen()

{

$today = $this->get\_today\_date;

$jammasuk = $this->jamMasuk;

$jampulang = $this->jamPulang;

$absen\_harian = $this->Absensi\_model->absen\_harian\_user($this->session->userdata('id\_pegawai'))->num\_rows();

if ($this->input->post('masuk')) {

$data = [

'tgl\_absen' => date('Y-m-d'),

'jam\_absen' => date('H:i:s'),

'keterangan\_absen'=>$this->input ->post('keterangan\_absen'),

'lat\_absen' => $this->input->post('lat\_absen'),

'long\_absen' => $this->input->post('long\_absen'),

'id\_pegawai' => $this->session->id\_pegawai,

'status\_absen' => 1

];

$result = $this->db->insert('absensi', $data);

if ($result == TRUE) {

$this->session->set\_flashdata('pesan', '<div class="alert alert-success d-flex align-items-center text-white" role="alert">

<svg class="bi flex-shrink-0 me-2" width="24" height="24" role="img" aria-label="Success:"><use xlink:href="#check-circle-fill"/></svg>

<div>

Selamat Absen Masuk Anda Sudah Dicatat

</div>

</div>');

} else {

$this->session->set\_flashdata('pesan', '<div class="alert alert-warning d-flex align-items-center" role="alert">

<svg class="bi flex-shrink-0 me-2" width="24" height="24" role="img" aria-label="Warning:"><use xlink:href="#exclamation-triangle-fill"/></svg>

<div>

Maaf, Data Anda Belum Dicatat

</div>

</div>');

}

redirect('absensi');

} elseif ($this->input->post('pulang')) {

$data = [

'jam\_absen\_pulang' => date('H:i:s'),

'status\_absen' => 2

];

$id\_absen = $this->input->post('id\_absen');

$id\_pegawai = $this->session->id\_pegawai;

$this->db->where('tgl\_absen', date('Y-m-d'));

$this->db->where('id\_pegawai', $id\_pegawai);

$result2 = $this->db->update('absensi', $data);

if ($result2 == TRUE) {

$this->session->set\_flashdata('pesan', '<div class="alert alert-success d-flex align-items-center text-white" role="alert">

<svg class="bi flex-shrink-0 me-2" width="24" height="24" role="img" aria-label="Success:"><use xlink:href="#check-circle-fill"/></svg>

<div>

Selamat Absen Pulang Anda Sudah Dicatat

</div>

</div>');

} else {

$this->session->set\_flashdata('pesan', '<div class="alert alert-warning d-flex align-items-center" role="alert">

<svg class="bi flex-shrink-0 me-2" width="24" height="24" role="img" aria-label="Warning:"><use xlink:href="#exclamation-triangle-fill"/></svg>

<div>

Maaf, Data Anda Belum Dicatat

</div>

</div>');

}

redirect('absensi');

}

}

# LAMPIRAN B: KODE PROGRAM CONTROLLER DETAIL PRESENSI

public function detailKehadiranPegawai($id\_absen)

{

if ($this->session->userdata('user\_type') == "Karyawan") {

redirect("auth/blocked");

}

$data = array(

'title' => 'Data Detail Kehadiran Anda',

'user' => $this->Absensi\_model->getUserDetailPegawai($id\_absen)->result()

);

$this->load->view('layout/header', $data);

$this->load->view('layout/sidebar');

$this->load->view('absen/detailkehadiranPegawai', $data);

$this->load->view('layout/footer');

}

# LAMPIRAN C: KODE PROGRAM CONTROLLER HAPUS PRESENSI

function deleteAbsensi($id\_absen)

{

if ($this->session->userdata('user\_type') == "Karyawan") {

redirect("auth/blocked");

}

$this->Absensi\_model->delete\_absen($id\_absen);

if ($this->db->affected\_rows() > 0) {

$this->session->set\_flashdata('psn', '<div class="alert alert-success" role="alert">

<strong style="color:white">Data Absensi Berhasil Dihapus!</strong>

</div>');

} else {

$this->session->set\_flashdata('psn', '<div class="alert alert-success" role="alert">

<strong style="color:white">Data Gagal Dihapus!</strong>

</div>');

}

redirect('Absensi/dataKehadiran');

}

# LAMPIRAN D: KODE PROGRAM CONTROLLER EXPORT PRESENSI

function do\_export()

{

if ($this->session->userdata('user\_type') == "Karyawan") {

redirect("auth/blocked");

}

error\_reporting(0); // AGAR ERROR MASALAH VERSI PHP TIDAK MUNCUL

$pdf = new FPDF('L', 'mm', 'A4');

$pdf->AddPage();

$pdf->SetFont('Arial', 'B', 16);

$pdf->Cell(0, 7, 'DAFTAR PRESENSI CV. ASIAN TEKNOLOGI INSPIRA', 0, 1, 'C');

$pdf->Cell(10, 7, '', 0, 1);

$pdf->SetFont('Arial', 'B', 10);

$pdf->Cell(10, 6, 'No', 1, 0, 'C');

$pdf->Cell(40, 6, 'Nama Pegawai', 1, 0, 'C');

$pdf->Cell(40, 6, 'Tanggal Absen', 1, 0, 'C');

$pdf->Cell(40, 6, 'Jam Datang', 1, 0, 'C');

$pdf->Cell(40, 6, 'Jam Pulang', 1, 0, 'C');

$pdf->Cell(40, 6, 'Status Kehadiran', 1, 0, 'C');

$pdf->Cell(40, 6, 'Latitude Maps', 1, 0, 'C');

$pdf->Cell(40, 6, 'Longtitude Maps', 1, 1, 'C');

$pdf->SetFont('Arial', '', 10);

$id\_pegawai = $this->input->post('id\_pegawai');

$tanggal = $this->input->post('tanggal');

$vbulan = date("m", strtotime($tanggal));

$no = 0;

$this->db->select('\*');

$this->db->from('absensi');

$this->db->join('user', 'absensi.id\_pegawai = user.id\_pegawai');

$this->db->where('absensi.id\_pegawai', $id\_pegawai);

$this->db->where('month(absensi.tgl\_absen)', $vbulan);

$this->db->where('year(absensi.tgl\_absen)', $tanggal);

$pegawai = $this->db->get()->result();

$i = 1;

foreach ($pegawai as $data) {

$date = $data->tgl\_absen;

$pdf->Cell(10, 6, $i++, 1, 0, 'C');

$pdf->Cell(40, 6, $data->username, 1, 0);

$pdf->Cell(40, 6, $date, 1, 0);

$pdf->Cell(40, 6, $data->jam\_absen, 1, 0);

$pdf->Cell(40, 6, $data->jam\_absen\_pulang, 1, 0);

$pdf->Cell(40, 6, $data->keterangan\_absen, 1, 0);

$pdf->Cell(40, 6, $data->lat\_absen, 1, 0);

$pdf->Cell(40, 6, $data->long\_absen, 1, 1);

}

$pdf->Output();

}

# LAMPIRAN E: KODE PROGRAM CONTROLLER FILTER PRESENSI

public function filter()

{

if ($this->session->userdata('user\_type') == "Karyawan") {

redirect("auth/blocked");

}

$id\_pegawai = $this->input->post('id\_pegawai');

$tanggal = $this->input->post('tanggal');

$vbulan = date("m", strtotime($tanggal));

if ($this->input->post('tanggal') == '') {

$data = [

'title' => 'Halaman Filter Data Absensi',

'nama' => $this->db->get('user')->result(),

'fil' => $this->Absensi\_model->filter\_name($id\_pegawai)->result(),

'identitas' => $this->db->get('user')->row(),

'inp' => $this->input->post('tanggal'),

'namaselect' => $this->Absensi\_model->select\_name\_filter($id\_pegawai)->row()

];

} else {

$data = [

'title' => 'Halaman Filter Data Absensi',

'nama' => $this->db->get('user')->result(),

'fil' => $this->Absensi\_model->filter\_username($id\_pegawai, $tanggal, $vbulan)->result(),

'countPresensi' => $this->Absensi\_model->countPresensi($id\_pegawai, $tanggal, $vbulan),

'countSakit' => $this->Absensi\_model->countSakit($id\_pegawai, $tanggal, $vbulan),

'countCuti' => $this->Absensi\_model->countCuti($id\_pegawai, $tanggal, $vbulan),

'identitas' => $this->db->get('user')->row(),

'inp' => $this->input->post('tanggal'),

'namaselect' => $this->Absensi\_model->select\_name\_filter($id\_pegawai)->row()

];

}