BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan Teknologi Informasi sekarang ini sudah menjangkau ke semua bidang, baik dalam bidang keamanan, bidang pendidikan, dan bidang perkantoran, serta segi-segi kehidupan manusia lainnya. Kebutuhan informasi yang akurat, tepat dan cepat dalam menyajikan data yang sangat lengkap merupakan salah satu tujuan penting, untuk ini *smartphone* berperan aktif dalam segala bidang dan akan mempermudah pekerjaan manusia. Informasi sangat penting sekali bagi semua orang, dengan adanya informasi akan terjadi pula timbal balik pada kemajuan baik di segala bidang. Dari adanya sebuah kemajuan tersebut maka akan terjadi sebuah perubahan, saat ini dunia sedang mengalami proses revolusi penerapan teknologi komputer yang disebut dengan digitalisasi.

Dengan adanya *smartphone android* yang saat ini begitu banyak, memungkinkan beberapa perusahaan memperbaharui sistemnya menggunakan *smartphone android*. Karena lebih mudah pengoperasiannya dan dapat melakukan absensi dimana saja sehingga waktu yang dipergunakan tidak banyak terbuang. Dalam hal penggunaan *smartphone android* sangatlah mudah, hampir lapisan masyarakat memiliki *smartphone android*, karena *smartphone android* memiliki banyak manfaat apabila dipakai untuk hal positif. Oleh karena itu, *smartphone android* diminati untuk mempermudah urusan seseorang.

Begitupun dengan sistem absensi, butuh pembaharuan yang sangat baik agar karyawan dapat absen dengan cepat dan tidak perlu antri untuk absen. Solusi yang terbaik adalah dengan menggunakan *smartphone android.* Sistem absensi dengan cara ini juga dapat memaksimalkan waktu perkerjaan dalam suatu perusahaan dibanding karyawan harus antri didepan alat *finger print*. Hal ini sangat membuang waktu perkerjaan. Oleh sebab itu, penggunaan *android* sangatlah diperlukan dalam pembaharuan sistem absensi.

Apabila perusahaan beralih dengan menggunakan *smartphone android*, perusahaan hanya perlu menyiapkan server untuk mengendalikan/ mengatur data yang masuk dan pengaturan jam masuk dan pulang. Dan perlu juga jaringan *WIFI* lokal untuk menghubungkan *smartphone android* dengan *server*, jaringan ini juga diatur agar jangkauannya hanya wilayah perusahaan saja karena untuk menghindari karyawan curang yang absen dari luar perusahaan.

Absensi dapat dikatakan suatu pendataan kehadiran yang merupakan bagian dari aktivitas pelaporan yang ada dalam sebuah institusi. Absensi disusun dan diatur sehingga mudah untuk dicari dan dipergunakan ketika diperlukan oleh pihak yang berkepentingan (Setiawan dan Kurniawan : 2015).

Bosowa Corp adalah perusahaan induk yang memiliki berbagai bidang bisnis mulai dari semen hingga *dealer* mobil, yang mempunyai anak perusahaan di berbagai daerah. Salah satu anak perusahaan ini berada di kota Parepare dan bergerak pada bidang *dealer* mobil (Mitsubishi). Bosowa Parepare saat ini menerapkan absensi berupa alat *finger print* yang tersedia di perusahaan tersebut. Hal ini menimbulkan kebocoran waktu dan pelanggaran lainnya. Kedisiplinan

karyawan sangat mendukung keberhasilan suatu perusahaan dalam mencapai tujuannya. Dalam kesehariannya, sistem absensi karyawan di perusahaan masih dengan cara mendeteksi sidik jari pada sebuah alat, yang nantinya akan direkap dengan cara meng*ekspor* data dari alat *finger print* menuju komputer dengan menggunakan *flashdisk* atau USB. Rekap data dengan cara seperti ini memakan banyak waktu dan hanya dapat di akses oleh admin yang bertugas saja, sehingga karyawan tidak dapat melihat data absensi mereka.

Kelemahan lainnya dari proses rekapitulasi data adalah data kehadiran yang terekam dari mesin absensi *finger print* tidak *real time* dan tetap memerlukan rekapitulasi secara manual walau data yang digunakan sudah berbentuk *spreadsheet*. Karena rekapitulasi data berjalan manual dan tidak *real time*, maka hal tersebut menyebabkan kemungkinan besar adanya celah untuk memanipulasi data kehadiran karyawan. Disamping itu, jam kehadiran karyawan menentukan kedisiplinan dan komitmen kerja karyawan. File data *spreadsheet* dapat diubah-ubah oleh oknum tertentu sehingga jam kehadiran dari karyawan pun tidak valid.

Selain itu, proses pengabsensian di Bosowa Parepare dapat dikatakan sangat beresiko di tengah keadaan virus yang sedang menimpa dunia. Penularan virus *covid-19* dapat menyebar hanya dengan sentuhan benda-benda yang telah terinveksi. Mesin *finger print* bisa menjadi media penularan paling tidak disadari, sebab layar pada mesin *finger print* disentuh banyak orang secara bergantian.

Mesin *finger print* juga membutuhkan perawatan yang rutin supaya mesin selalu berjalan maksimal. Sebab, jika kondisi mesin *finger print* memburuk dan

rusak tiba-tiba, hal ini akan menghambat proses pencatatan absensi. Layar *scanner* pada mesin juga harus tetap bersih dan cara membersihkannya pun tidak boleh sembarangan, harus dengan cara yang tepat.

Pemanfaatan sistem teknologi informasi yang semakin maju dapat mengatasi hal–hal yang tidak diinginkan diatas dengan adanya sistem absensi yang digitalisasi dapat memberikan pencapaian kinerja pada perusahaan tersebut. Dengan keadaan sekarang yang serba praktis, maka penulis mengusulkan judul **"Aplikasi Absensi Karyawan Menggunakan *Geolocation* dan *Finger Print* Berbasis *Android*"**. Penggunaan aplikasi sistem absensi karyawan ini juga akan membuat sistem absensi ini menjadi lebih efektif dan efisien.

B. Rumusan Masalah

Berdasarkan uraian latar belakang yang telah diuraikan, maka akan dibuat suatu penelitian pada masalah yang sedang dihadapi oleh perusahaan Bosowa Parepare. Adapun latar belakang masalah yang dapat diidentifikasikan sebagai berikut :

1. Bagaimana menganalisa, merancang, dan membangun suatu aplikasi absensi karyawan menggunakan *geolocation* dan *finger print* berbasis *Android*?
2. Bagaimana menerapkan *geolocation* pada aplikasi absensi di Bosowa Parepare yang dapat meningkatkan efektifitas sistem absensi karyawan?
3. Bagaimana menerapkan sistem *finger print* pada *smartphone* untuk disematkan pada aplikasi absensi karyawan berbasis *Android*?

C. Batasan Masalah

Adapun agar pembahasan menjadi lebih terarah dan tidak menyimpang dan juga sesuai dengan latar belakang yang sudah diuraikan sebelumnya, maka penulis membatasi permasalahan penelitian pada Aplikasi Absensi Karyawan Menggunakan *Geolocation* dan *Finger Print* Berbasis *Android*. Maka penulis membatasi masalah sebagai berikut :

1. Ruang lingkup penelitian ini difokuskan atau terpusat di Bosowa Parepare, dalam hal ini sumber data diperoleh dari perusahaan tersebut.
2. Penelitian ini berfokus pada pengelolaan data absensi karyawan, sistem ini tidak menangani masalah penggajian, akuntansi dan kinerja karyawan.
3. Aplikasi ini akan dibuat dengan menggunakan Bahasa pemograman *PHP Javascript* dan *HTML* dengan integrasi *DBMS MySQL*.
4. Aplikasi ini mendeteksi GPS *Handphone* karyawan, bukan orang/karyawan itu sendiri.

D. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah didefinisikan sebelumnya tujuan dari penelitian dan perancangan sistem absensi karyawan dalam penelitian ini adalah :

1. Membangun dan merancang suatu aplikasi absensi karyawan dengan menggunakan *geolocation* berbasis *Android.*
2. Menerapkan *geolocation* untuk sistem absensi karyawan pada Bosowa Parepare.

3. Mengetahui bagaimana menerapkan sistem *finger print* pada *smartphone* untuk disematkan pada aplikasi absensi berbasis *Android*.

E. Manfaat Penelitian

Adapun Salah satu fungsi dari penelitian adalah bisa memberikan manfaat dan kegunaan yang secara nyata akan diterima oleh berbagai pihak yang terkait dengan obyek penelitian seperti aspek akademis dan praktis. Dalam penelitian kali ini peneliti berharap bisa memberikan kegunaan, antara lain :

1. Dalam Bidang Akademik
    a. Untuk mengaplikasikan teori yang di dapat di bangku perkuliahan dengan kondisi real di lapangan, khususnya terkait dengan permasalahan yang dibahas terkait penelitian dan *tools* atau metode penyelesaian masalah.
    b. Diharapkan Sebagai bahan masukan bagi para pengkaji ilmu komunikasi di bidang *Public Relations* yang berminat untuk meneliti masalah yang sama.
2. Bagi Penulis
    a. Meningkatkan kemampuan berfikir tentang konsep seperti apa yang harus penulis angkat kedalam penelitian, tentunya berkaitan dekat dengan keadaan di perusahaan yang menjadi tempat penelitian.
    b. Mengimplementasikan ilmu yang penulis dapatkan selama proses perkuliahan.
    c. Sebagai tugas akhir untuk mendapatkan gelar sarjana strata 1 (S.1) pada Program Studi Teknik Informatika, Universitas Muhammadiyah Parepare.

3. Bagi Perusahaan
    a. Menciptakan sistem absensi karyawan yang modern dengan menggunakan pemanfaatan *geolocation* dan *finger print* di perusahaan.
    b. Membangun sebuah sistem absensi di Bosowa Parepare yang akan memberikan kemudahan dan pegelolaan data yang lebih baik dan detail kepada pengunjung peserta dan pelaksana operasional di perusahaan tersebut.
    c. Program aplikasi yang dibuat juga akan sangat bermanfaat bagi perusahaan, bahkan seluruh Indonesia dan dapat dijadikan bahan untuk penelitian lebih lanjut di bidang yang berkaitan.
    d. Sebagai sumber informasi bagi pelaku pengambil kebijakan untuk memonitoring kehadiran karyawan di perusahaan.

F. Sistematika Penulisan

Sistematika penulisan skripsi ini dibagi ke dalam enam bab yang masing-masing menjelaskan bagian-bagian yang berbeda namun tetap memiliki korelasi satu dengan yang lain. Setiap bab-nya terdiri dari beberapa sub tersendiri. Secara singkat akan diuraikan sebagai berikut.

BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan tentang latar belakang penulisan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini berisi uraian tentang landasan teori-teori dari perancangan "Aplikasi Absensi Karyawan Menggunakan *Geolocation* Berbasis *Android*". Bab ini juga membahas mengenai berbagai teori yang mendasari dari analisis permasalahan yang berhubungan dengan topik yang dibahas.

BAB III : METODOLOGI PENELITIAN

Dalam bab ini membahas mengenai metode penelitian yang digunakan dalam merancang dan membangun sistem. Pada bab ini juga membahas tentang identifikasi masalah, analisis kebutuhan aplikasi yang meliputi kebutuhan masukan, proses dan keluaran aplikasi yang akan dibangun. Penjelasan mengenai proses perancangan dengan menggunakan rancangan dan UML.

BAB IV : ANALISIS DAN PERANCANGAN SISTEM

Pada Pada bab ini membahas mengenai hasil dari tahap analisis dan perancangan sesuai dengan metode yang digunakan pada alat dan sistem yang dibuat serta hasil dari pengujian.

BAB V : IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini berisi tentang implementasi dan pengujian dua unit testing ialah pengujian *White-Box* dan *Black-Box*. Pada bab ini

menjelaskan bentuk-bentuk implementasi aplikasi beserta penerapannya.

BAB VI : PENUTUP

Pada bab akhir ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran-saran yang diberikan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih maksimal.

DAFTAR PUSTAKA

Pada bagian ini akan dipaparkan tentang sumber-sumber literatur, tutorial, buku maupun situs-situs yang digunakan dalam pembuatan laporan tugas akhir ini.

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Berbagai penelitian sebelumnya merupakan hal yang sangat perlu dan dapat dijadikan sebagai data pendukung. Salah satu data pendukung yang menurut peneliti perlu dijadikan bagian tersendiri adalah penelitian terdahulu yang relevan dengan permasalahan yang sedang dibahas dalam penelitian ini. Oleh karena itu, peneliti melakukan kajian terhadap beberapa hasil penelitian berupa tugas akhir/skripsi atau jurnal melalui *internet*.

1. Dhani Wibawa (2016) dengan skripsi yang berjudul **"Aplikasi *Presensi* Kepegawaian Kelurahan Menggunakan Teknologi *Fingerprint* (Studi kasus di Kantor Kelurahan Sindumartani Ngemplak Sleman Yogyakarta)"**. Berdasarkan hasil penelitian yang dilakukan oleh peneliti tersebut kriteria yang digunakan yaitu pegawai dari kelurahan, dengan melakukan pendaftaran atau mengisikan biodata pada aplikasi dengan teknologi *fingerprint*/ sidik jari, selanjutnya sistem presensi pegawai dengan mesin sidik jari dapat melakukan absensi pegawai kelurahan untuk harian maupun bulanan.
2. Ely *Mulyadi, Andis Trihariprasetya, I Gede Wiryawan* (2020) dengan skripsi yang berjudul **"Penerapan Sistem *Presensi Mobile* dengan Menggunakan Sensor GPS (Klinik Pratama X di Jember)"**. Peneliti diatas membuat salah satu solusi

dari permasalahan pada klinik pratama yakni sistem *presensi* ini diterapkan pada salah satu klinik kesehatan di Jember. Implementasi sistem *presensi* yang dikembangkan secara umum dapat berjalan dengan baik dan dibuktikan dengan hasil pengujian dengan metode *Black-box* dan kuisioner sederhana untuk pengguna sistem. Hal yang masih perlu menjadi pertimbangan untuk pengembangan kedepannya adalah kecepatan respon dari sistem presensi ini.

3. Diana (2017) dengan jurnal yang berjudul **"Perancangan Sistem Informasi Absensi Guru Dan Siswa Berbasis *Web* Di *Sweet School* Batam"**. Peneliti diatas merancang sebuah sistem untuk mempermudahkan orang tua siswa-siswi di *sweet school* dalam melakukan kegiatan pengisian data absensi guru dan siswa, mata pelajaran, ruangan kelas, dan mendapatkan informasi data nilai siswa, selain itu dengan berbasiskan *web* maka informasi data dapat diakses dengan waktu dan tempat yang tidak ditentukan.

Pada ketiga penelitian terkait diatas memanfaatkan teknologi internet berbasis *online* untuk mendapatkan saja, dari tempat tertentu, tanpa adanya pemberitahuan secara langsung, jadi proses kontrolnya tidak langsung terjadi, masih ada waktu selang. Sehingga kemungkinan terjadi manipulasi data yang dikirim. Jika dibandingkan dengan penelitian saat ini, tentunya lebih unggul aplikasi ini, karena sistem ini menerapkan teknologi *Geolocation* dan *Finger Print* untuk aplikasi absensi karyawan.

B. Tinjauan Teori

1. Aplikasi (*Application*)

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja to *apply* yang dalam Bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu *sub* kelas perangkat lunak komputer yang menggunakan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Menurut Kamus Besar Bahasa Indonesia (KBBI, 2015) Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Sementara menurut Hartono (1999) aplikasi adalah penggunaan dalam suatu komputer, insturksi *(instruction)* atau pernyataan *(statement)* yang disusun sedemikian rupa sehingga komputer dapat memproses *Input* menjadi *output*.

2. Absensi

Absensi atau kartu jam hadir ialah dokumen yang mencatat jam hadir setiap karyawan di perusahaan. Catatan jam hadir karyawan tersebut dapat berupa daftar hadir biasa, dapat juga pula berbentuk kartu hadir yang diisi dengan mesin pencatat waktu. Pekerjaan mencatat waktu pada dasarnya dapat dipisahkan menjadi 2 (dua) bagian yakni pencatatan waktu hadir *(attendance time keeping)* dan juga pencatatan waktu kerja. Pencatatan jam hadir pada hadir kartu jam hadir yang dilakukan oleh pada setiap pegawai atau pekerja bisa mempengaruhi gaji bersih atau *take home pay* yang akan diterima oleh si pegawai atau pekerja setiap

bulannya. Karena jika pegawai atau pekerja lupa ataupun tidak mencatatkan jam hadirnya pada kartu jam hadir akan dapat mempengaruhi komponen-komponen yang ada pada gaji, terutama sekali pada pos tunjangan, dikarena tunjangan yang diberikan perusahaan pada setiap pegawai atau pekerja tergantung dari beberapa banyak pegawai atau pekerja hadir pada jam kerja. Seperti tunjangan makan dan juga transportasi.

Apabila pegawai atau pekerja tidak mencatatkan jam hadirnya pada kartu jam hadir maka tunjangan makan dan hingga transpotasinya yang diterima pegawai atau pekerja setiap bulannya akan berkurang dan juga akan mempengaruhi gaji bersih yang diterima pegawai atau pekerja tersebut. Pencatatan waktu hadir tersebut dimaksudkan untuk mengumpulkan data mengenai jumlah jam hadir pegawai atau pekerja dalam suatu periode pembayaran dan juga kadang-kadang juga tentang tarif upah untuk pekerjaan yang dilakukan. Pencatatan waktu kerja dimaksudkan untuk dapat mencatat jam kerja sesungguhnya yang digunakan oleh pegawai atau pekerja dalam setiap pekerjaan *(job)* atau departemennya.

3. *Geolocation*

Geolocation adalah teknologi yang menggunakan data yang diperoleh dari komputer atau perangkat seluler individu untuk mengidentifikasi atau mendeskripsikan lokasi fisik aktual pengguna. Dua jenis data dapat dikumpulkan yaitu dapat berupa informasi berbasis pengguna/perangkat aktif dan data berbasis/korelasi berbasis server pasif dan kemudian saling direferensikan

terhadap satu sama lain untuk menciptakan hasil yang paling akurat. (ISACA, 2016)

Geolocation membantu dalam mengidentifikasi lokasi pengguna, yang dapat digunakan untuk memberikan informasi spesifik lokasi atau rincian navigasi posisi pengguna. *Geolocation* melacak keberadaan perangkat yang digunakan dan disimpulkan dengan memanfaatkan RFID, WiFi dan *Bluetooth* MAC alamat, dan GPS/ CDMA ID sel, serta *Input* pengguna. Namun *Geolocation* ini bisa diajak kompromi, artinya selama pengguna tidak menyetujui, posisi tidak tersedia hal ini untuk menjaga *privacy* dari pengunjung. (Widyatmoko, 2012)

4. *Finger Print*

a. Pengertian *Finger Print*

Fingerprint adalah sebuah alat untuk memenuhi kebutuhan data yang cepat dengan menggunakan verifikasi sidik jari. Sebelum sensor *Fingerprint* ditemukan, dahulu sebuah data diamankan dengan menggunakan *password* atau ID, ada juga yang menggunakan pola guna mengamankan suatu data.

Dahulu *Fingerprint* hanya digunakan pada peralatan elektronik seperti untuk mengamankan pintu dan untuk absensi karyawan. Namun semakin berkembangnya zaman, kini *FIngerprint* sudah diterapkan unutk mengamankan data yang ada di *smartphone* setiap penggunanya.

Sedangkan pada penerapan alat absensi karyawan, *Fingerprint* berguna untuk meminimalisir kecurangan karyawan yang bisa mengakibatkan kerugian perusahaan. Sehingga dengan adanya *Fingerprint*, setiap karyawan bertanggung jawab atas kehadirannya sendiri di kantor.

Selain itu penerapan *Fingerprint* pada sebuah pintu seperti brankas juga meminimalisir akses, karena dengan *Fingerprint* akses dari suatu pintu brankas hanya bisa dibuka oleh orang-orang tertentu saja, bahkan hanya untuk satu orang saja, Sehingga keamanan barang yang berada di dalamnya bisa dijamin.

b. Fungsi *Finger Print*

Secara umum fungsi dari *Fingerprint* hanya satu yaitu untuk mengamankan dan sebagai media *verifikasi*, sama seperti *password* dan pola, hanya saja media verifikasi yang digunakan pada *Fingerprint* adalah bentuk sidik jari manusia, itulah yang akan digunakan sebagai media untuk ID *Primary Key*.

c. Cara Kerja *Finger Print*

Cara kerja sensor *Fingerprint* adalah dengan merekam data sidik jari untuk pertama kalinya digunakan sebagai acuan. Data sidik jari tersebut akan disimpan dalam *database*. Kemudian data sidik jari hasil scanning ulang tersebut akan dicocokan apakah sama seperti data sidik jari yang sudah pernah disimpan dalam *database*? Jika data tersebut sama, maka akses akan dibuka. Sedangkan jika data tersebut berbeda dengan data yang ada di *database*, maka akses akan tetap tertutup.

5. *Android*

a. Pengertian *Android*

Menurut (Arifianto, 2011), *Android* merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang Menggunakan *Linux*. Menurut

(Hermawan, 2011), *Android* merupakan OS *(Operating System) Mobile* yang tumbuh ditengah *OS* lainnya yang berkembang dewasa ini. Berdasarkan pendapat diatas, maka dapat ditarik kesimpulan bahwa *Android* adalah sistem operasi Menggunakan *Linux* yang sedang berkembang ditengah OS lainnya.

b. Karakteristik *Android*

Android memiliki empat karakteristik, yaitu terbuka, semua aplikasi dibuat sama, memecahkan hambatan pada aplikasi, dan pengembangan aplikasi yang cepat dan mudah.

c. *Android Software Development Kit* (SDK)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java. Android* merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang release oleh *Google.* Saat ini di sediakan *Android* SDK *(Software Development Kit)* sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemograman java (Nazruddin, 2011).

d. *Java SE Development Kit* (JDK)

JDK adalah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *java* ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE *(Java Runtime Envirotment).* JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi Menggunakan

java, namun tidak wajib terinstall di komputer yang akan menjalankan aplikasi.

e. *Android Development Tools* (ADT)

Android Development Tools (ADT) yang lebih dikenal *plugin Eclipse*. *Plugin* ini yang membuat *Eclipse* dapat membuat *project* yang Menggunakan *Android*. ADT adalah *plugins* di *Eclipse* yang harus diinstall sehingga *Android* SDK dapat dihubungkan dengan IDE *Eclipse* yang digunakan sebagai tempat coding aplikasi *Android* nantinya. ADT adalah kepanjangan dari *An*droid Development Tolls yang menghubung antara IDE *Eclipse* dengan *Android* SDK (Nazruddin, 2011). *Android Virtual Device* (AVD) AVD yang merupakan emulator untuk menjalankan program aplikasi *Android* yang dibuat. AVD ini nantinya yang kita jadikan sebagai tempat tes dan menjalankan aplikasi *Android* yang dibuat. AVD berjalan di *virtual Manchine* (Nazruddin, 2011).

6. *Cloudinary*

Cloudinary adalah layanan penyimpanan gambar dan file, di mana pemilik blog dapat menyimpan gambar yang akan digunakan di blog mereka lalu mengintegrasikan keduanya sehingga dapat menghemat ruang penyimpanan hosting secara signifikan.

Kelebihan-kelebihan yang di tawarkan Cloudinary adalah ruang penyimpanan sebesar 500MB, jika butuh space lebih bisa di-upgrade. Maksimal 50 ribu gambar, Bandwidth bulanan 1GB, akses super cepat, Mendukung 'featured image', fitur ini jarang ditemukan di layanan serupa

lainnya, Khusus para blogger dimudahkan dengan sebuah plugin untuk proses integrasi, Terdapat fitur untuk merubah ukuran gambar, judul, *keyword* dan juga target *link*.

C. Tinjauan Perangkat Lunak

1. *JavaScript*

JavaScript adalah bahasa pemrograman tingkat tinggi dan dinamis. *JavaScript* populer di internet dan dapat bekerja di sebagian besar penjelajah *web* populer seperti *Google Chrome, Internet Explorer* (IE), *Mozilla Firefox, Netscape* dan *Opera.* Kode *JavaScript* dapat disisipkan dalam halaman *web* menggunakan *tag SCRIPT. JavaScript* merupakan salah satu teknologi inti *World Wide Web* selain *HTML* dan *CSS. JavaScript* membantu membuat halaman *web* interaktif dan merupakan bagian aplikasi *web* yang esensial.

Awalnya hanya diimplementasi sebagai *client-side* dalam penjelajah *web,* kini *engine JavaScript* disisipkan ke dalam perangkat lunak lain seperti dalam *server-side* dalam server *web* dan basis data, dalam program *non web* seperti perangkat lunak pengolah kata dan pembaca *PDF,* dan sebagai *runtime environment* yang memungkinkan penggunaan *JavaScript* untuk membuat aplikasi *desktop* maupun *mobile.*

Kode *javascript* dapat diselipkan pada dokumen *html "*.html"* atau dapat pula disimpan pada file terpisah dengan ekstensi *"*.js"* lalu dipanggil dari dokumen html dengan tujuan agar kode *javascript* dapat lebih *reusable*.

Kelebihan menggunakan *clientside scripting* adalah dapat digunakan untuk membangun sebuah tampilan yang dinamis dan lebih atraktif. Namun, kekurangannya adalah karena merupakan *client-side scripting*, maka kode program dapat di lihat oleh browser pengguna, hal ini dapat menimbulkan celah keamanan jika salah dalam penggunaannya. Berikut ini adalah beberapa contoh *sintaks* penulisan dalam bahasa pemrograman *JavaScript* :

a. Menentukan variable dan lainnya

```
Var x = 1;
Var y = 2;
Var z = 3;
```

Gambar 2.1 Deklarasi variabel dan nilai dalam *JavaScript*

b. Menghasilkan teks *Output* ke dalam *console*

```
console.log("hello world");
```

Gambar 2.2 Sintaks untuk menampilkan teks *Output* ke dalam *console*

c. Contoh sintaks *rekursif*

```
function factorial(n) {
    if (n === 0) {
        return 1;
    }
    return n * factorial(n - 1);
}
```

Gambar 2.3 Sintaks *rekursif* dalam *JavaScript*

d. Contoh sintaks fungsi *anonymous*

```
var displayClosure = function() {
    var count = 0;
    return function () {
        return ++count;
    };
}
var inc = displayClosure();
inc(); // returns 1
inc(); // returns 2
inc(); // returns 3
```

Gambar 2.4 Sintaks fungsi *anonymous* dalam *JavaScript*

2. *Tensorflow.js*

TensorFlow™ adalah koleksi *software open source* untuk komputasi numerik yang menggunakan grafik aliran data. *Node* pada grafik menunjukkan operasi matematika, sedangkan tepi-tepi grafik menunjukkan susunan data multidimensi *(Tensor)* yang dikomunikasikan antartepi grafik. Arsitekturnya yang fleksibel dapat dimanfaatkan untuk menerapkan komputasi ke satu atau beberapa CPU atau GPU pada *desktop*, server, atau perangkat seluler dengan cukup menggunakan satu *API*. TensorFlow awalnya dikembangkan oleh para peneliti dan *engineer* yang bekerja di Google Brain Team dalam organisasi riset *Machine Intelligence Google*, untuk keperluan menjalankan *machine learning* dan riset jaringan *deep neural network*. Namun, sistem ini juga cukup untuk diterapkan pada aneka domain lainnya. *TensorFlow* ini digunakan sebagai *backend* dari Keras, yang artinya adalah *TensorFlow* sebagai bagian belakang layar dari sebuah aplikasi atau *websit*e dengan menggunakan bahasa pemrograman *Python* dan dikombinasikan dengan *library Keras* dari *deep learning*. (Deka M Wildan, 2014)

3. *Vue.js*

Vue js ialah *progressive framework* yang digunakan untuk membangun tampilan halaman *website*. Berbeda dengan *framework* lainya yang menggunakan prinsip MVC *(Model View Controller) Vue js* hanya difokuskan untuk membangun tampilan/ hanya bekerja pada *view layer*. Dalam hal lainnya *Vue js* juga dapat membantu programmer untuk membuat *web* dengan aplikasi *single page* yang canggih dan dapat di kombinasikan dengan *library browser* lainnya.

Selain itu *Vue js* menggunakan template *sintaks* berbasis HTML yang memungkinkan pengguna untuk mendeklarasikan data/ *state* kedalam DOM. Semua *template Vue js* adalah HTML yang valid yang dapat diuraikan oleh *browser* sesuai spesifikasi dan parser HTML. (Madcoms Madiun, 2012)

4. *ExpressJS*

Express.js adalah sebuah *framework* aplikasi berbasis *web* yang menggunakan *core* pemrograman *Node.js* dengan komponen modul *Http* dan *Connect* (Express.js, 2016). *Framework* ini dibuat untuk berjalan dalam mesin V8 *Chrome*, membuatnya berjalan beriringan dengan *Node* sehingga dapat melakukan kerja dengan sangat cepat.

Framework Express.js menggunakan proses paradigma kerja yang disebut dengan *event-driven programming*. Paradigma ini berpengaruh pada *flow* sistem yang telah dibuat dengan membuat semua fungsi berjalan berdasarkan *event*. Semua *event* akan terus diawasi oleh *code* yang disebut dengan *event* handler yang bekerja dengan cara mendeteksi *event* aktif kemudian memanggil *handler*.

Kebanyakan *handler* memiliki tipe sebagai *function* atau *method*. Dalam handler ini terjadi proses fungsi yang tengah dipicu *user* (Vivah, 2017). Seluruh proses tersebut tidak terjadi secara otomatis. Multi proses yang terjadi di server ini dihandle oleh *Node* dengan menggunakan *Node Event Loop*.

5. *Node.js*

Node.js merupakan salah satu peranti pengembang yang bisa digunakan untuk membuat aplikasi berbasis *Cloud. Node.js* dikembangkan dari engine *JavaScript* yang dibuat oleh *google* untuk browser *Chrome/ Chromium* (V8) ditambah dengan *libUV* serta beberapa pustaka internal lainnya. Dengan menggunakan *Node.js*, semua pengembangan akan dilakukan menggunakan *JavaScript*, baik pada sisi klien maupun server. *Node.js* dibuat pertama kali oleh Ryan Dahl *(twitter.com/ryah)* dan sampai saat ini dikembangkan oleh komunitas sebagai perangkat lunak bebas dengan pendanaan utama dari *Joyent*, perusahaan tempat Ryan Dahl bekerja. (Purnomosidi, 2013 : 15)

6. *MongoDB*

MongoDB adalah database dokumen dengan skalabilitas dan fleksibilitas yang anda inginkan dengan *query* dan pengindeksan yang anda butuhkan. *MongoDB* merupakan salah satu database *NoSQL*, yang merupakan sebuah konsep penyimpanan data tanpa perlu adanya table *schema* dan tidak ada bahasa *sql* yang terlibat dalam pemakaian database.

Unit terkecil pada *MongoDB* adalah *document*, lalu kumpulan dari *document* adalah *collection*. Dokumen dalam *MongoDB* memiliki atribut yang

berbeda-beda dengan dokumen yang lainnya walaupun dalam satu *collection*. Metode penyimpanan yang digunakan oleh *MongoDB* adalah *object-oriented* dan dukungan tipe *index* yang berbeda untuk *lookups* terhadap data tertentu dan memiliki kemampuan *clustering* secara *default*.

7. *Visual Studio Code*

Visual Studio Code adalah editor *source code* yang dikembangkan oleh *Microsoft* untuk *Windows, Linux* dan *MacOS.* Ini termasuk dukungan untuk *debugging, GIT Control* yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring.* Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi. *Visual Studio Code* gratis dan *open-source*, meskipun unduhan resmi berada di bawah *lisensi proprietary.*

Kode *Visual Studio* didasarkan pada *Elektron*, kerangka kerja yang digunakan untuk menyebarkan aplikasi *Node.js* untuk *desktop* yang berjalan pada Blinklayout. Meskipun menggunakan kerangka *Elektron, Visual Studio Code* tidak menggunakan *Atom* dan menggunakan komponen editor yang sama (diberi kode nama "Monaco") yang digunakan dalam *Visual Studio Team Services* yang sebelumnya disebut *Visual Studio Online* (Lardinois, 2015).

8. *Google Maps API*

Google Maps API atau *Application Programming Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya *API* ini,

maka memudahkan programmer untuk "membongkar" suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. *API* dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan sistem *function*. Proses ini dikelola melalui *operating system*. Keunggulan dari *API* ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh *Google Map* yang terdiri dari *HTML, Javascript* dan *AJAX* serta *XML* memungkinkan untuk menampilkan peta *Google Map* di *website* lain.

Google juga menyediakan layanan *Google Maps API* yang memungkinkan para pengembang untuk mengintegrasikan *Google Maps* ke dalam *website* masing-masing dengan menambahkan data *point* sendiri. Dengan menggunakan *Google Map API*, *Google Maps* dapat ditampilkan pada *website* eksternal. Agar aplikasi *Google Maps* dapat muncul di *website* tertentu, diperlukan adanya *API key*. *API key* merupakan kode unik yang digenerasikan oleh *google* untuk suatu *website* tertentu, agar server *Google Maps* dapat mengenali. Pada *Google Maps API* terdapat 4 jenis pilihan model peta yang disediakan oleh *Google*, diantaranya adalah :

a. Roadmap, ini yang sering dipilih untuk menampilkan peta biasa 2 dimensi, karena relatif lebih ringan.
b. Satelite, untuk menampilkan foto satelit.

c. Terrain, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, misalnya akan menunjukkan gunung dan sungai dengan ketinggian masing-masing.
d. Hybrid, menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan dan nama kota).

D. Perancangan Basis Data

Menurut (Jogiyanto Hartono, 2005:700). Basis data (*Database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasi. Sedangkan menurut penadapat saya Basis Data adalah data data yang tersimpan dalan komputer dan masih harus di proses agar saling berhubungan satu sama lain. Database merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakai.

Sistem basis data ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

1. *Entity Relationship Diagram* (ERD)

Menurut Andi, ERD berguna untuk mengorganisasi data-data (*hanya data*) dalam hubungannya dengan data-data yang lain. Sedangkan menurut saya ERD berfungsi untuk mengelompokan data-data agar tejadi keselarasan hubungan antar

data lainnya. ERD mempunyai koleksi objek-objek dasar yang dinamakan entitas serta hubungan (*relationship*) antara entitas –entitas itu.

Entitas adalah sesuatu atau objek pada dunia nyata yang dapat dibedakan satu terhadap yang lainnya, yang bermanfaat bagi aplikasi yang sedang kita kembangkan. Sedangkan *Relationship* adalah hubungan antara beberapa entitas.

2. Normalisasi.

Menurut Andi, Normalisasi sebagai proses untuk mengubah suatu relasi yang memiliki masalah tertentu ke dalam dua buah relasi atau lebih yang tak memiliki masalah tersebut. Sedangkan Normalisasi menurut pendapat saya adalah proses pengolahan data yang masih salah dan memiliki data ganda untuk kemudian di perbaiki sehingga menghasilkan relasi-relasi yang lebih baik dari sebelumnya. Normalisasi dapat dipahami sebagai tahapan yang masing-masing berhubungan dengan bentuk normal.

Bentuk normal adalah keadaan relasi yang dihasilkan dengan menerapkan aturan sederhana yang bekaitan dengan konsep kebergantungan fungsional pada relasi yang besangkutan sebagai berikut.

a. Bentuk normal pertama.

   Bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak (*Multivalues attribute*) telah dihilangkan sehingga kita akan menjumpai nilai tunggal (*mungkin saja nilai null*) pada perpotongan setiap baris dan kolom.

b. Bentuk normal kedua.

Suatu relasi adalah dalam bentuk normal kedua jika dia berada dalam bentuk normal pertama dan setiap atribut bukan kunci bergantung penuh pada kunci *primer.*

c. Bentuk normal ketiga.

Relasi adalah dalam bentuk normal ke tiga adalah jika berada dalam bentuk normal kedua dan tidak dijumpai kebergantungan *transitif.* Kebergantungan *transitif* dalam suatu relasi adalah kebergantungan fungsional antara 2 atau lebih atribut bukan kunci.

3. Tabel Relasi.

Tabel relasi adalah tabel hubungan antara suatu himpunan entitas dengan entitas yang lainnya.

4. Kunci.

Kunci (*key*) merupakan suatu atribut yang unik yang dapat digunakan untuk membedakan suatu entitas dengan entitas yang lainnya dalam suatu himpunan entitas. *Primary Key* adalah atribut *(field)* yang dipilih untuk menentukan struktur storage pada organisasi file multi *key*, adapun *key* lainnya disebut dengan *secondary key*. Pengertian lainya *Primary Key* adalah suatu nilai dalam basis data yang digunakan untuk mengidentifikasi suatu baris dalam tabel. Nilai dari *primary key* adalah unik. pengertian lain. *Primary key* : Salah satu atribut dari candidat *key* dapat dipilih menjadi *primary key* dengan 3 kriteria.

E. Tahap-tahap Perancangan Sistem

1. *Use Case*

Use case diagram adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* bekerja mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

2. Kamus Data

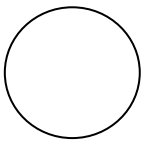
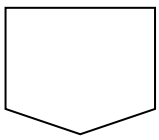
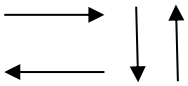
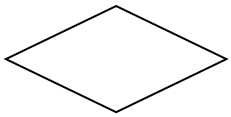
Kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan informasi yang mengalir pada sistem dengan lengkap dari suatu sistem informasi. Kamus data di buat berdasarkan arus data yang mengalir pada *Data Flow Diagram* (DFD).

Kamus data merupakan tempat penyimpanan dari aliran-aliran data, *file-file* dan proses-proses dalam sebuah sistem dibangun. Bagian ini menjelaskan secara detail proses-proses yang terjadi disetiap proses, *file* dan struktur data mengenai model sistem yang digambarkan. Kamus data digunakan untuk menjelaskan semua data yang mengalir atau digunakan dalam sistem, yaitu mengenai arus data yang masuk ke dalam sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

3. Bagan Alir Program

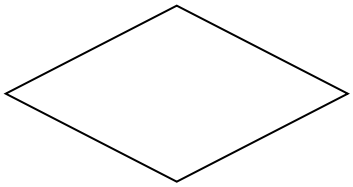
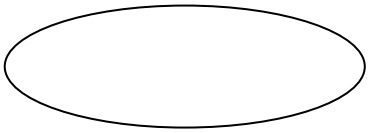
Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derifikasi bagan alir system. Bagan alir program dibuat dengan menggunakan simbol-simbol sebagai berikut :

Tabel 2.1 Simbol *Flowchart*

SIMBOL	KETERANGAN
	Simbol *Input/Output*digunakan untuk mewakili data *Input/output.*
	Simbol proses, digunakan untuk mewakili suatu proses.
	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
	Simbol titik terminal, digunakan untuk awal dan akhir suatu proses.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus dihalaman yang masih sama.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus dihalaman yang berbeda.
	Simbol garis alir, menunjukkan arus dari proses.
	Simbol keputusan, digunakan unuk suatu penyelesaian kondisi didalam program.

4. *Entity Relationship Diagram*

Tabel 2.2 *Simbol Entity Relationship Diagram*

NOTASI	KETERANGAN
	Entitas, yaitu kumpulan dari objek yang di identifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak dan banyak ke banyak.
	Atribut, yaitu karakteristik dari *entity* atau relasi yang merupakan penjalasan detail tentang entitas.
	Garis, hubungan antara *entity* dengan atributnya dan himpunan entitas dengan himpunan relasi.
	*Input/ Output* data, yaitu proses *Input/ Output* data, parameter, informasi.

Entity Relationship Diagram (ERD) adalah adalah suatu pemodelan konseptual yang didesain secara khusus untuk mengidentifikasikan entitas yang menjelaskan data dan hubungan antar data. ERD merupakan suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu :

a. Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

b. Atribut Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendes-kripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol *elips.*

c. Hubungan / Relasi ialah Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Relasi dapat digambarkan sebagai berikut : Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

1) Satu ke satu (*One to one*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2) Satu ke banyak (*One to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3) Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

5. UML *(Unified Modelling Language)*

UML *(Unified Modelling Language)* menurut Munawar dalam buku "Pemodelan Visual dengan UML" (2005 : 17) menyatakan UML adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi *(sharing)* dan mengkomunikasikan rancangan mereka dengan yang lain.

Sedangkan menurut menurut Adi nugroho (2010:6), "*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek". Pemodelan *(modeling)* sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Jurnal CCIT (2008:70) *Unified Modelling Language* (UML) didefinisikan oleh *Good Corporate* sebagai berikut :

Unified Modelling Language (UML) adalah "suatu alat bantu yang sangat handal dalam bidang pengembangan yang berorientasi objek. Hal ini disebabkan karena uml menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem untuk membuat cetak cetak biru atas visi mereka dalam bentuk baku". *Metode Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu :

a. Sesuatu *(things)*

Ada empat *things* dalam *Unified Modelling Language* (UML) :

1) *Structural things*, bagian yang relative statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
2) *Behavorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang ruang dan waktu.
3) *Grouping things*, bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta *subsitem-subsistem*.
4) *An notational things*, merupakan bagian yang meperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML.

b. Relasi *(relationship)*

Untuk upaya tersebut UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya *statis* atau *dinamis*. Ke 9 jenis diagram dalam UML itu adalah :

1) *Diagram* kelas Bersifat statis Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula *diagram* kelas memuat kelas-kelas aktif.

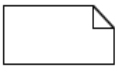
2) *Diagram* Objek Besifat statis. *Diagram* ini memperlihatkan objek-objek serta relasi-relasi antar objek. Diagram objek memperlihatkan instalasi statis dari segala sesuatu yang dijumpai pada diagram kelas.
3) *Use-Case Diagram* Bersifat statis. *Diagram* ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.
4) *Sequence Diagram* Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5) *Collaboration Diagram* Bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6) *Statechart Diagram* Bersifat dinamis. Diagram state ini memperlihatkan *state-state* pada sistem memuat *state, transisi, event,* serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7) *Activity Diagram* Bersifat dinamis. *Diagram* aktivitas ini adalah tipe khusus dari *diagram state* yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama

penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8) *Component Diagram* Bersifat statis. Dengan kemponen ini memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. *Diagram* ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
9) *Deployment Diagram* Bersifat statis. *Diagram* ini memperlihatkan konfigurasi saat aplikasi dijalankan. *Diagram* ini membuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. *Diagram* ini sangat berguna pada banyak mesin (Sholiq, 2006).

Tabel 2.3 *Table Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek.
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.

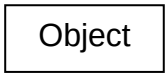
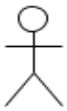
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan suatu paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan dalam sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
NO	**GAMBAR**	**NAMA**	**KETERANGAN**
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

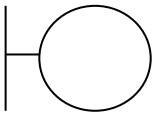
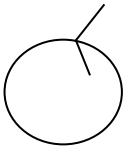
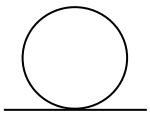
Tabel 2.4 *Symbol Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan cara bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain

2		*Action*	State dari suatu sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap-tahap tertentu dan berubah menjadi beberapa aliran

Tabel 2.5 *Symbol Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Objekct* *(Partisipan)*	*Object* atau biasa juga disebut partisipan merupakan *instance* dari sebuah *class* dan dituliskan tersusun secara *horizontal*.
2		*Actor*	*Actor* juga dapat berkomunikasi dengan *object*, maka *actor* juga dapat diurutkan sebagai kolom.
3		*Life line*	*Life line* mengidentifikasikan keberadaan sebuah *object* dalam basis waktu. Notasi untuk *lifeline* adalah garis putus-putus *vertical* yang di tarik oleh sebuah *object.*
4		*Collaborations*	*Activication* dinotasikan sebagai kotak segi empat yang digambar pada sebuah *lifeline action* mengidentifikasikan sebuah *object* yang akan melakukan sebuah aksi.

5		*Boundary*	*Boundary* terletak diantara *system* dengan dunia di sekelilingnya. Semua form, laporan-laporan, antara muka ke perangkat keras seperti *printer* atau *scanner* dan antara muka ke *system* lainnya adalah termasuk dalam kategori.
6		*Control*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaharui elemen yang bergantun padanya elemen yang tidak mandiri.
7		*Entity*	Entity digunakan menengani informasi yang mungkin akan disimpan secara permanen. Entity biasa juga merupakan subuah *table*.

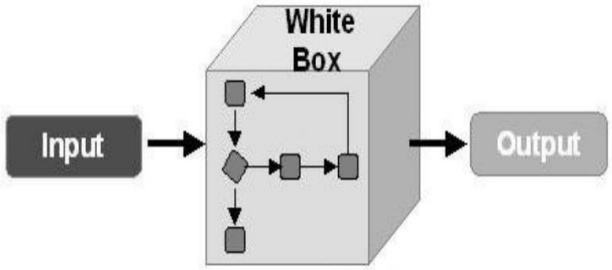
F. Metode Pengujian Perangkat Lunak

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang *tipical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu :

1. *White Box Testing*

Uji coba *White Box testing* merupakan metode perancangan *testcase* yang mengunakan struktural untuk mendapatkan *testcase*, *test* ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di *test* dengan kondisi dan perulangan secara fisik.



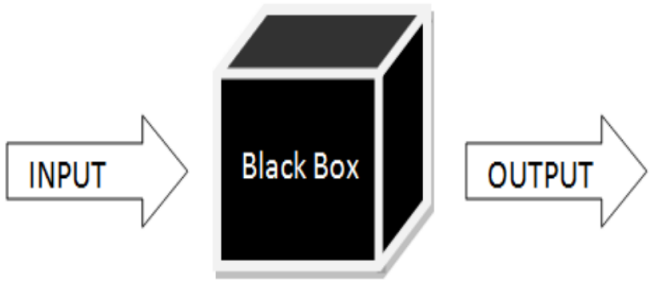
Gambar 2.5 Pengujian *White Box*

Contoh pengujian *White Box* testing ini merupakan peringatan ketika *user* meng*Input*kan *password user* yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada *user* mengenai kesalahan yang dilakukan.

2. *Black Box*

Black Box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama

seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya *(interfacenya)*, fungsionalitas tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui *Input* dan *output*).



Gambar 2.6 Pengujian *Black Box*

Berikut dari bagian-bagian dalam teknik yang digunakan dalam *Black Box Testing* antara lain :

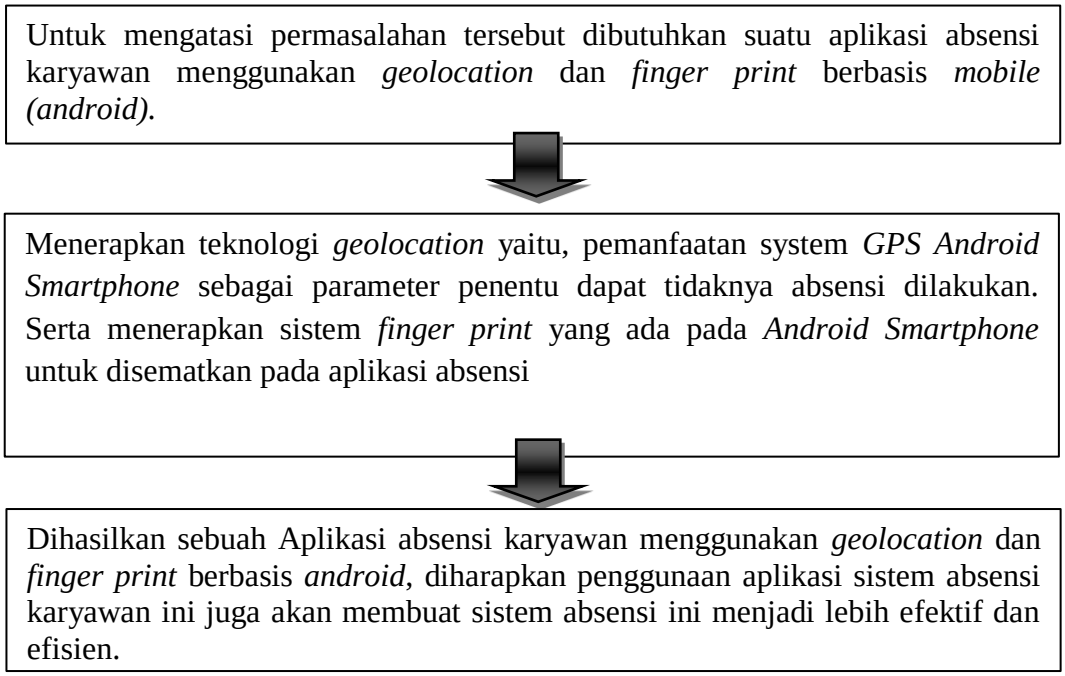
a. Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak.
b. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran (*output*) yang dihasilkan.
c. Kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya.

G. Kerangka Fikir

Untuk lebih memperjelas kerangka berpikir yang akan ditulis dalam penelitian akhir ini. Maka berikut ini digambarkan kerangka pikir tersebut pada gambar 2.7.

Bosowa Parepare telah menggunakan alat *Finger Print* untuk proses pengabsensiannya. Absensi dilakukan dengan menempelkan sidik jari pada alat *Scanner* yang telah disediakan. Namun, alat tersebut bisa rusak atau fungsinya melemah jika tidak berada dalam perawatan yang baik dan tepat. Serta

penggunaan alat absensi secara umum, dapat memperbesar resiko penularan *Covid-19.*



Gambar 2.7 Kerangka Fikir

BAB III
METODOLOGI PENELITIAN

A. Tempat dan Waktu Penelitian

1. Penelitian yang dilakukan oleh penulis bertempat di Bosowa Parepare. Jl.Melati No.2, Mallusetasi, Kec.Ujung, Kota Parepare, Sulawesi Selatan 91111.
2. Pelaksanaan penelitian ini dilakukan selama (±) 2 bulan.

B. Metode Penelitian

Agar dalam perancangan sistem lebih terarah dan terfokus di dalam mendapatkan data dan informasi yang akurat yang berhubungan dengan pokok pembahasan maka pengumpulan data dilakukan dengan tiga cara adalah metode *literature riview*, wawancara, observasi dan kajian pustaka.

1. Penelitian Kepustakaan *(Literature riview)*

Penulis melakukan studi kepustakaan dengan mencari informasi yang akurat sesuai dengan judul tugas akhir menggunakan beberapa teori, literatur, jurnal, buku dan kepustakaan. Penulis berharap agar program aplikasi yang dibuat nantinya tidak menyimpang dari defenisi-defenisi yang ada sehingga benar-benar memiliki acuan yang sesuai dan lengkap. Studi literatur merupakan salah satu metode pengumpulan data untuk mendapatkan data-data yang sifatnya teoritis yaitu dengan cara membaca literatur yang relevan dengan pengamatan yang penulis lakukan.

2. Wawancara *(Interview)*

Wawancara merupakan teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab langsung antara pengumpul data terhadap narasumber/ sumber data. Wawancara dilakukan untuk mendapatkan informasi yang akurat sesuai dengan data yang dibutuhkan dimana pengumpulan data dilakukan penelitian wawancara langsung dengan pihak Bosowa Parepare, yakni dengan cara mengadakan tanya jawab langsung kepada general manager, supervisor dan karyawan atau staf Bosowa Parepare, yang dapat memberikan penjelasan langsung ataupun data sebagai pelengkap penelitian ini.

3. Pengamatan *(Observation)*

Studi lapangan *(observasi)* merupakan teknik pengumpulan data dengan langsung terjun ke lapangan untuk mengamati permasalahan yang terjadi secara langsung di tempat kejadian secara sistematik kejadian-kejadian, perilaku, objek-objek yang dilihat dan hal-hal lain yang diperlukan dalam mendukung penelitian yang sedang berlangsung. Dalam penelitian ini, peneliti melakukan pengamatan langsung ke lokasi-lokasi yang dianggap perlu.

C. Metode Pengumpulan Data

Pada tahap ini peneliti melakukan wawancara dengan pihak yang terkait dengan penelitian, serta melakukan pencatatan dan pengamatan *(observasion)* langsung di tempat penelitian. Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data analisis, perancangan, pengujian dan implementasi. Adapun uraian dari tahapan tersebut adalah sebagai berikut :

1. Persiapan Penelitian

   Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Studi *Literature*

   Pada tahapan ini peneliti melakukan apa yang disebut dengan kajian pustaka, yaitu mempelajari buku-buku referensi dan hasil penelitian sejenis sebelumnya yang pernah dilakukan oleh orang lain.

3. Pengumpulan Data

   Pada tahap ini peneliti melakukan wawancara dengan pihak yang terkait dengan penelitian, serta melakukan pencatatan dan pengamatan langsung di tempat penelitian.

4. Analisis

   Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

5. Perancangan

   Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

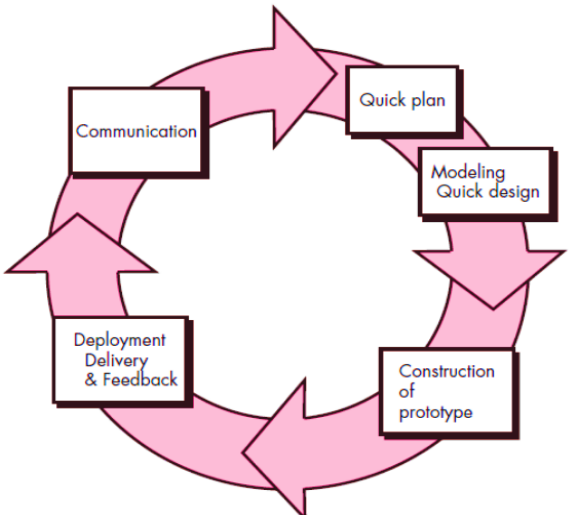
6. Pengujian

   Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

D. Metode Pengembangan Sistem

Metode Pengembangan sistem merupakan suatu kerangka kerja mengenai bagaimana proses pelaksanaan Aplikasi Absensi Karyawan Menggunakan *Geolocation* dan *Finger Print* berbasis *Android*, adapun metode pengembangan yang digunakan yaitu *Prototype* hal ini dikarenakan pada Bosowa Parepare belum sama sekali memiliki aplikasi yang berkaitan dengan absensi karyawan sehingga sangat cocok menggunakan metode *prototype*.

Menurut Pressman (2012), dalam melakukan perancangan sistem yang akan dikembangkan dapat mengunakan metode *prototype*. Metode ini cocok digunakan untuk mengembangkan perangkat-perangkat lunak yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, dalam hal ini pengguna dari perangkat yang dikembangkan adalah Aplikasi Absensi Karyawan Menggunakan *Geolocation* dan *Finger Print* berbasis *Android*. Tahap berikutnya membuat rancangan kilat yang selanjutnya akan di evaluasi kembali sebelum diproduksi secara benar. Metode pengembangan sistem yang digunakan adalah model *prototype* dengan 5 tahapan yaitu *communication, quick plan, modeling quick design, construction of prototipe, deployment, delivery & feedback.*



Gambar 3.1 Model *Prototype*

Berikut adalah tahapan-tahapan dalam metode *prototype* :

1. *Communication*

Pada tahap ini, antara pelanggan dengan tim pemgembang perangkat lunak berkomunikasi mengenai spesifikasi kebutuhan yang diinginkan. Tim pengembang melakukan pengumpulan data yang akan dgunakan dalam pembuatan sistem yang dibutuhkan.

2. *Quick Plan*

Tim pengembang memberikan gambaran besar mengenai sistem yang akan dibuat kepada *general manager* dan spv. Dalam hal ini sistem yang akan dibuat adalah Aplikasi Absensi Karyawan Menggunakan *Geolocation* berbasis *Android*. kemudian pengembang merancang penggunaan spesifikasi *hardware* dan *software* yang akan digunakan dalam pembuatan sistem.

3. *Modeling Quick Design*

Pada tahap ini, pengembang membuat perancangan sistem menggunakan *use case diagram, activity diagram, sequence diagram* dan *user interface*.

4. *Construction of Prototipe*

Setelah tahap pemodelan, maka pengembang mulai melakukan pengkodean program. Bahasa pemograman yang digunakan adalah *PHP* dan *JavaScript* menggunakan database *MySQL* dengan *tools PhpMyAdmin*.

5. *Deployment, Delivery & Feedback*

Tahap pengkodean program dibarengi oleh tahapan implementasi dan pengujian sistem. Sistem diuji dengan metode *White Box* dan *Black Box* testing dengan pengujian fungsional. Setelah sistem lulus uji coba, maka sistem yang baru diberikan kepada pengguna dan pengembang mengajari pengguna dalam penggunaan sistem yang baru. Jika ada kekurangan atau penambahan kebutuhan sistem, maka pihak terkait akan mengkomunikasikan kembali dengan tim pengembang.

E. Sumber Data Penelitian

Adapun jenis dan sumber data yang dibutuhkan dalam penelitian ini adalah sebagai berikut :

1. Data Primer yaitu data yang diperoleh melalui wawancara yang dilakukan langsung dengan responden yang dapat mewakili beberapa Sumber data. Dalam penelitian ini merupakan subjek penelitian utama yakni Bosowa Parepare.
2. Data sekunder yaitu merupakan data yang diperoleh melalui studi kepustakaan atau dari berbagai literatur dengan menelah buku-buku dan tulisan-tulisan atau

internet, jurnal hukum, serta peraturan perundang-undangan yang relavan dengan permasalahan yang diteliti.

F. Alat dan Bahan

Tahap desain dilakukan untuk menentukan ruang lingkup pembuatan aplikasi yang ditangani, yang selanjutnya dijabarkan dalam bentuk diagram alir. Adapun *Hardware* dan *Software* yang digunakan adalah :

1. Perangkat keras *(hardware)*

Perangkat keras yang digunakan untuk membuat aplikasi menggunakan PC *(Personal Komputer)* dengan spesifikasi yang dapat dilihat pada tabel 3.1 berikut :

Table 3.1 Spesifikasi Perangkat Keras

Jenis	Spesifikasi
Laptop	*Acer*
Processor	*Processor Intel Core i7-7500U Processor (up to 4.0GHz)*
Ram	8 GB DDR 4 2400MHz
Hardisk	1 TB 5400 RPM
Smartphone	*Android* versi 8.0 *(Oreo)*

2. Perangkat Lunak *(software)*

Perangkat lunak yang digunakan untuk membangun aplikasi ini dapat dilihat pada tabel 3.2 berikut :

Table 3.2 Spesifikasi Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	*Windows* 10 *ultimate* 64 bit
Bahasa Pemrograman	*JavaScript, Flutter*
Database	*MonggoDB*
Tools	*Visual Studio Code*
Bahasa Pemrograman	*Windows* 10 *ultimate* 64 bit

BAB IV

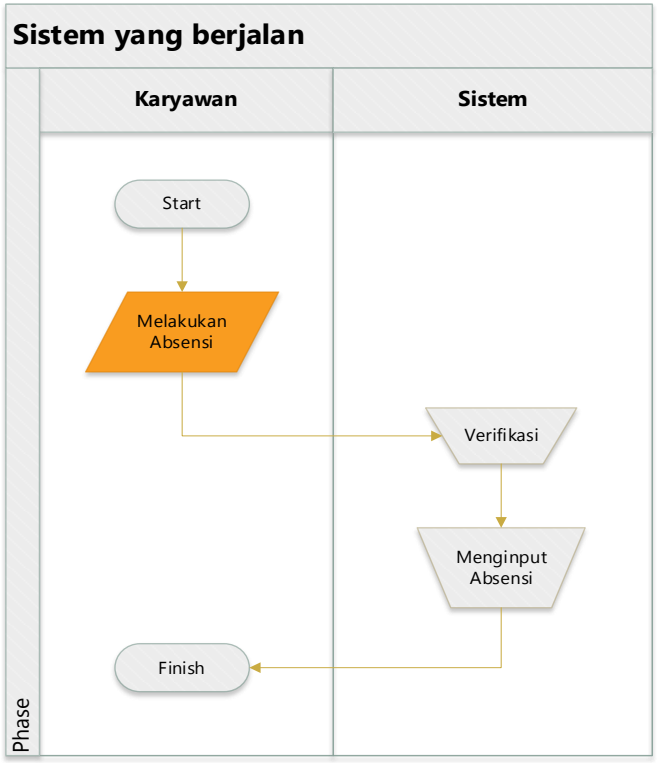
HASIL DAN PEMBAHASAN

A. Hasil

Pada bagian ini akan dipaparkan Analisis Sistem yang Berjalan saat ini pada PT.Bosowa Berlian Motor cabang Parepare serta Sistem yang diusulkan

1. Analisa Sistem yang berjalan.

Sistem yang sedang berjalan saat ini di PT. Bosowa Berlian Motor Parepare adalah absensi menggunakan alat *finger print*. Dalam kesehariannya, absensi karyawan di perusahaan ini adalah dengan cara mendeteksi sidik jari pada sebuah alat absensi yang disediakan oleh perusahaan untuk mencatat kehadiran karyawan. Cara kerja alat absensi ini mirip dengan mesin fotocopy, dimana jari diletakkan di atas scanner yang biasanya berbahan kaca. Kemudian, dari bawah scanner muncul pemancar cahaya yang menerangi jari yang diletakkan tadi. Alat akan membaca dan memverifikasi sidik jari untuk kemudian diinput dalam pencatatan kehadiran. Proses absensi karyawan tersebut dilakukan dalam beberapa tahap seperti yang dapat dilihat pada *flowmap diagram* berikut :



Gambar 4.1 *Flowmap* Prosedur Absensi Manual

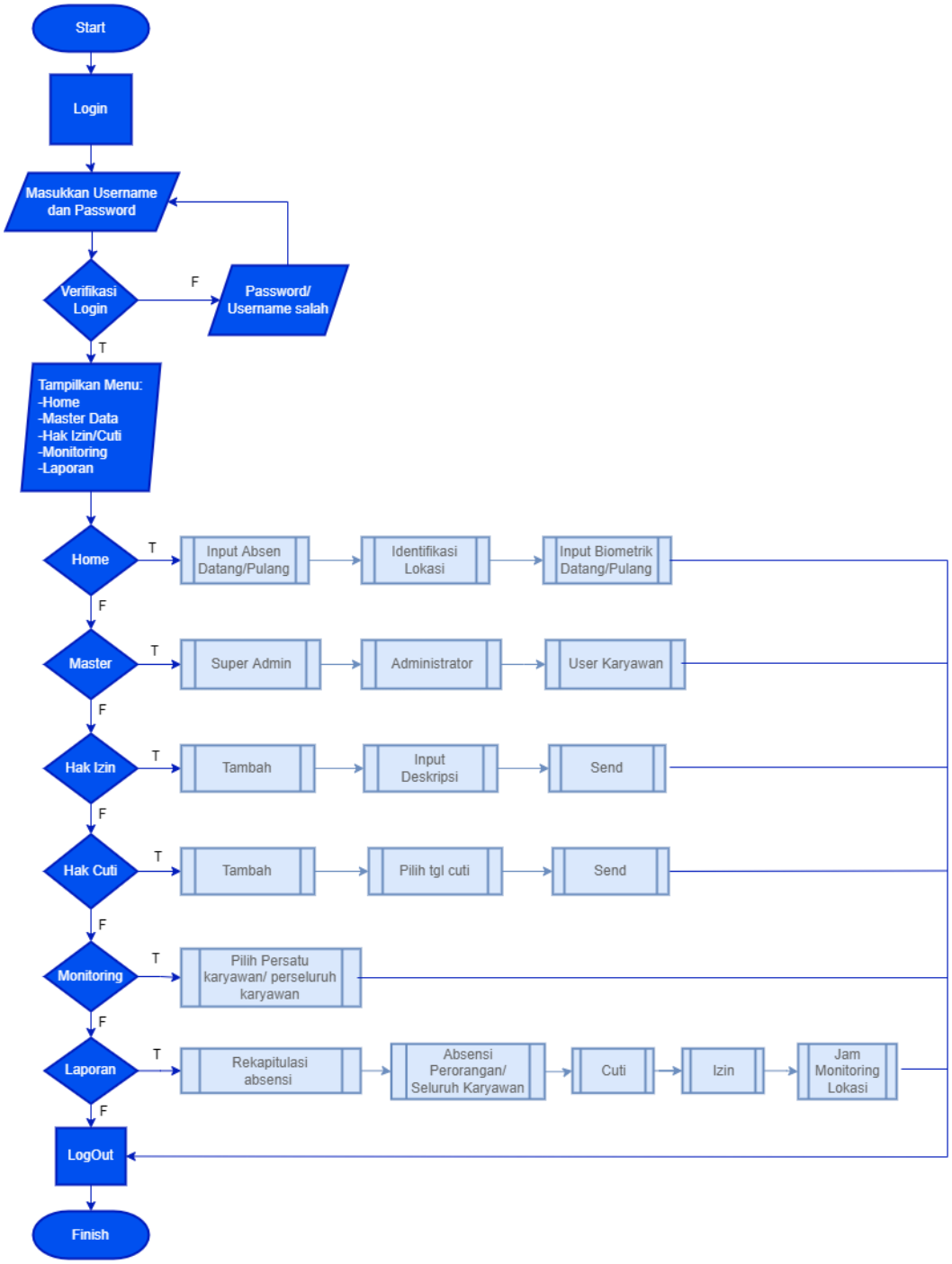
Flowmap merupakan bagan yang menunjukan arus pekerjaan dari sistem. bagan flowmap ini berfungsi untuk menjelaskan tentang urutan-urutan dari prosedur yang ada di dalam *system* yang menggambarkan aliran data atau dokumen dari satu entitas ke entitas lainnya. Untuk menjalankan prosedur sistem, digunakan *flowmap* yang terbentuk dari analisis prosedur. Berdasarkan Gambar 4.1 *Flowmap Diagram* yang berjalan saat ini :

a. Terdapat 1 sistem yang mencakup seluruh kegiatan.
b. Terdapat 2 actor yang melakukan kegiatan diantaranya : Karyawan dan Sistem.
c. 5 *Flowmap Diagram* yang dilakukan oleh *actor* tersebut

2. Analisa Sistem yang diusulkan.

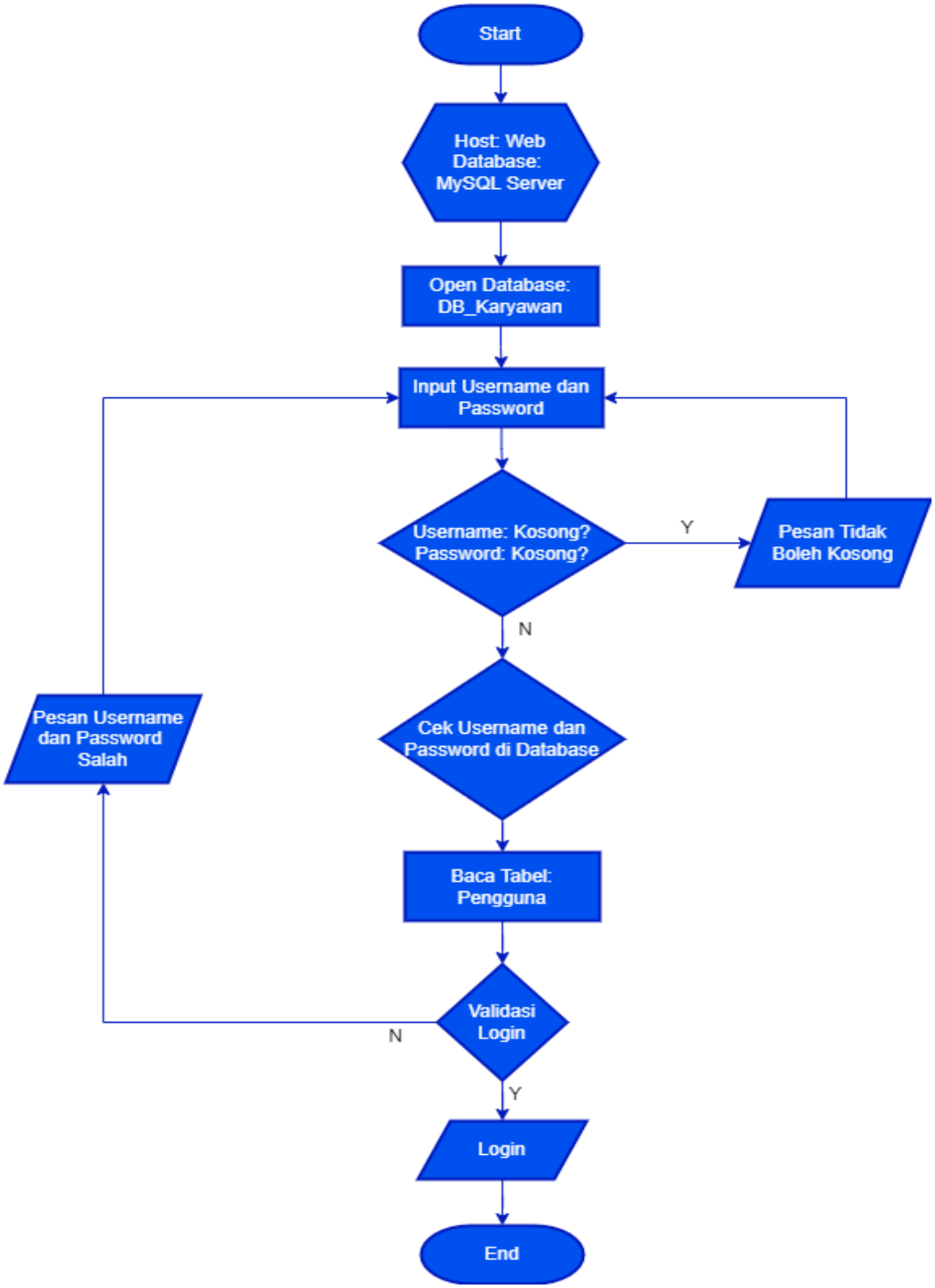
Sistem yang diusulkan menggunakan *Flowchart. Flowchart* merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam alur penelitian. Gambaran dari *Use Case Diagram* pada aplikasi *presensi* karyawan menggunakan *geolocation* dan *finger print* di PT. Bosowa Berlian Motor Parepare yang akan dibangun dapat dilihat pada gambar 4.2 adalah sebagai berikut.

a. *Flowchart Sistem*



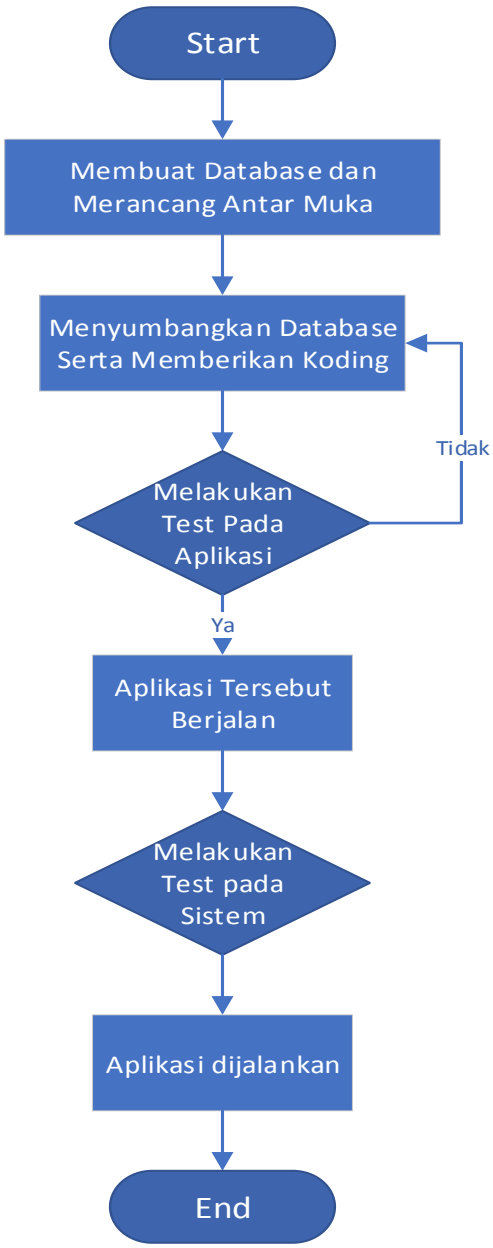
Gambar 4.2 Sistem yang Diusulkan

b. *Flowchart Login*



Gambar 4.3 *Flowchart Login*

c. Flowchart Pembuatan Aplikasi



Gambar 4.4 *Flowchart* Pembuatan Aplikasi

B. Pembahasan

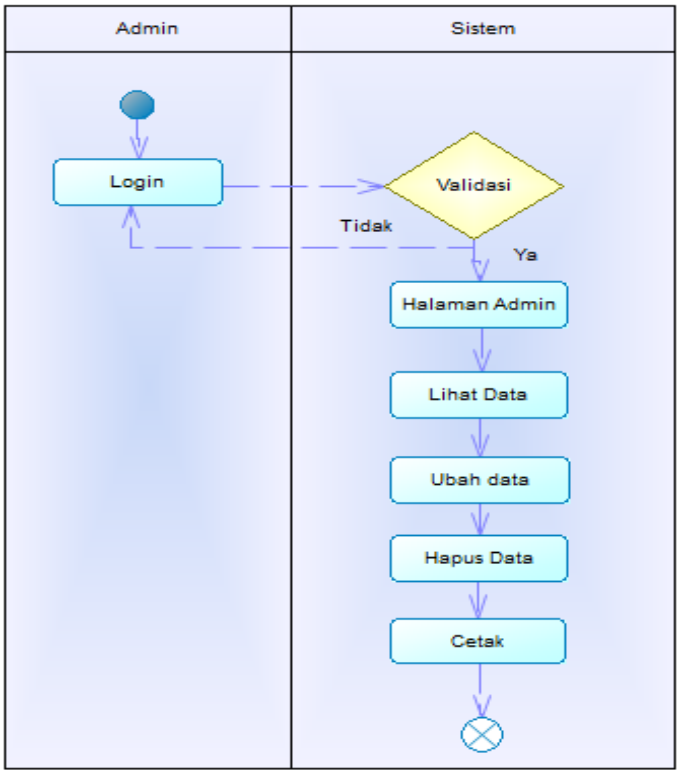
1. Analisis Aliran Data UML

Analisis aliran data yang diusulkan ini bertujuan mengetahui proses informasi, dalam analisis sistem penulis menggunakan pengembangan orientasi objek sehingga menggunakan *Activity Diagram,* dan *Sequence diagram*.

1. *Activity Diagram*

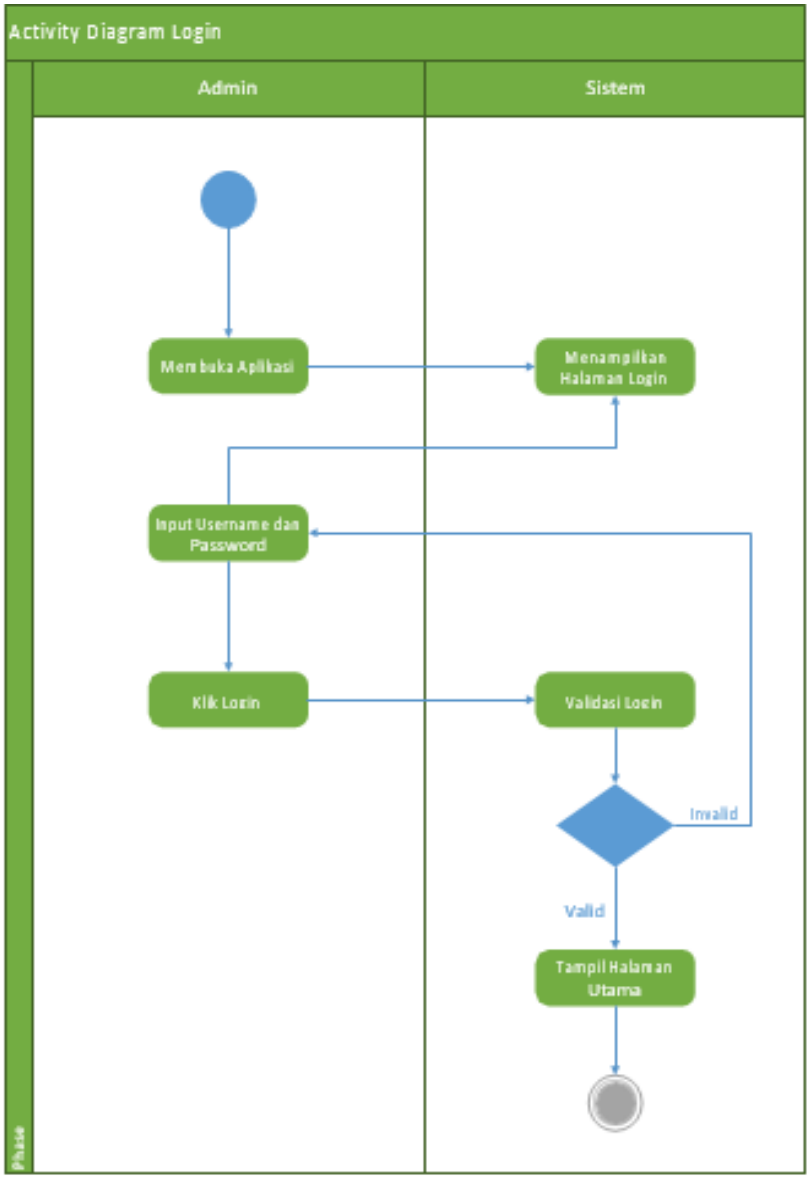
Activity diagram adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak *Activity diagram* ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sebuah *system*.

a. *Activity Diagram Admin*



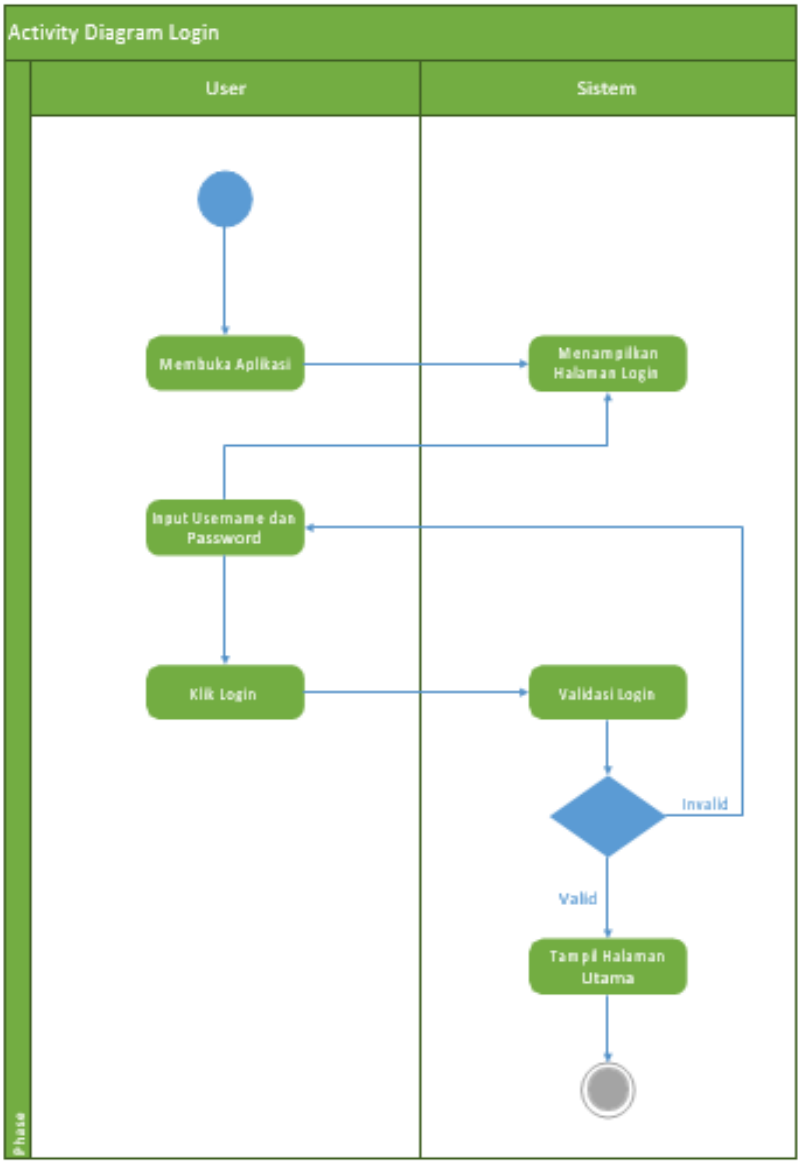
Gambar 4.5 *Activity Diagram Admin*

b. *Activity Diagram Login Admin*



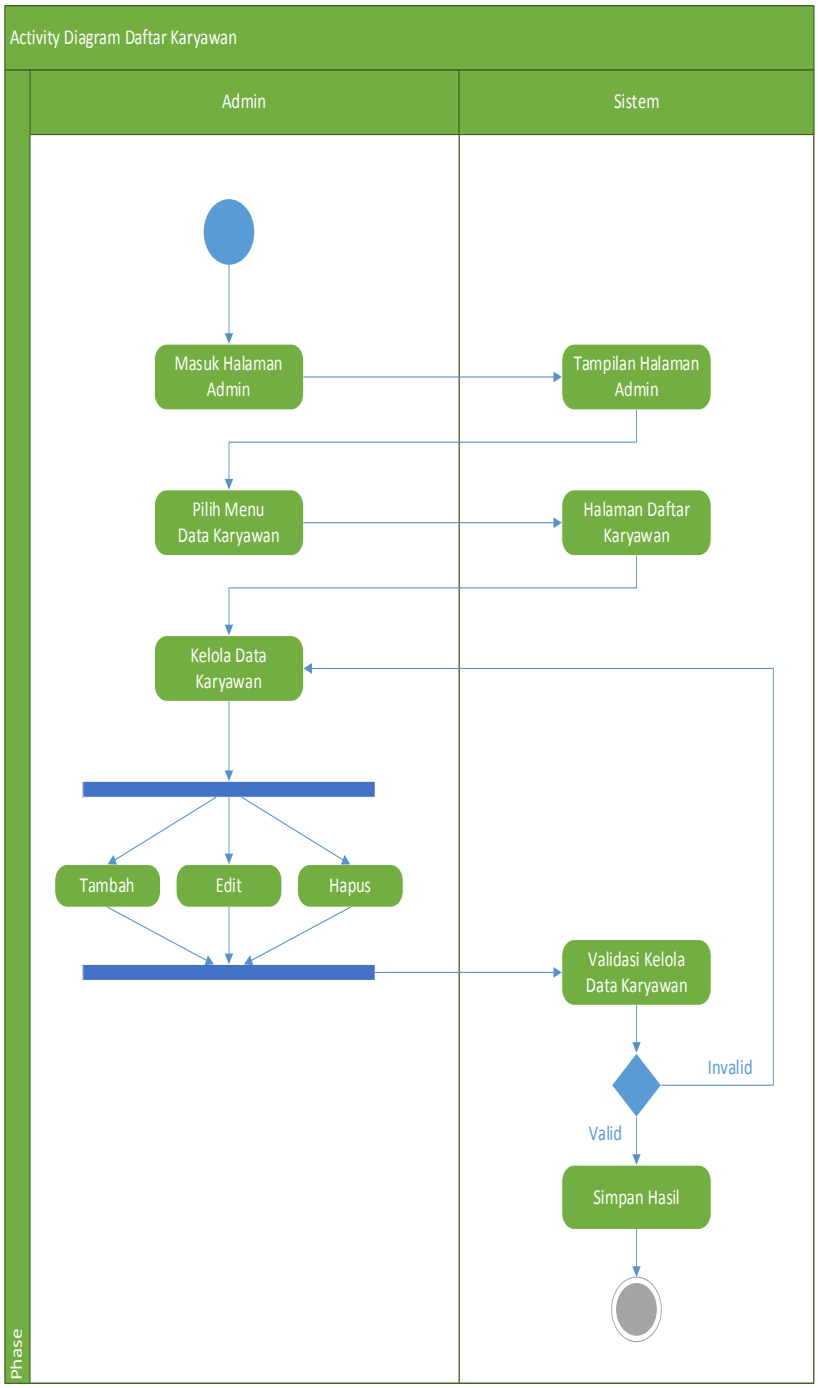
Gambar 4.6 *Activity Diagram Login Admin*

c. *Activity Diagram Login User*



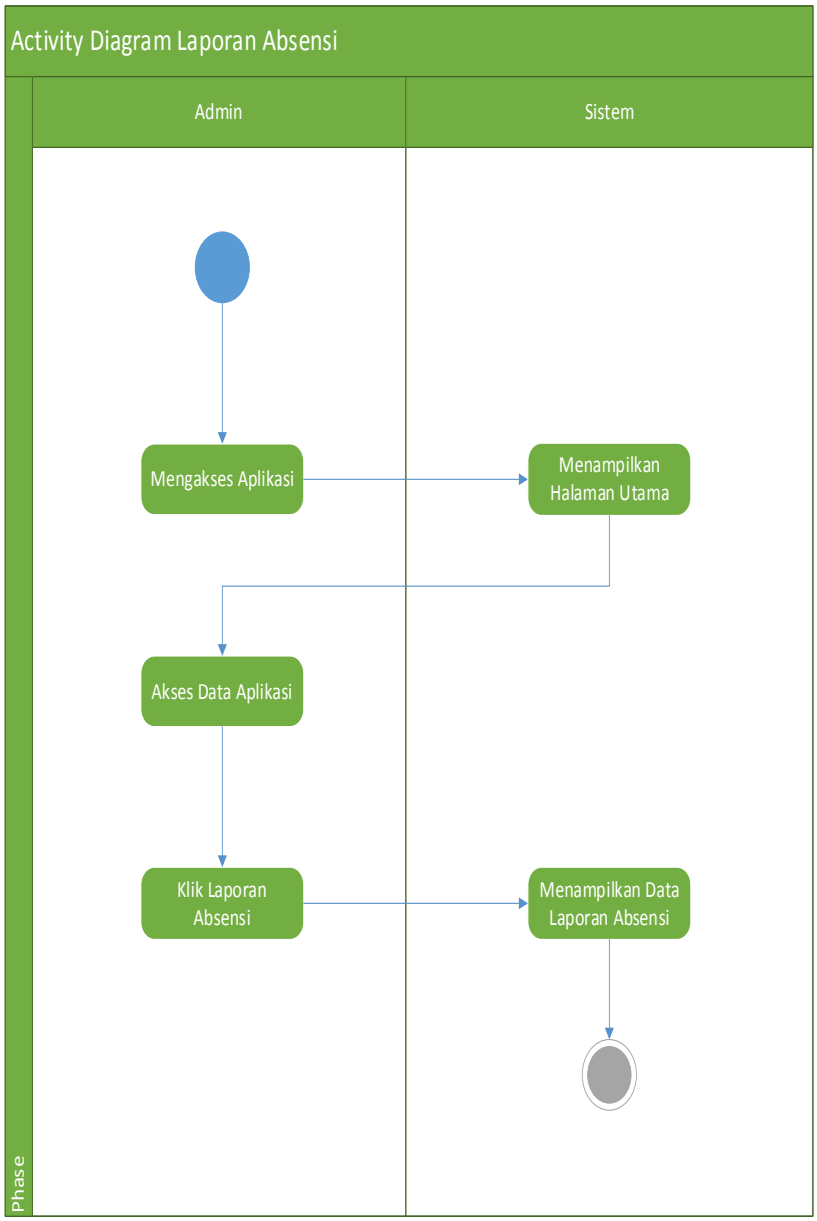
Gambar 4.7 *Activity Diagram Login User*

d. *Activity Diagram* Daftar Karyawan



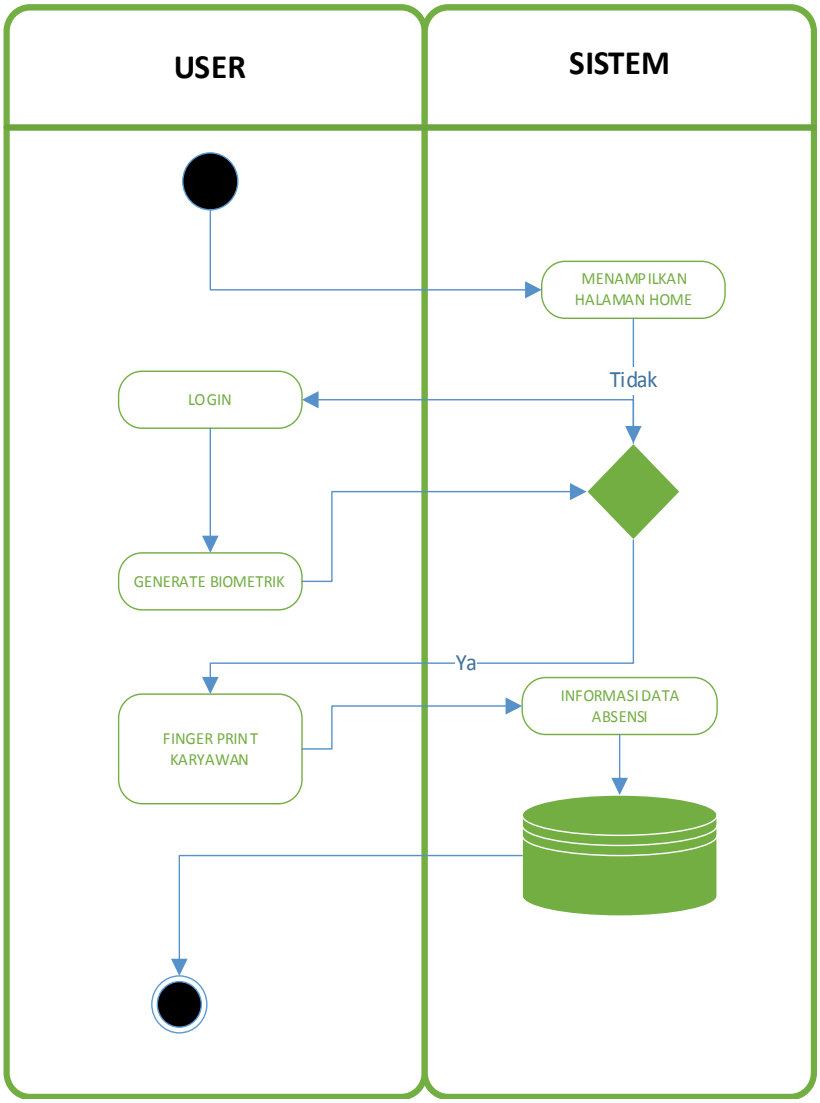
Gambar 4.8 *Activity Diagram* Data Karyawan

e. *Activity Diagram* Laporan Absensi



Gambar 4.9 *Activity Diagram* Laporan Absensi

f. *Activity Diagram User*

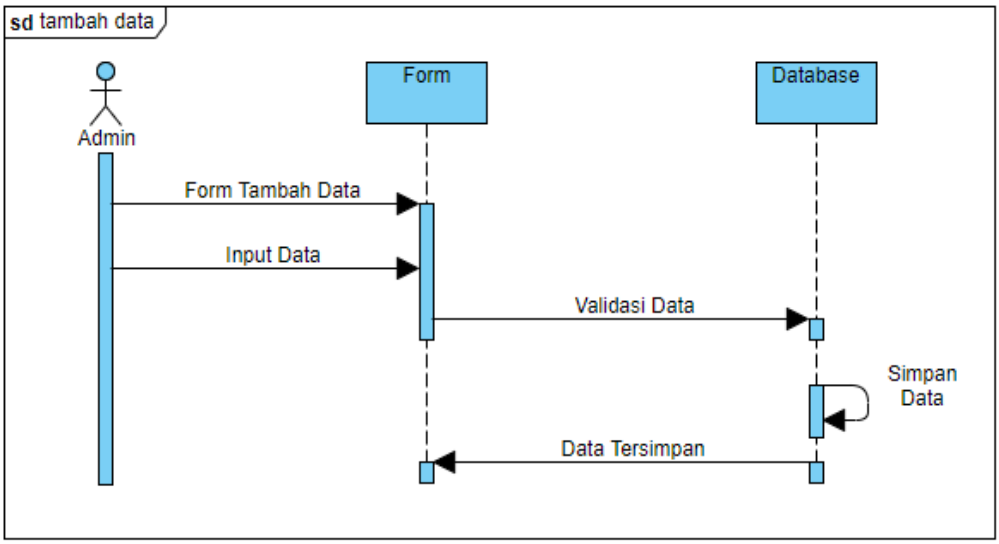


Gambar 4.10 *Activity Diagram User*

2. *Sequence Diagram*

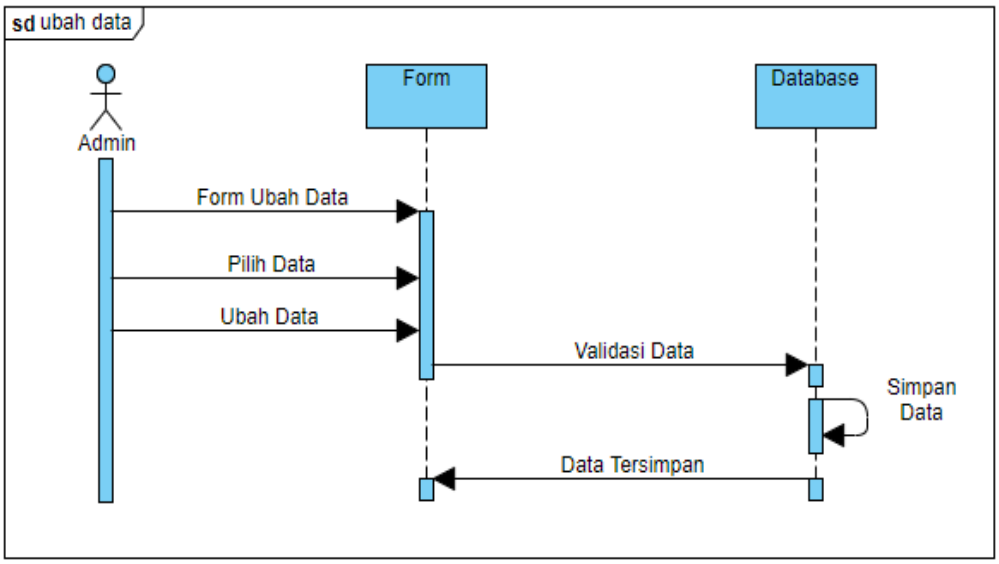
Sequence Diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. *Sequence Diagram* Tambah Data



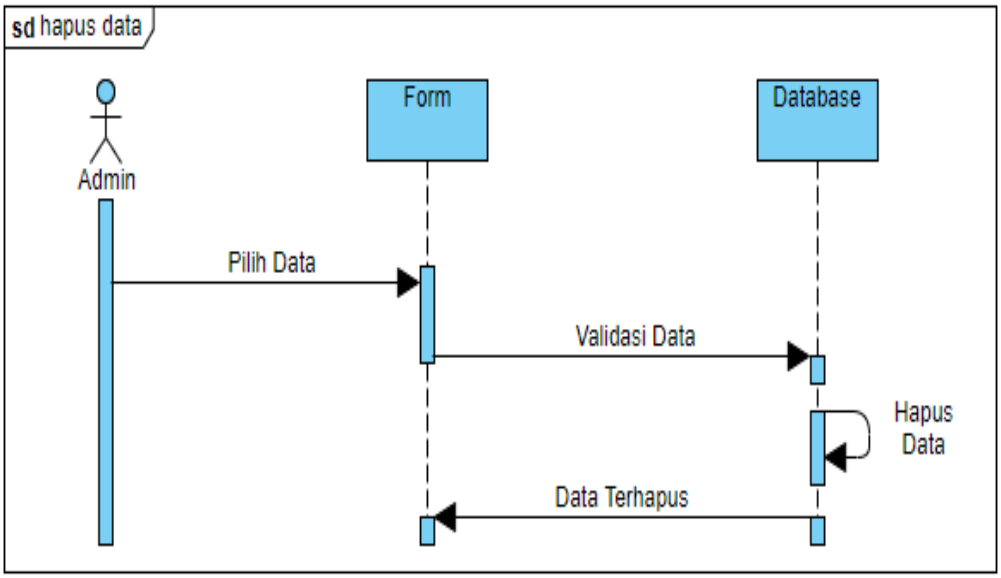
Gambar 4.11 *Sequence Diagram* Tambah Data

b. *Sequence Diagram* Ubah Data *Admin*



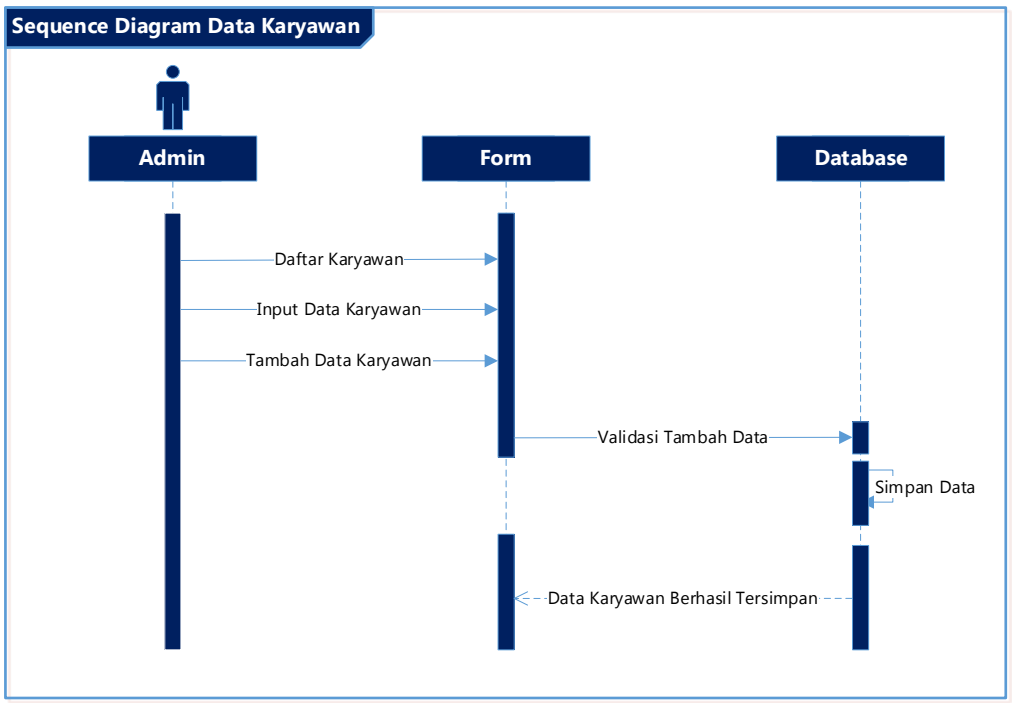
Gambar 4.12 *Sequence Diagram* Ubah Data

c. *Sequence Diagram* Hapus Data *Admin*



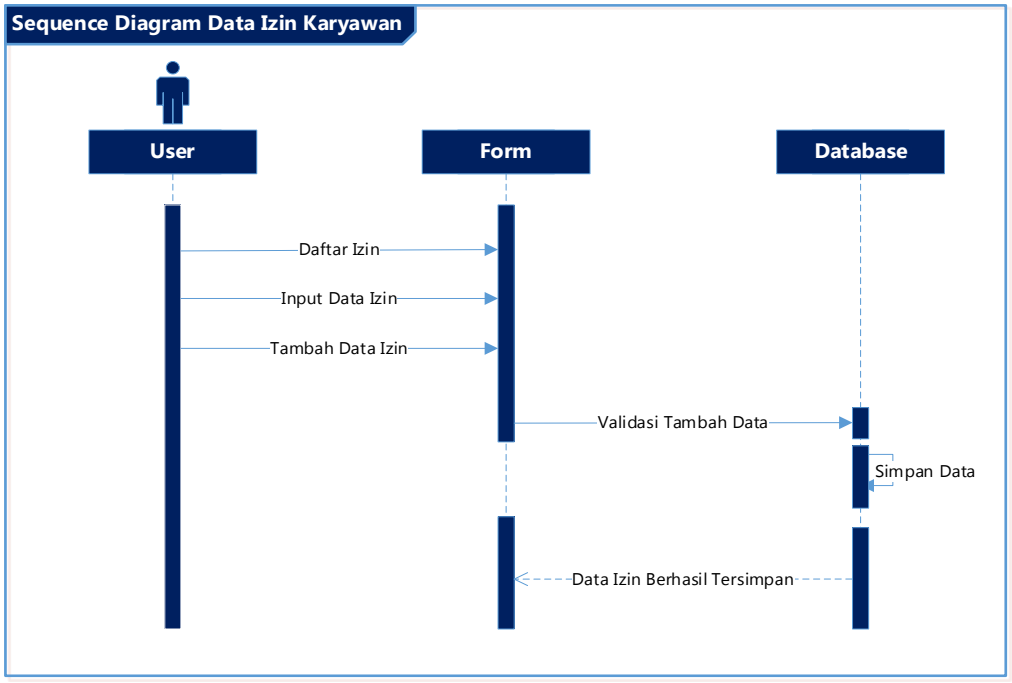
Gambar 4.13 *Sequence Diagram* Hapus Data

d. *Sequence Diagram* Data Karyawan



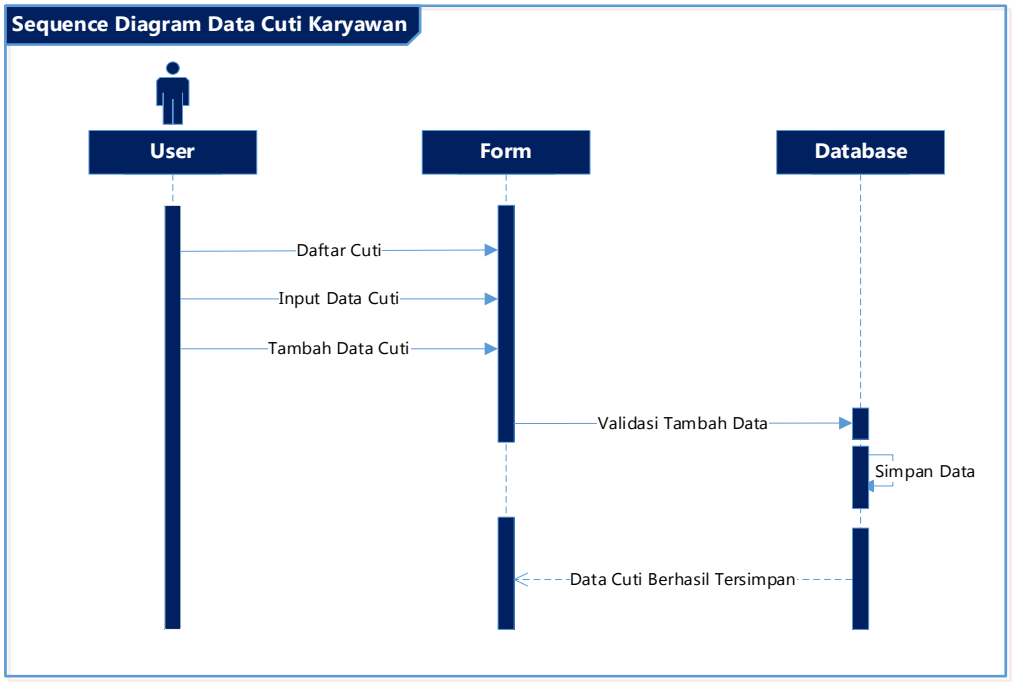
Gambar 4.14 *Sequence Diagram* Data Karyawan

e. *Sequence Diagram* Data Izin Karyawan



Gambar 4.15 *Sequence Diagram* Data Izin Karyawan

f. *Sequence Diagram* Data Cuti Karyawan

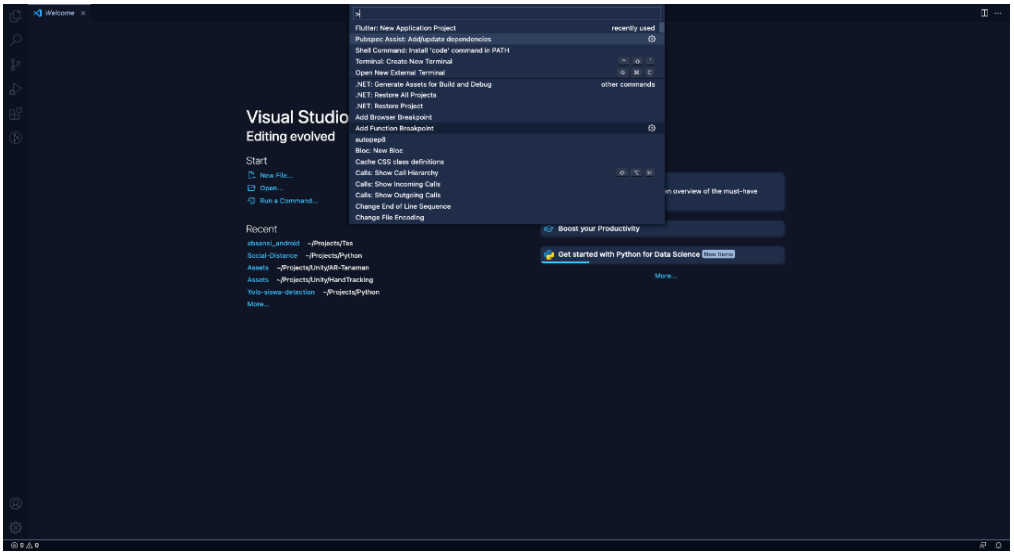


Gambar 4.16 *Sequence Diagram* Data Cuti Karyawan

2. Perancangan Sistem

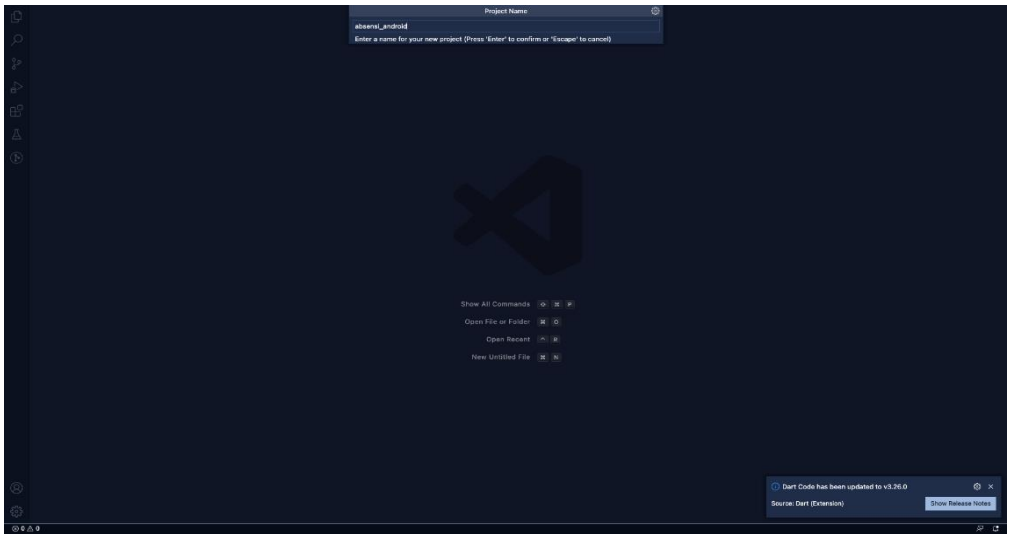
Berikut ini adalah *proses* gambaran secara garis besar dari *proses* perancangan *system*.

a. Buka *Visual Studio Code*, lalu tekan CTRL + SHIFT + P pada *keyboard* untuk membuat *project flutter* baru.



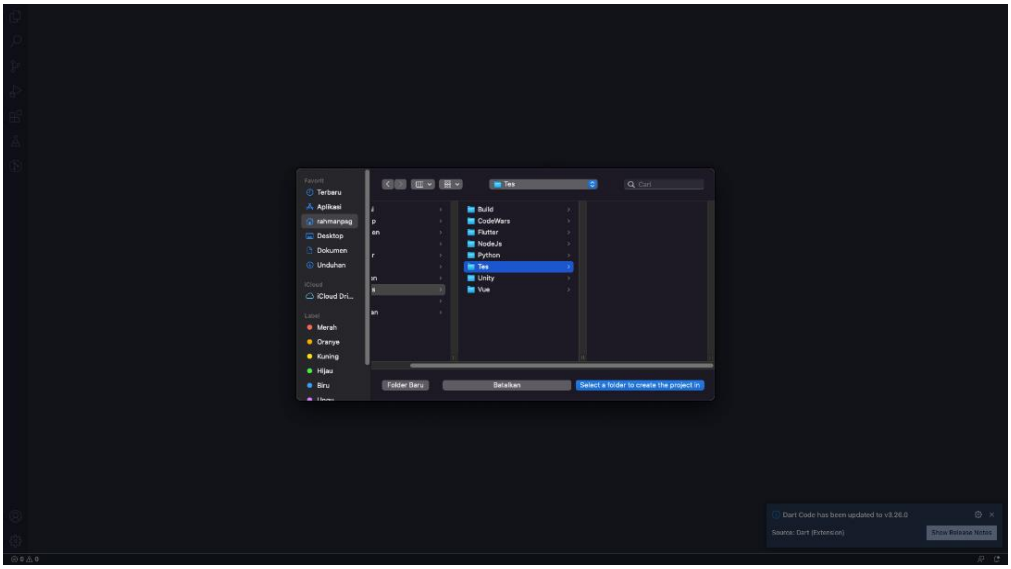
Gambar 4.17 Tampilan *Visual Studio Code*

b. Selanjutnya masukkan nama *project* yang akan dibuat.



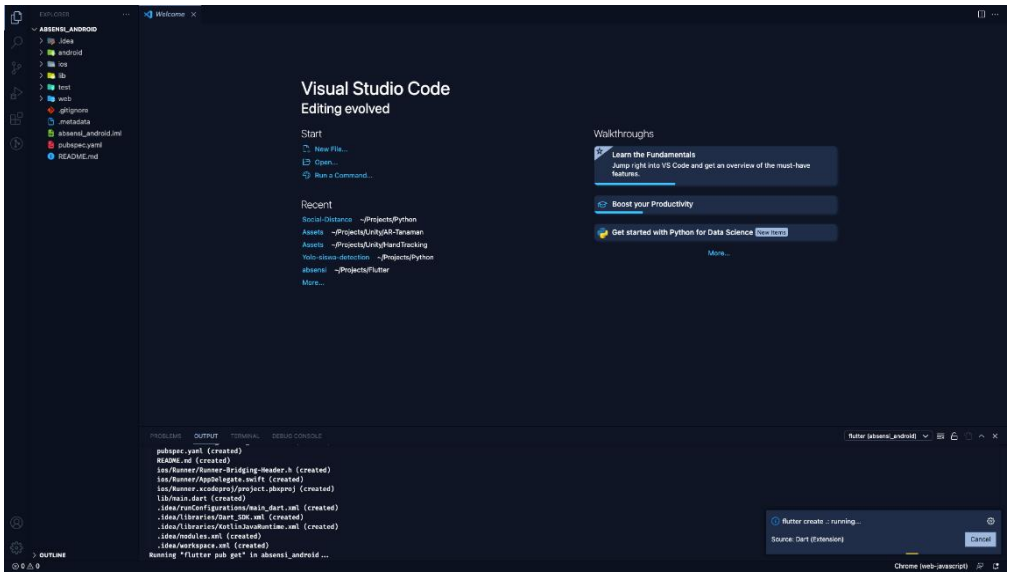
Gambar 4.18 Tampilan Nama Project

c. Pilih lokasi *project* yang akan disimpan



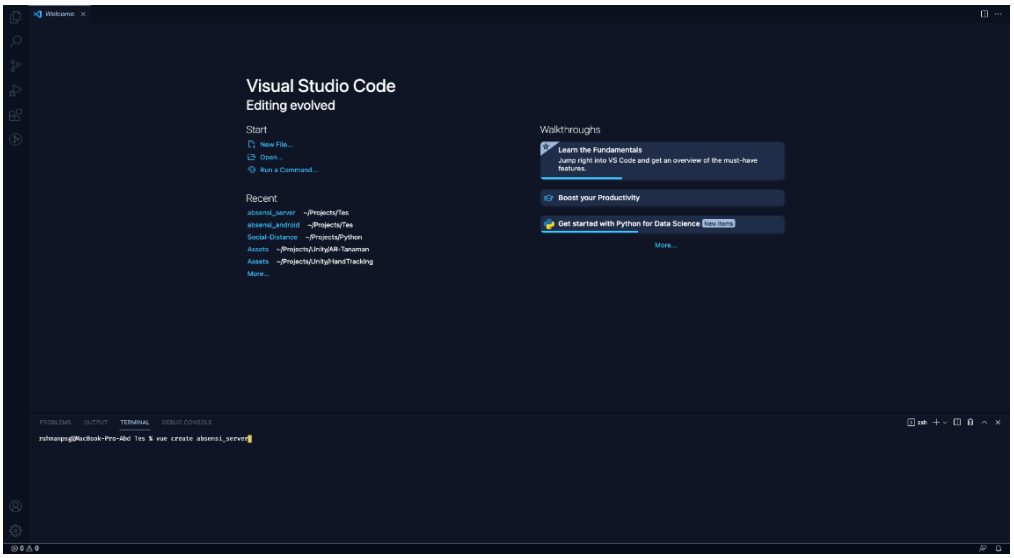
Gambar 4.19 Tampilan Penempatan Lokasi *Project*

d. *Project* baru *flutter* telah berhasil dibuat



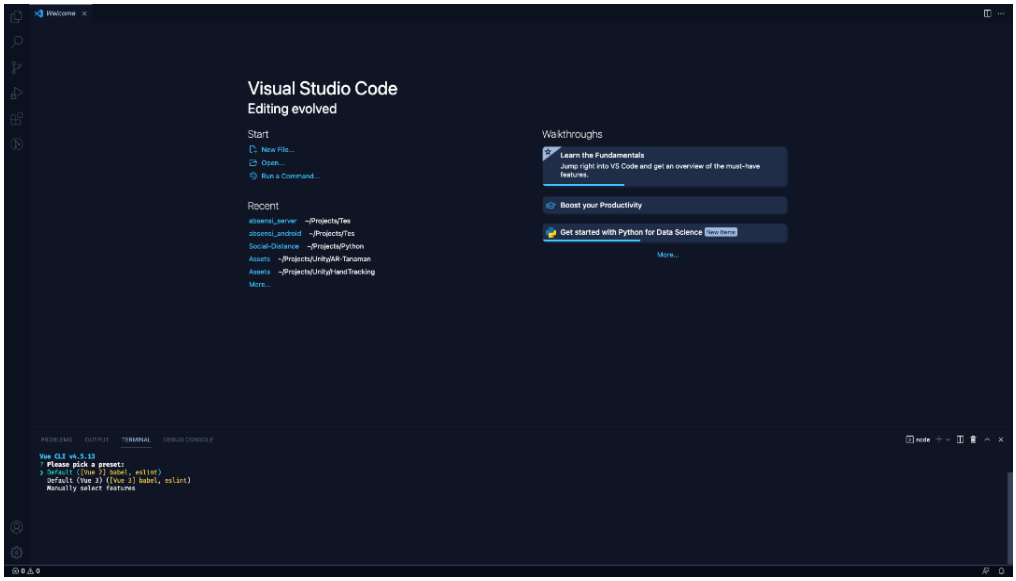
Gambar 4.20 Tampilan *New Project Flutter*

e. Buka *Visual Studio Code*, Lalu jalankan terminal kemudian ketik *vue create* nama_*project* lalu *Enter*.



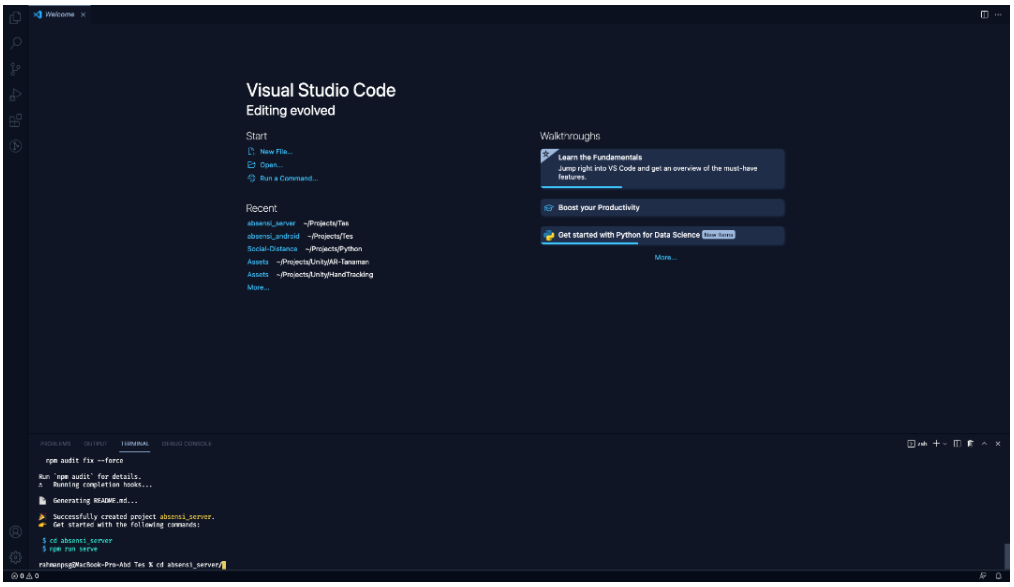
Gambar 4.21 Tampilan awal *Visual Studio Code*

f. Pada pilihan pembuatan *project vue*, silahkan pilih pada pilihan default lalu tekan tombol *Enter*.



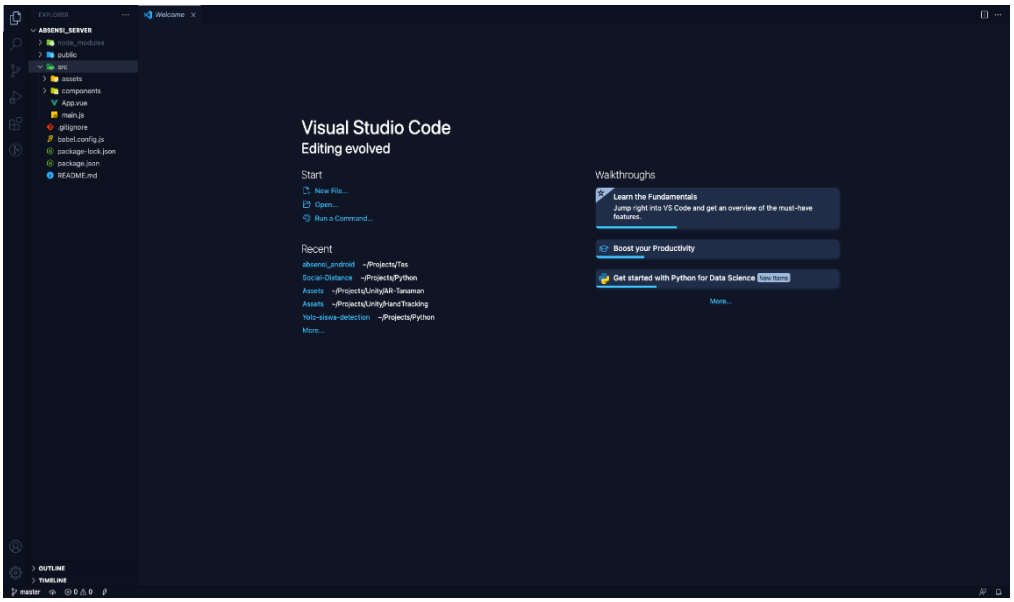
Gambar 4.22 Tampilan Proses Pembuatan *Projek Vue*

g. Setelah *project* berhasil dibuat ketik pada terminal : cd *nama_project* untuk mengubah lokasi terminal ke lokasi yang sama dengan *project* yang telah dibuat.



Gambar 4.23 Tampilan Setelah *Project* Berhasil Dibuat

h. Projek baru *vue* telah berhasil dibuat

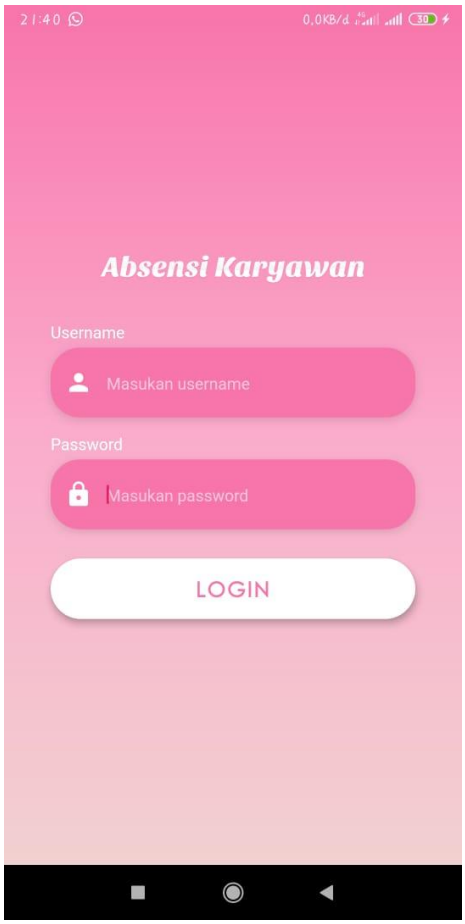


Gambar 4.24 Tampilan *New Project Vue*

3. Detail Aplikasi

1. Karyawan *(User)*

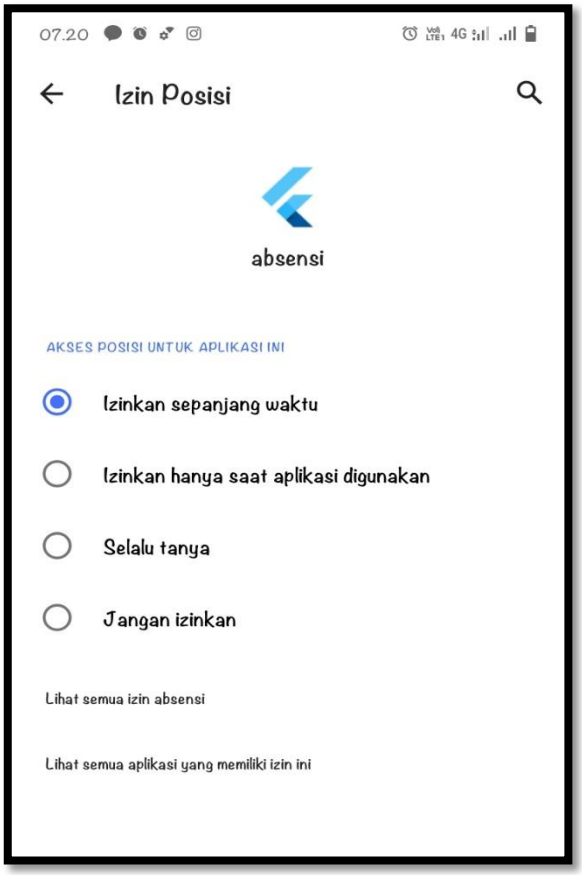
   a. Halaman *Login Admin*



Gambar 4.25 Halaman *Login Admin*

Halaman menu *login* merupakan halaman *login user* karyawan. *Login* merupakan tampilan awal yang ada pada sistem, yang digunakan *user* untuk masuk ke halaman selanjutnya.

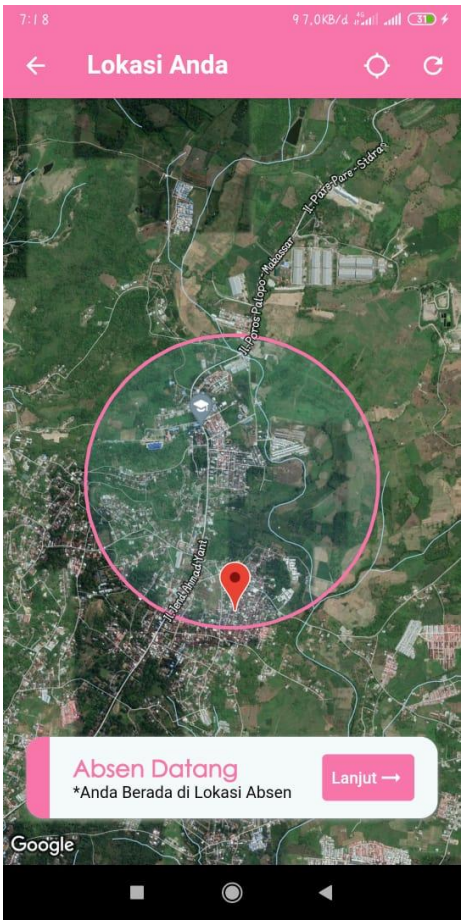
b. Halaman Permintaan Izin



Gambar 4.26 Halaman Permintaan Izin

Pada halaman perminaan izin ini, akan muncul permintaan izin dari aplikasi kepada pengguna/user untuk dapat mengakses lokasi bahkan ketika aplikasi sedang tidak dijalankan. Dalam permintaan izin ini, pengguna/*user* dalam hal ini adalah karyawan, harus memberikan izin agar aplikasi dapat mengakses lokasi karyawan sepanjang jam kerja, sehingga mendukung aplikasi untuk dapat memonitoring lokasi karyawan tersebut.

c. Halaman Menu Absen Hadir



Gambar 4.27 Halaman Menu Absen Hadir

Pada menu absen datang ini pengguna baru bisa menggunakan menu ini apabila dia berada dilokasi perusahaan dan tepat pada waktu yang telah ditentukan. Dan apabila telah lewat dari jam kerja maka tombol maupun fungsi tombol ini tidak aktif. Untuk memulai proses absensi datang klik lanjut.

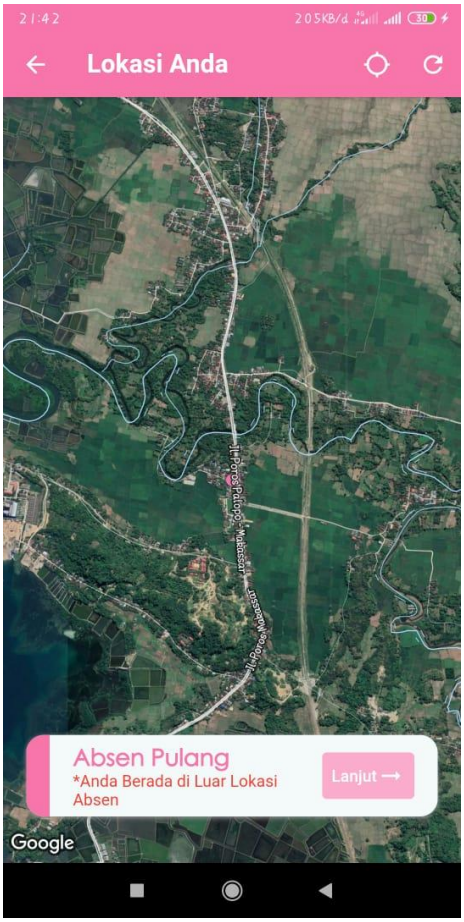
d. Halaman *Generate Biometrik*



Gambar 4.28 Halaman *Generate Biometrik*

Pada menu halaman *generate biometric* terdapat pengaksesan untuk *user*. Halaman ini akan tampil ketika telah berhasil absen/ menambah kehadiran karyawan. Pada halaman ini merupakan tampilan model header *Biometrik* yang akan diakses oleh karyawan.

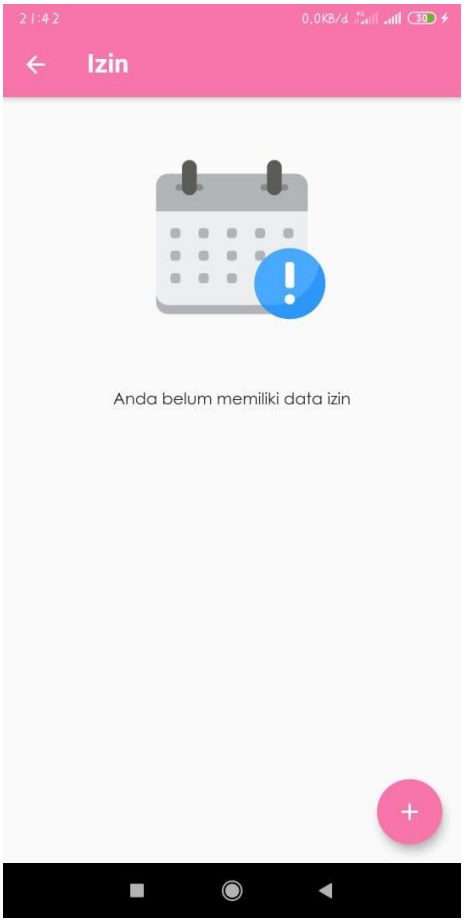
d. Halaman Menu Absen Pulang



Gambar 4.29 Halaman Menu Absen Pulang

Pada menu halaman absen pulang ini hanya dapat diakses apabila telah masuk waktu pulang kerja, dalam hal ini pengguna tidak dapat mengabsen jam pulang sebelum waktu pulang kerja yang telah ditentukan oleh perusahaan. Sama halnya dengan absen datang, tombol absen pada menu absen pulang ini hanya akan aktif dan dapat di akses apabila kita berada di lokasi perusahaan.

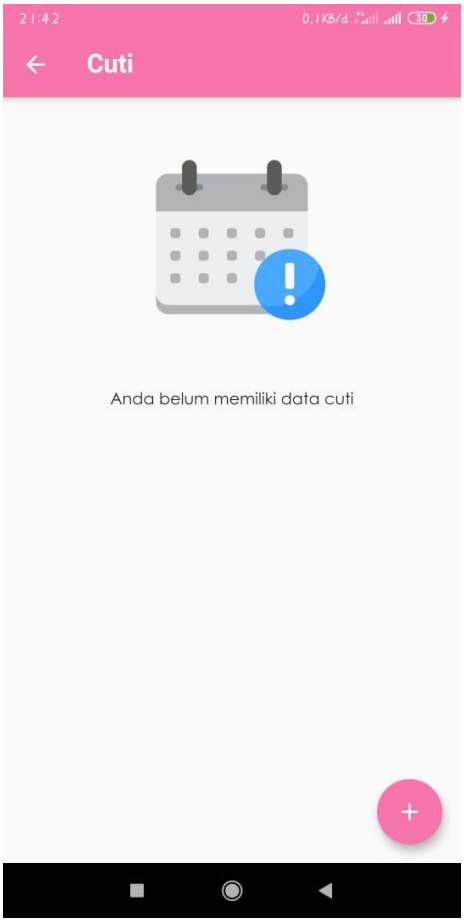
e. Halaman Menu Izin



Gambar 4.30 Halaman Menu Izin

Pada menu halaman menu izin disediakan apabila ada karyawan yang berhalangan atau tidak sempat hadir di tempat kerja dengan beberapa alasan tertentu, pengguna dapat menekan tombol tambah yang berada pada pojok kanan bawah untuk meng*Input* beberapa komentar lalu mengirimnya pada admin perusahaan sebagai pemberitahuan ketidakhadiran.

f. Halaman Menu Cuti



Gambar 4.31 Halaman Menu Cuti

Pada gambar diatas ini merupakan tampilan menu cuti yang hanya bisa diakses oleh *user*. Halaman notifikasi didalamnya terdapat informasi menu cuti, menu ini disediakan untuk memudahkan karyawan apabila ingin mengambil cuti selama beberapa hari, pengguna dapat menekan tombol tambah pada sudut kanan bawah untuk menambah komentar lalu mengirim ke admin perusahaan bersangkutan.

g. Halaman Menu Histori Absen

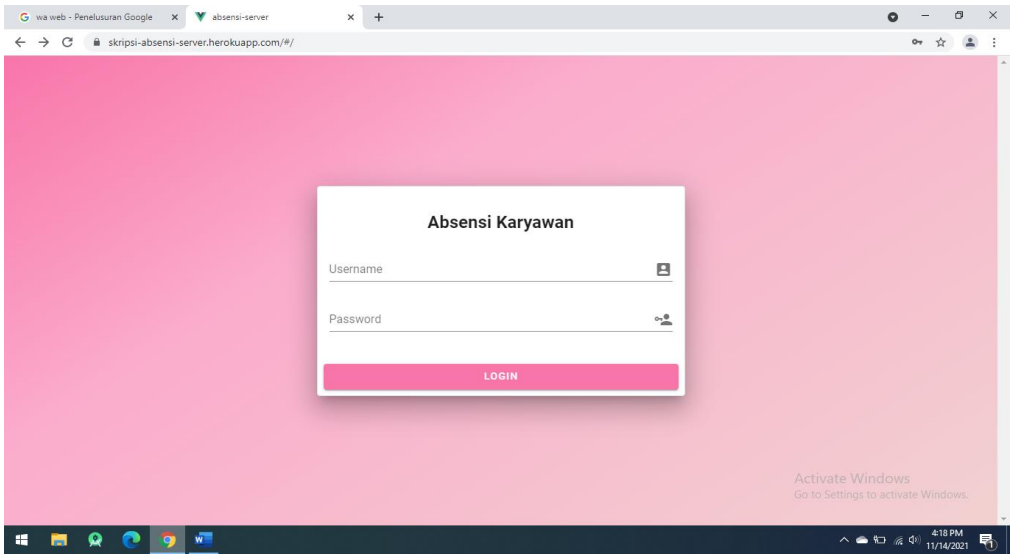


Gambar 4.32 Halaman Menu Histori Absen

Pada halaman ini merupakan menu untuk melihat semua arsip absen, baik hadir, izin, cuti maupun alfa. Pada histori absen ini telah dilengkapi fitur bulan dan tahun untuk memudahkan pengguna maupun admin untuk mengolah data kehadiran.

2. Personalia *(Admin)*

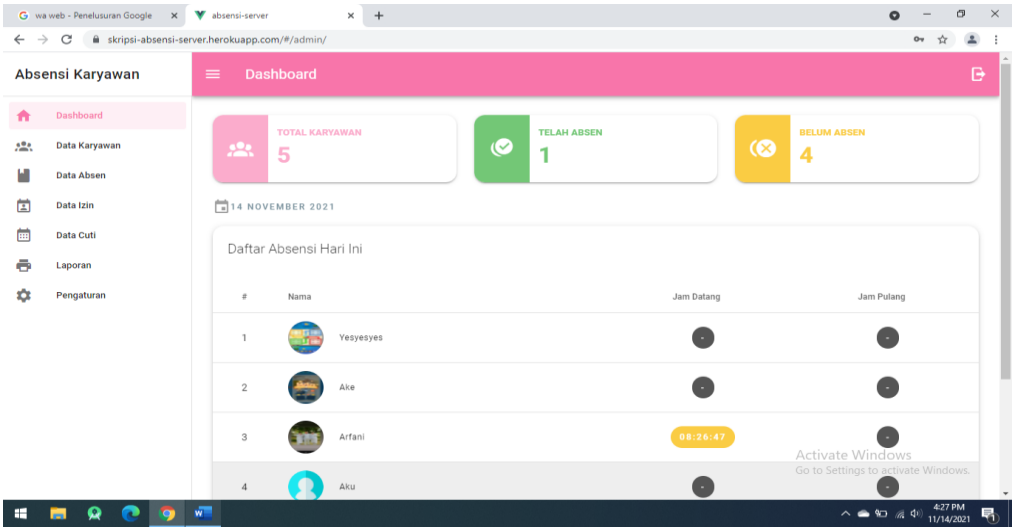
a. Halaman *Login Admin*



Gambar 4.33 Halaman *Login Admin*

Halaman menu *login* merupakan halaman *login admin*. *Login* merupakan tampilan awal yang ada pada sistem, yang digunakan *user* apabila telah mendapatkan *username* dan *password* untuk masuk ke halaman selanjutnya.

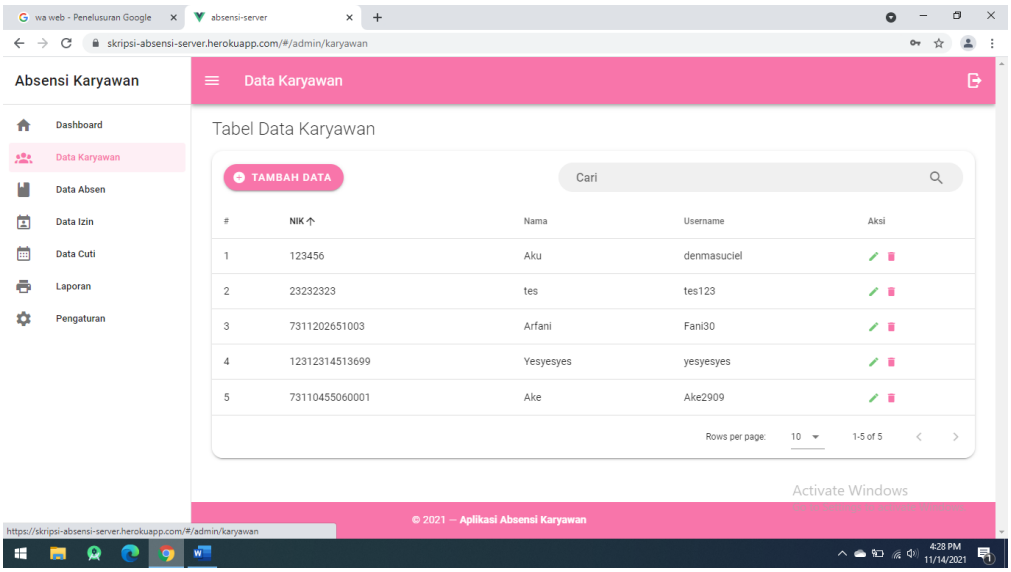
b. Halaman Menu *Home*



Gambar 4.34 Halaman Menu *Home*

Halaman menu *home* terdapat dalam halaman utama aplikasi, halaman ini digunakan *admin* untuk menampilkan data total karyawan, jumlah karyawan yang telah mengisi absensi dan jumlah karyawan yang belum mengisi absensi, serta daftar absensi pada hari tersebut.

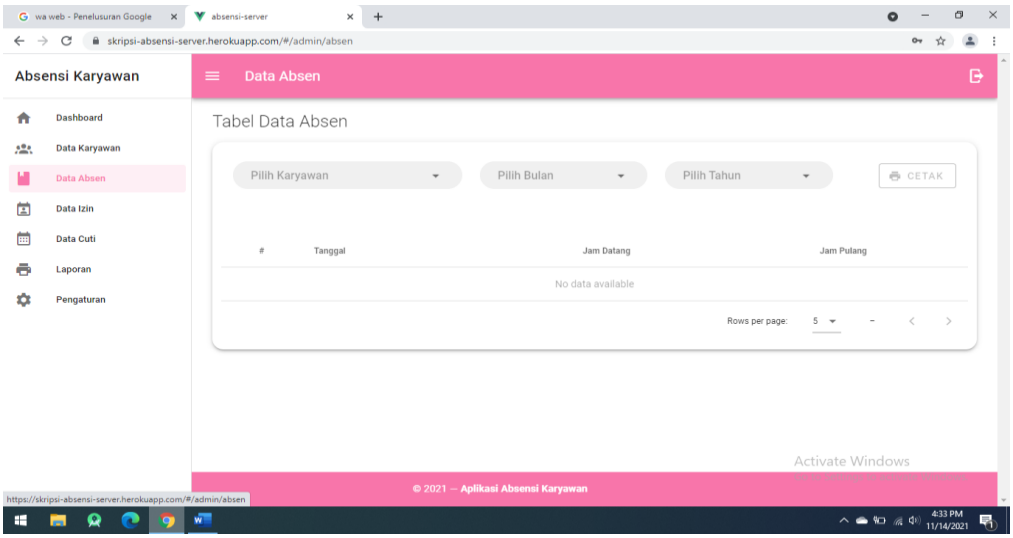
c. Halaman Menu Data Karyawan



Gambar 4.35 Tampilan Menu Data Karyawan

Pada halaman ini, seluruh daftar nama dan data para karyawan dapat di lihat oleh admin. Terdapat pula tombol tambah data untuk menambahkan data karyawan baru, juga terdapat kolom pencarian untuk memudahkan admin dalam mencari data karyawan dengan hanya mengetik nama karyawan yang dicari.

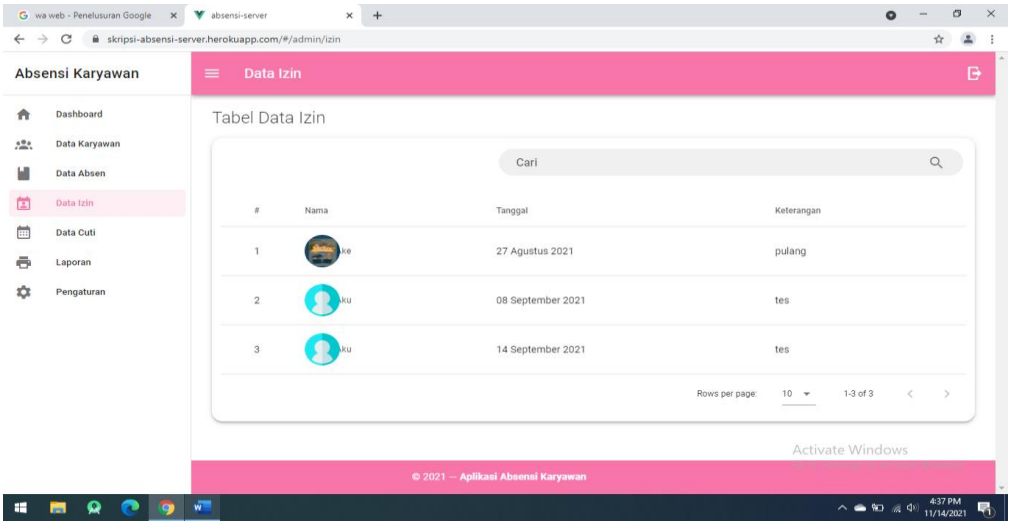
d. Tampilan Menu Data Absen



Gambar 4.36 Tampilan Menu Data Absen

Pada tampilan ini, admin dapat melihat dan memantau daftar rekap absensi karyawan, serta dapat mencetak daftar absensi tersebut untuk dijadikan arsip dalam bentuk print out. Admin hanya memilih bulan dan tahun absensi mana yang ingin di periksa dan di cetak, lalu tampillah nama karyawan dengan tanggal absensinya.

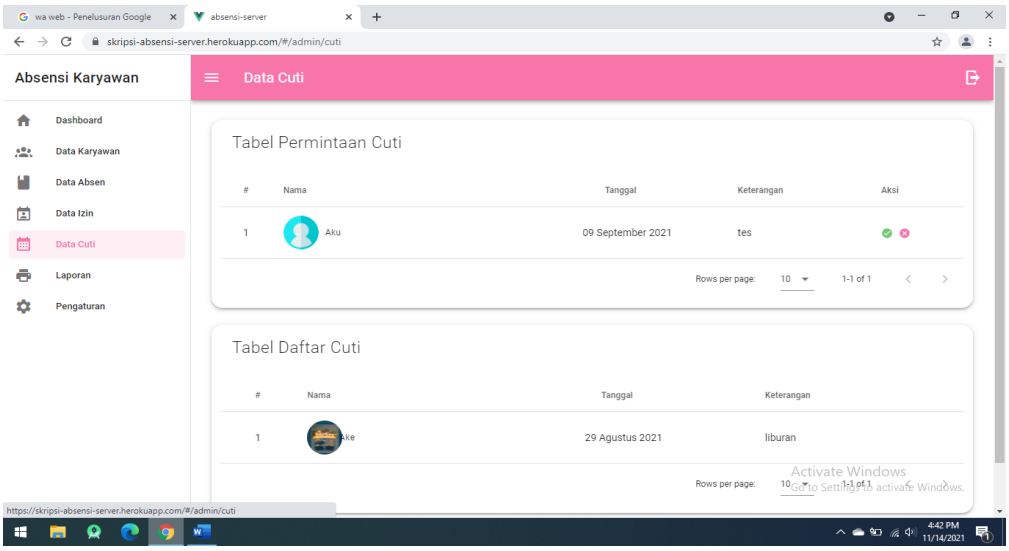
e. Tampilan Menu Izin



Gambar 4.37 Tampilan Menu Izin

Pada tampilan ini, admin dapat melihat daftar karyawan yang telah mengajukan izin beserta tanggal dan alasannya. Dalam tabel menu izin ini juga terdapat kolom pencarian untuk mencari data izin karyawan yang dituju untuk memudahkan admin mencari data.

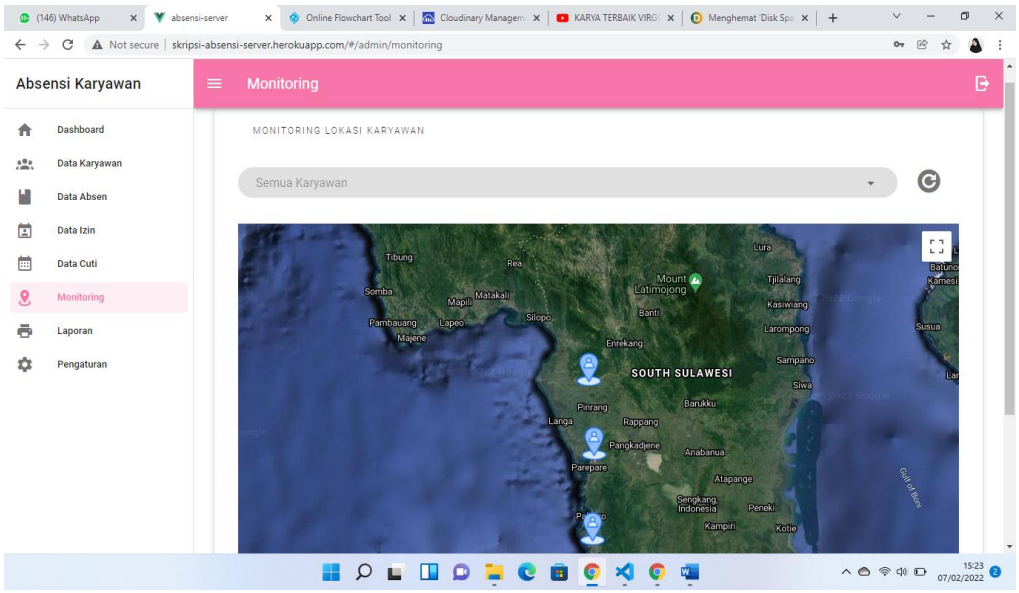
f. Tampilan Menu Pengajuan Cuti



Gambar 4.38 Tampilan Menu Pengajuan Cuti

Pada tampilan ini, admin dapat melihat daftar karyawan dan tanggal cuti yang di ajukan beserta keterangannya. Halaman ini juga terdapat tabel permintaan cuti. Tabel ini adalah tabel dimana admin dapat menyetujui atau menolak permintaan cuti yang diajukan karyawan, juga terdapat tabel daftar cuti. Tabel ini memperlihatkan data-data karyawan yang telah disetujui pengajuan cutinya.

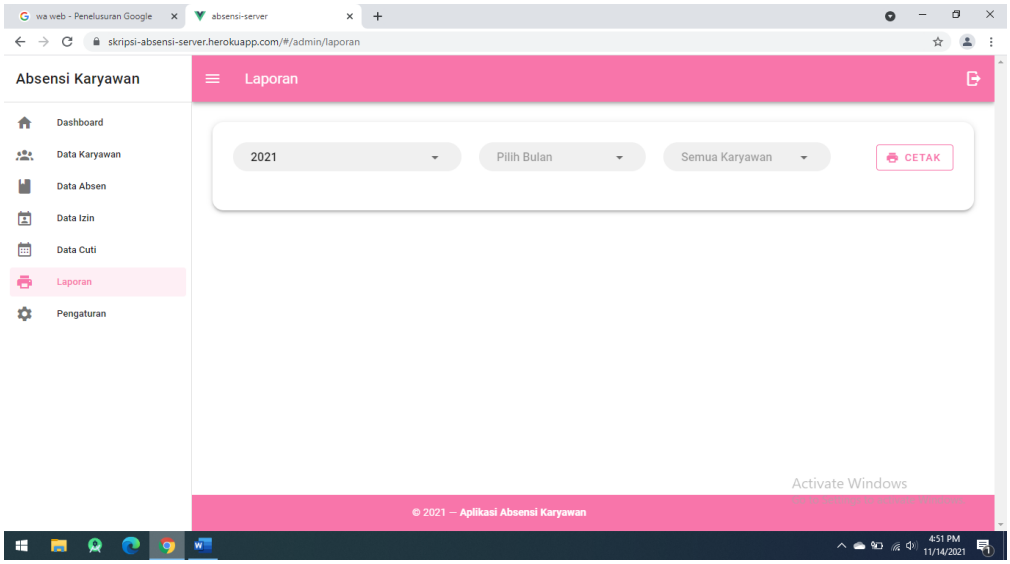
g. Tampilan Menu Monitoring Karyawan



Gambar 4.39 Tampilan Menu Monitoring Karyawan

Pada tampilan ini, admin dapat melihat lokasi karyawan yang sedang tidak berada di dalam perusahaan atau titik lokasi yang ditentukan. Pada halaman ini, admin dapat mengontrol dan memonitoring kedisiplinan karyawan demi mengurangi terjadinya kecurangan dalam proses pencatatan kehadiran. Fitur ini mengharuskan pengguna untuk memberikan izin mengakses lokasi pada perangkat *android mobile* mereka, agar sistem dapat mendeteksi lokasi dari perangkat *android mobile* karyawan.

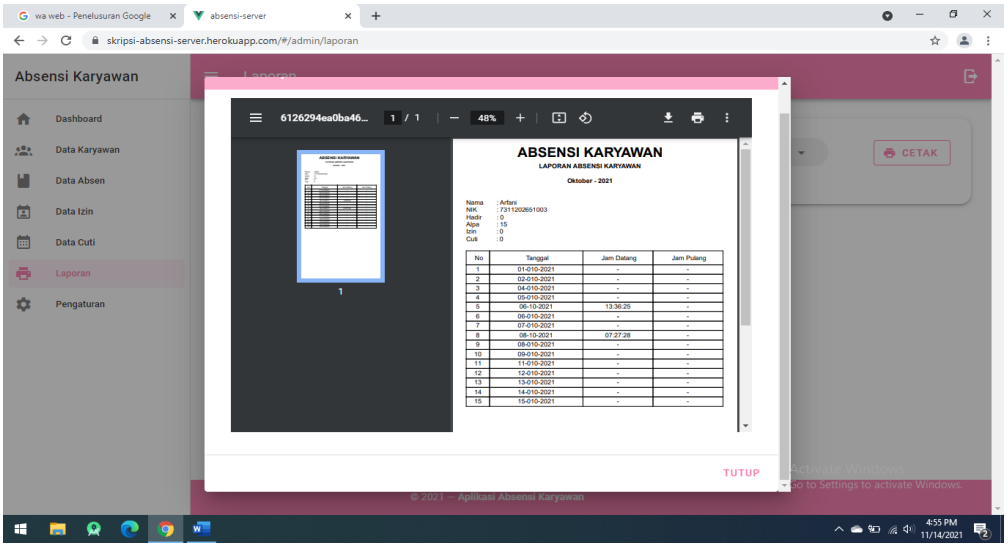
h. Tampilan Menu Laporan



Gambar 4.40 Tampilan Menu Laporan

Pada tampilan ini, admin dapat memeriksa sekaligus langsung mencetak hasil rekap laporan per bulan setiap tahunnya, dan mempunyai pilihan untuk mencetak laporan absensi untuk semua karyawan atau per satu orang karyawan saja.

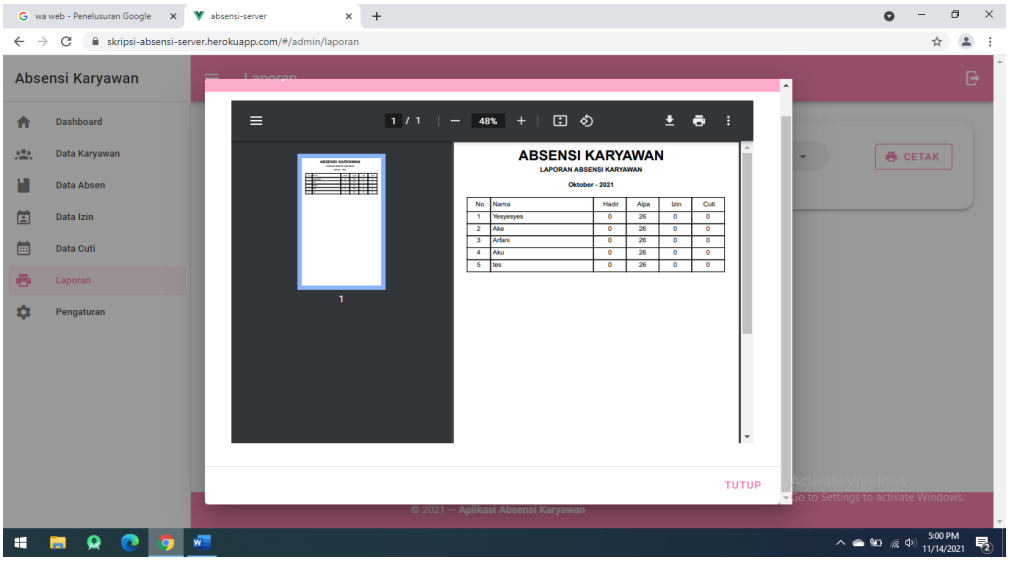
i. Tampilan cetak laporan per satu orang karyawan



Gambar 4.41 Tampilan cetak laporan per satu orang karyawan

Pada halaman ini merupakan tampilan laporan rekap absensi yang siap cetak. Isinya adalah rekapan absen per satu bulan untuk satu karyawan.

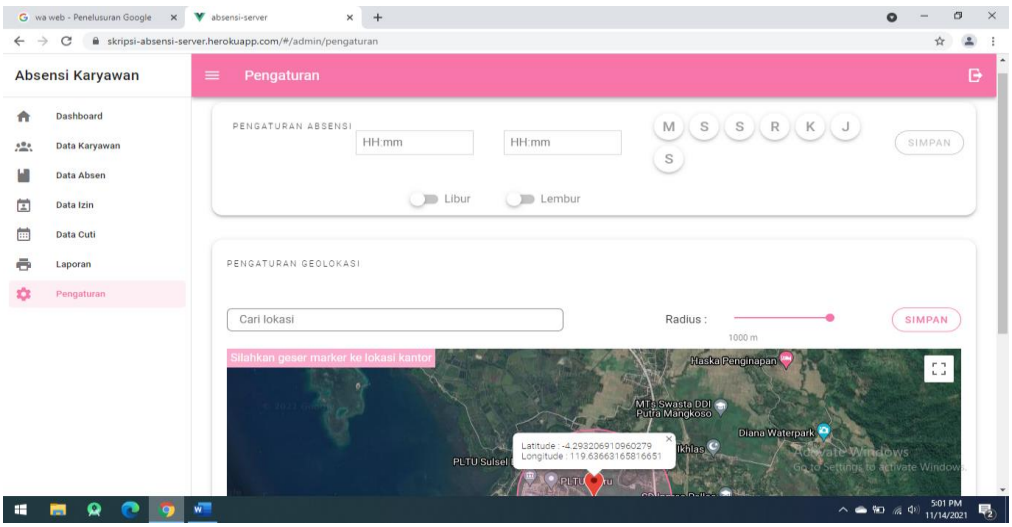
j. Tampilan Cetak Laporan seluruh karyawan



Gambar 4.42 Tampilan cetak laporan seluruh karyawan

Pada halaman ni merupakan tampilan laporan rekap absen yang siap cetak. Isinya adalah rekapan absen per satu bulan untuk seluruh karyawan.

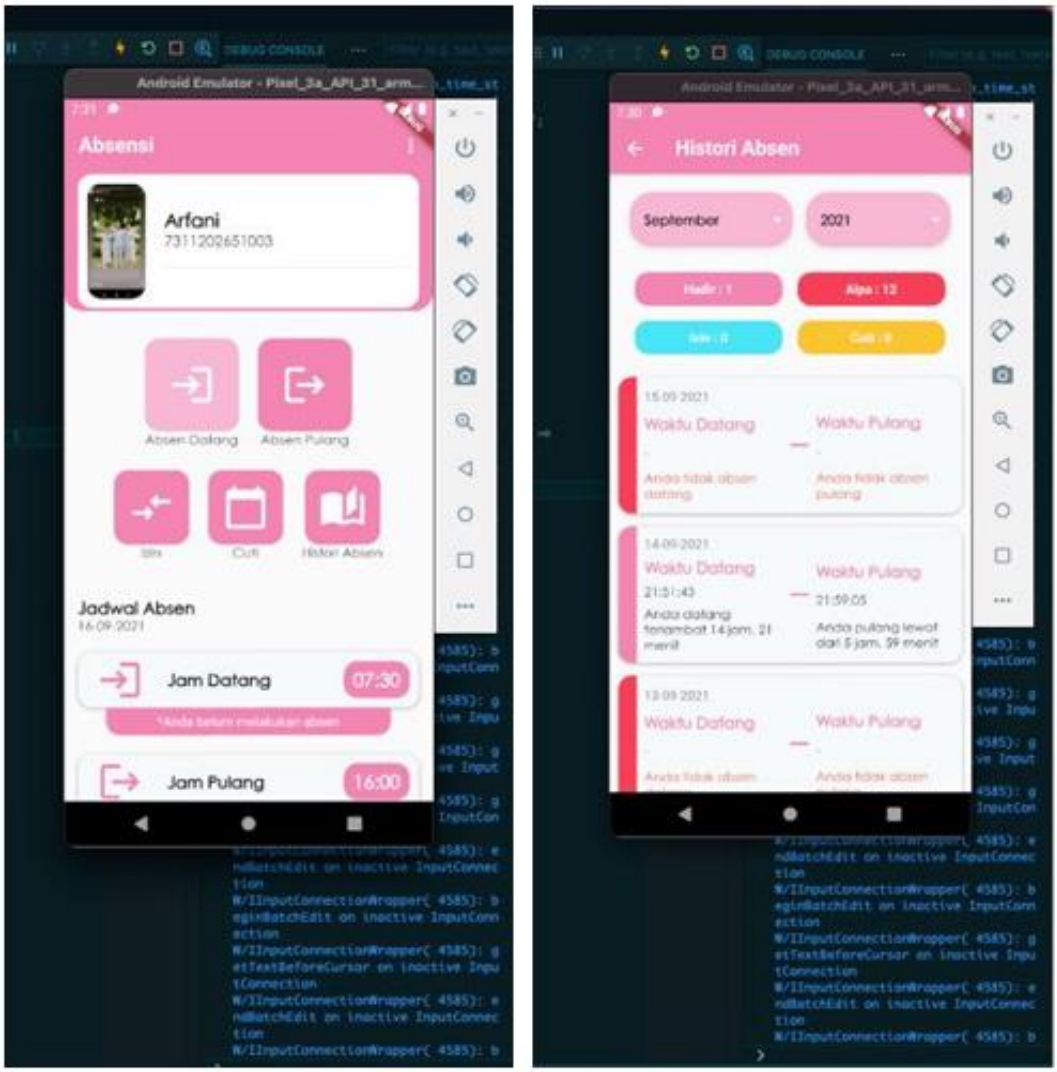
k. Tampilan Menu Pengaturan Aplikasi



Gambar 4.43 Tampilan Menu Pengaturan Aplikasi

4. Rancangan *Output* Program

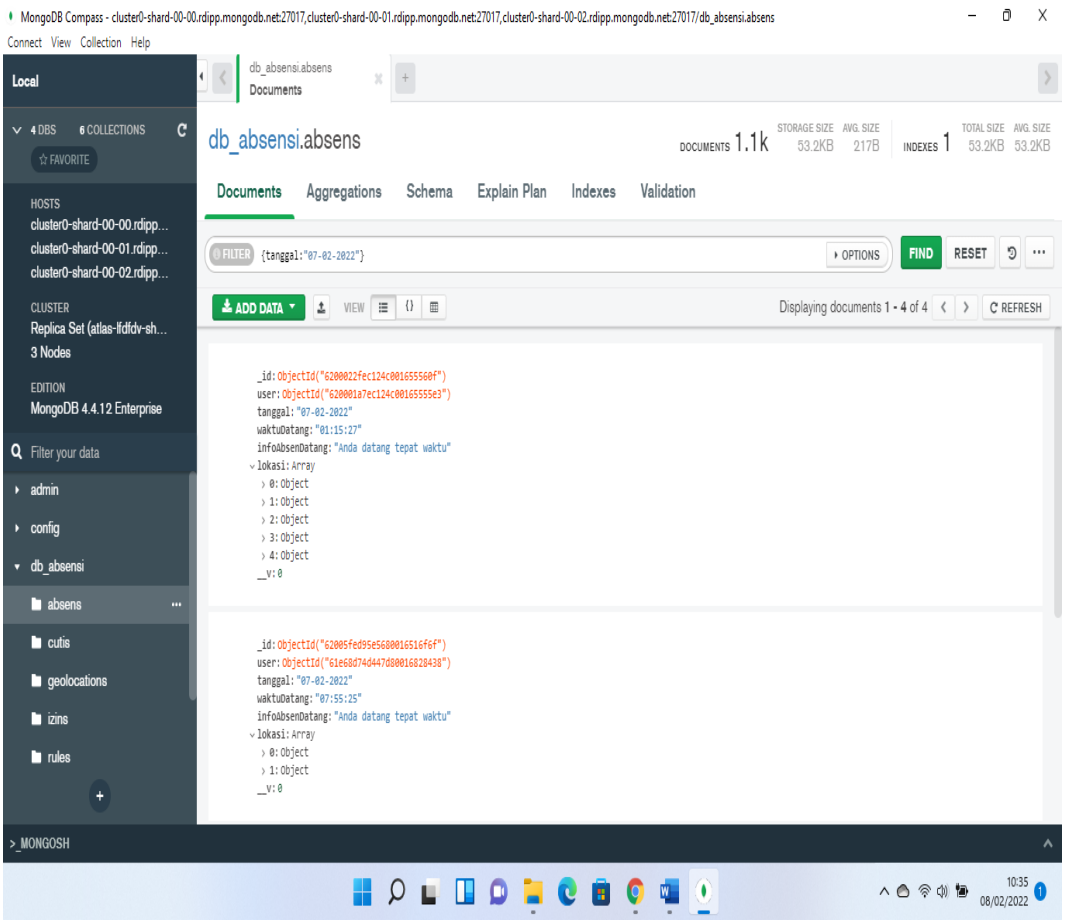
Perancangan *output* program merupakan perancangan yang dilakukan untuk mengetahui informasi dan data-data kehadiran karyawan berupa Aplikasi Absensi Karyawan Menggunakan *Geolocation* dan *Finger Print* Berbasis *Android.*



Gambar 4.44 Halaman Absensi

5. *Database* Aplikasi

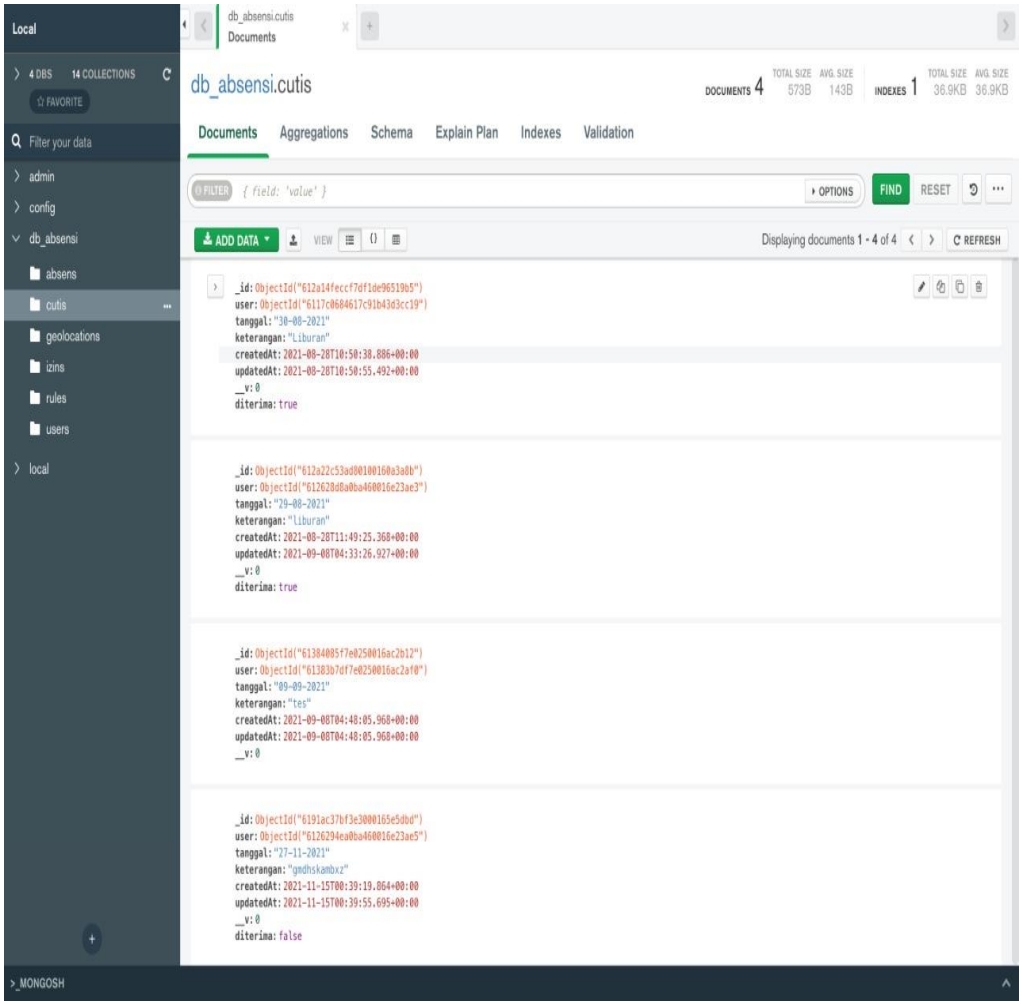
1. Tampilan *Database* histori Absen



Gambar 4.45 Tampilan Database Histori Absen

Tampilan diatas merupakan tampilan database histori absen. Setiap karyawan melakukan proses pengabsensian, seluruh data-data absensinya akan tersimpan rapi di database ini. Mulai dari pukul berapa karyawan mengisi absen masuk, pukul berapa karyawan mengakses absen pulang, tanggal absen dan keterangan keterlambatan.

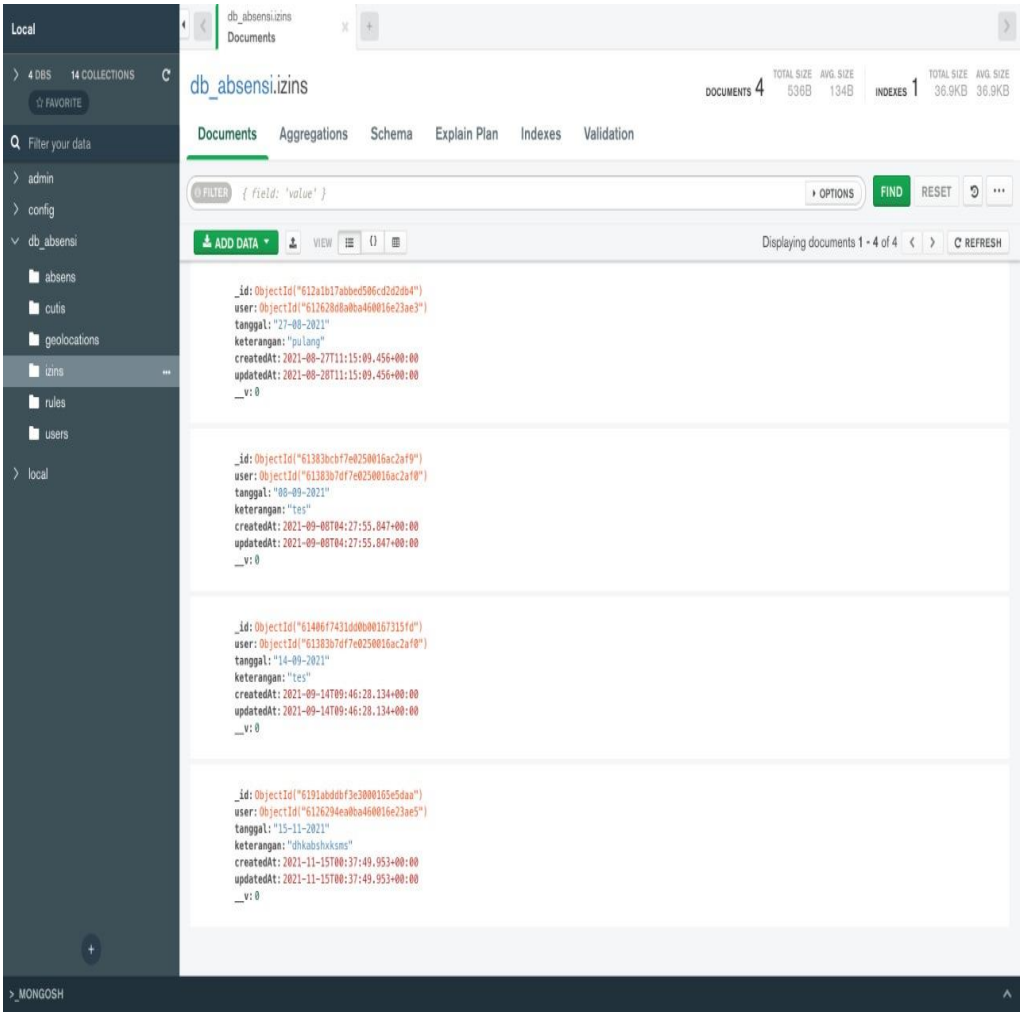
2. Tampilan *Database* Cuti



Gambar 4.46 Tampilan Database Cuti

Tampilan diatas merupakan tampilan database cuti. Setiap karyawan yang melakukan pengajuan cuti, seluruh datanya akan tersimpan rapi di database ini. Mulai dari tanggal pengajuan cuti, tanggal cuti yang diajukan, alasan pengajuan cuti dan keterangan konfirmasi pengajuan cuti ditolak atau diterima oleh admin.

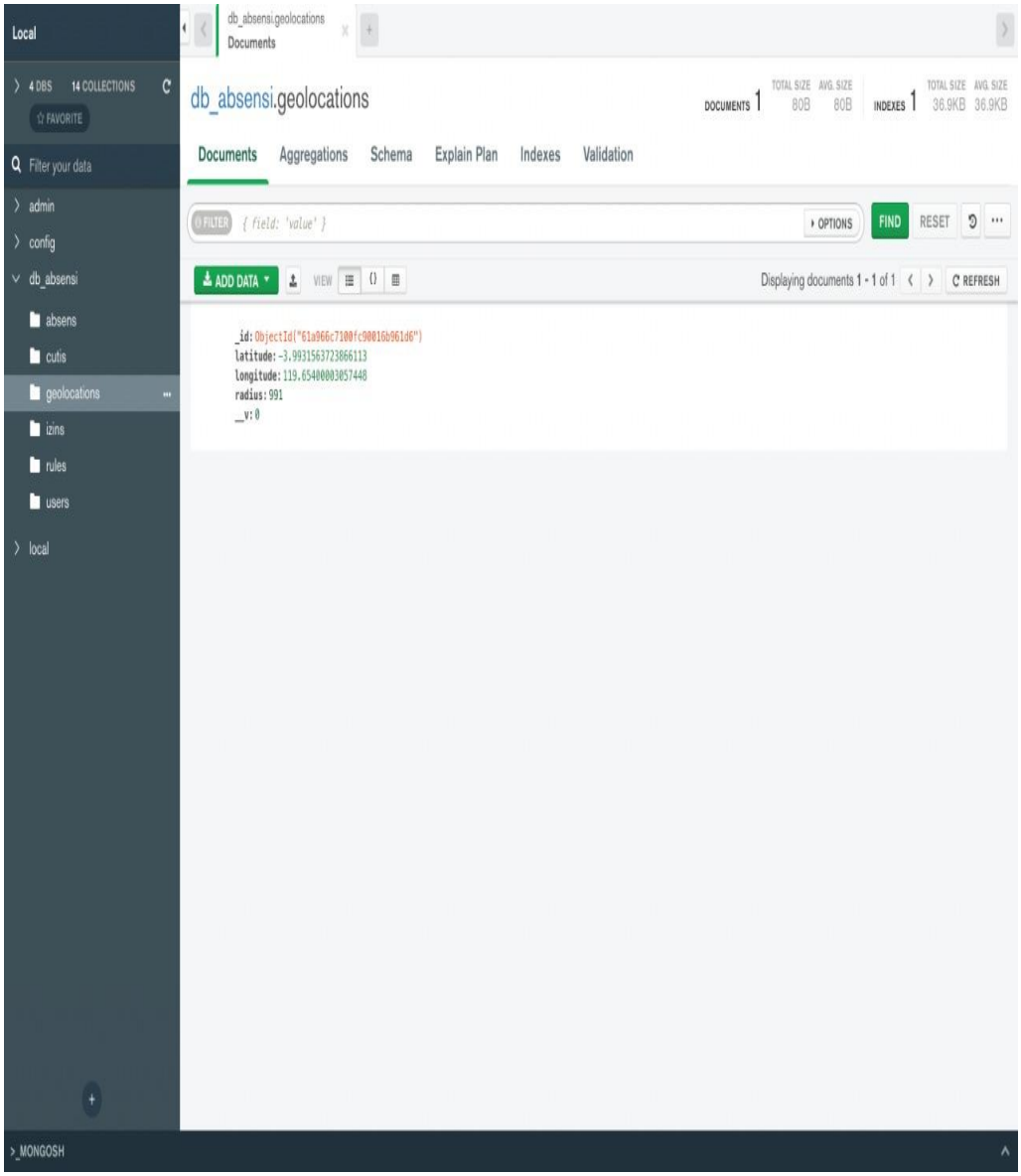
3. Tampilan Database Izin



Gambar 4.47 Tampilan Database Izin

Tampilan diatas merupakan tampilan database Izin. Seluruh karyawan yang mengisi form izin akan tercatat datanya di database ini. Mulai dari tanggal berapa karyawan izin, pukul berapa karyawan izin, serta alasan karyawan melakukan izin.

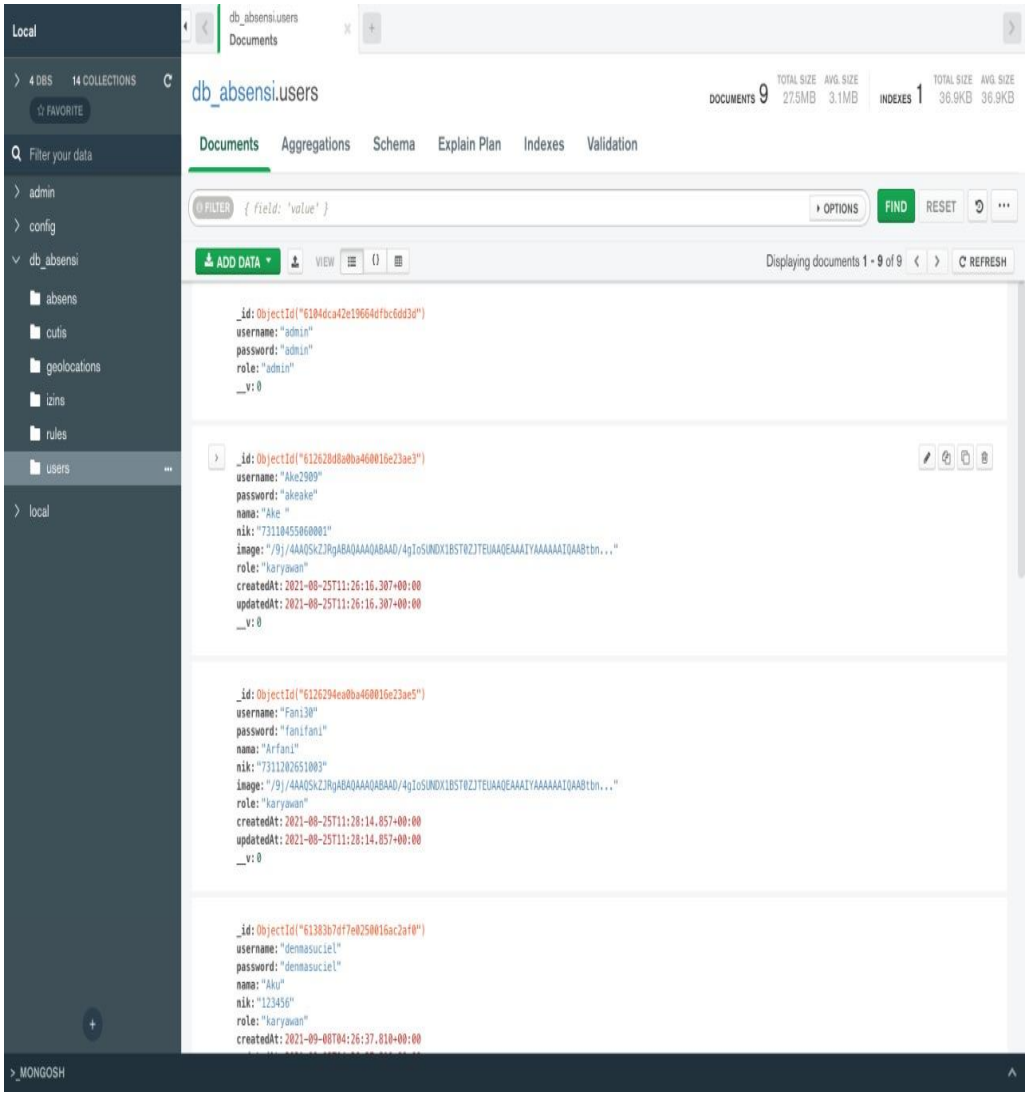
4. Tampilan Database Geolocation



Gambar 4.48 Tampilan Database Geolacation

Tampilan diatas merupakan tampilan database geolocation. Database geolocation menyimpan data titik koordinat lokasi kantor yang ditentukan. Mulai dari nilai latitude, longitude dan radius.

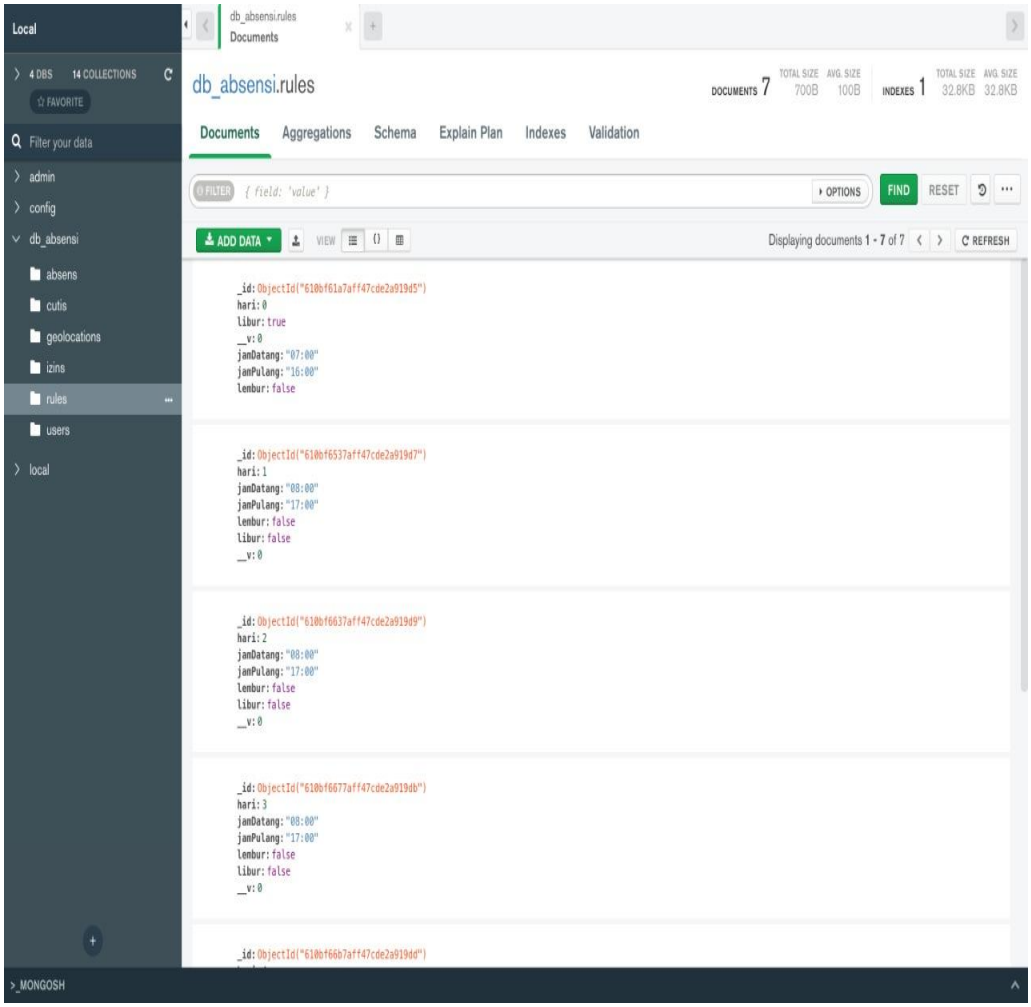
5. Tampilan Database User



Gambar 4.49 Tampilan Database User

Tampilan diatas merupakan tampilan database user. Seluruh data-data karyawan yang telah diinput dan mempunyai akses login tersimpan rapi di database ini. Mulai dari username dan password yang dibuat, nama karyawan, nomor induk karyawan, nama gambar yang diinput, dan tanggal diinputnya data karyawan tersebut.

6. Tampilan Database Rules



Gambar 4.50 Tampilan *Database Rules*

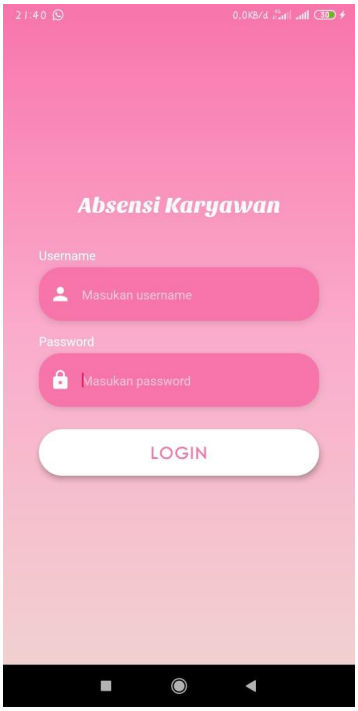
Tampilan diatas merupakan tampilan *database rules* atau aturan. Database ini menyimpan data-data pengaturan absensi yang telah diatur oleh admin melalui sistus web aplikasi. Data-data yang tersimpan berupa hari kerja dengan masing-masing jam kerja, keterangan libur, dan jam absensi dapat di akses.

6. Pengujian Sistem

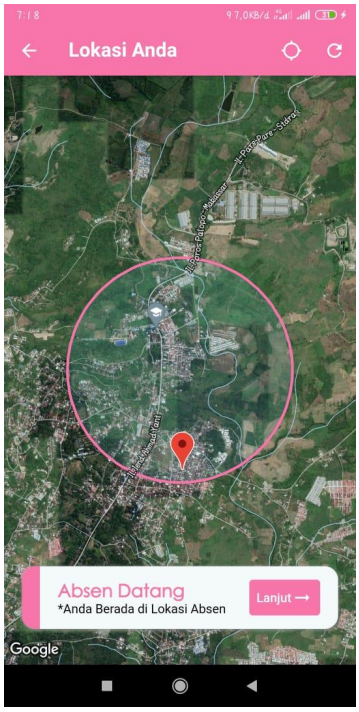
Metode pengujian perangkat lunak pada sistem ini menggunakan pengujian *Black Box, White Box* dan pengujian Akurasi.

a) Pengujian *Black Box*

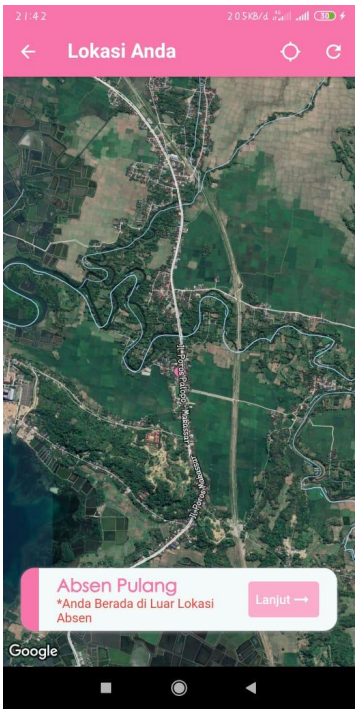
Pengujian *Black Box* didasarkan pada detail aplikasi, fungsi – fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan *proses* yang diinginkan oleh pengguna atau *user* pengujian ini tidak melihat dan menguji Logika program.

	st Faktor	**asil**	**eterangan**
	Menu Login	✓	rhasil, karena ketika aplikasi dibuka maka akan muncul tampilan *splash screen* menu Login.
			

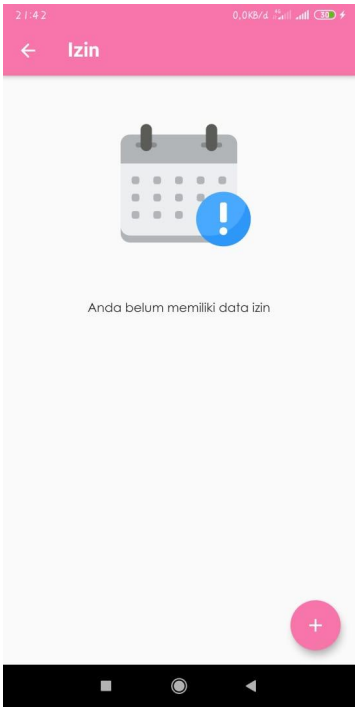
Tabel 4. 1 Pengujian *Black Box* Menu Login

	st Faktor	asil	terangan
	enu Absen Datang	✓	rhasil, karena ketika aplikasi ditekan tombol absen datang apabila kita berada dilokasi perusahaan maka akan tampil perintah absen datang, klik lanjut untuk memulai absen lalu tampil perintah harap lengkapi biometrik, lalu muncul proses loading dan berhasil mengabsen.
			

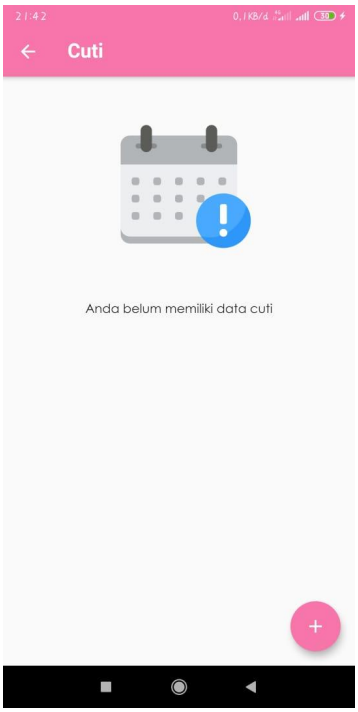
Tabel 4.2 Pengujian *Black Box* Menu Absen Datang

	st Faktor	asil	eterangan
	enu Absen Pulang	✓	rhasil, karena ketika aplikasi ditekan tombol absen pulang apabila kita berada dilokasi perusahaan dan telah memasuki jam pulang kerja maka akan tampil perintah absen pulang, klik lanjut untuk memulai absen, maka proses absen selesai.
			

Tabel 4.3 Pengujian *Black Box* Menu Absen Pulang

	st Faktor	asil	terangan
	Menu Izin	✓	rhasil, karena ketika aplikasi ditekan tombol izin lalu klik ikon tambah pada sudut kanan bawah maka kita diarahkan untuk mengisi kolom komentar lalu kirim, data yang dikirim akan sampai ke admin perusahaan.
			

Tabel 4.4 Pengujian *Black Box* Menu Izin

	st Faktor	asil	terangan
	Menu Cuti	✓	rhasil, karena ketika aplikasi ditekan tombol Cuti lalu klik ikon tambah pada sudut kanan bawah maka kita diarahkan untuk mengisi kolom komentar lalu kirim, data yang dikirim akan sampai ke admin perusahaan.
			

Tabel 4.5 Pengujian *Black Box* Menu Cuti

	st Faktor	asil	eterangan
	Menu Histori Absen	✓	rhasil, karena ketika aplikasi ditekan tombol Histori Absen kita dapat memilih bulan maupun tahun dari arsip absen karyawan dan akan tampil seluruh histori absen pada bulan dan tahun yang dipilih.



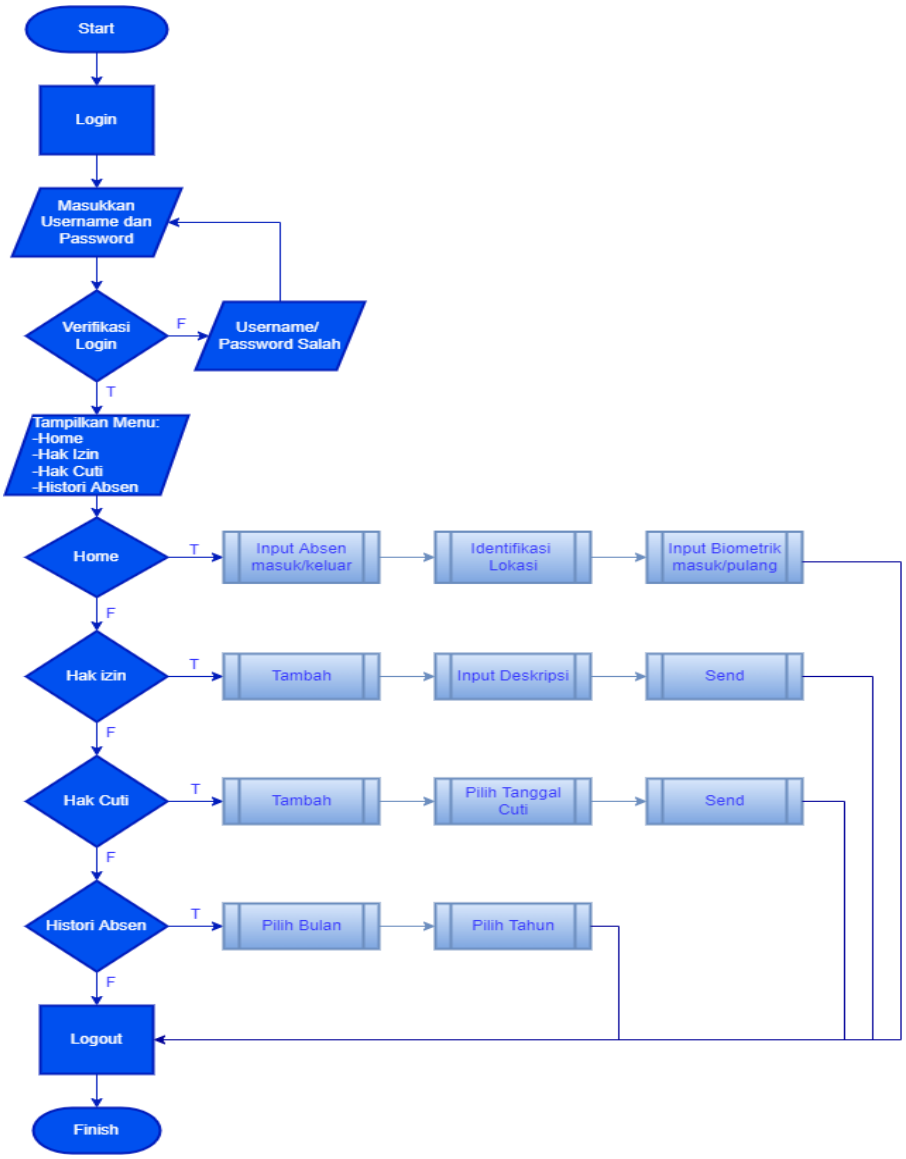
Tabel 4.6 Pengujian *Black Box* Menu Histori Absen

b) Pengujian *White Box*

Pengujian *White Box* merupakan pengujian yang dilakukan sampai detail pengecekan Logika program. Pengujian *White Box* berfokus pada efektifitas aplikasi yang dirancang.

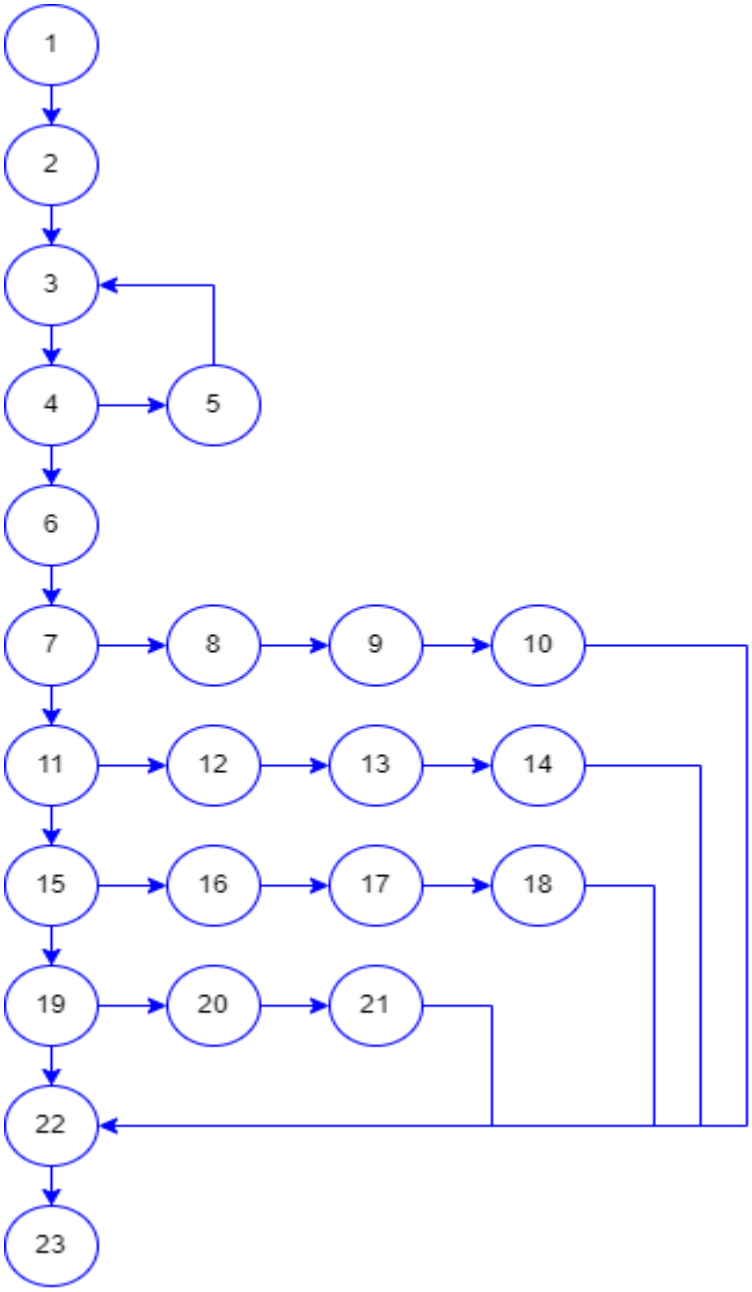
1) Aplikasi *Android*

a. *Flowchart* Aplikasi



Gambar 4.51 Flowchart Aplikasi

b. *FlowGraph* Aplikasi



Gambar 4.52 FlowGraph pada Aplikasi

Dari gambar *flowgraph* diatas dapat dilakukan *proses* perhitungan sebagai berikut:

(1) Mengitung *Cyclomatic Complexcity* V(G) dari *Edge* dan *Node* Dengan Rumus : V (G) = E – N + 2

N (*node*) = 23

E (*edge*) = 27

P(*Predikat node*)= 5

Penyelesaian : V(G) = E – N +2

= 27 – 23 + 2

= 6

Predikat = P + 1

= 5 + 1 = 6

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2

(3) Independent *Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3– 4 – 6– 7 – 11 – 15 - 19 – 22 - 23

Path 2 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 11 – 15 – 19 – 22 – 23

Path 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 22 – 23

Path 4 = 1 – 2 – 3 – 4 – 6 – 7 – 11 – 12 – 13 – 14 – 22 – 23

Path 5 = 1 – 2 – 3 – 4 – 6 – 7 – 11 – 15 – 16 – 17 – 18 – 22 – 23

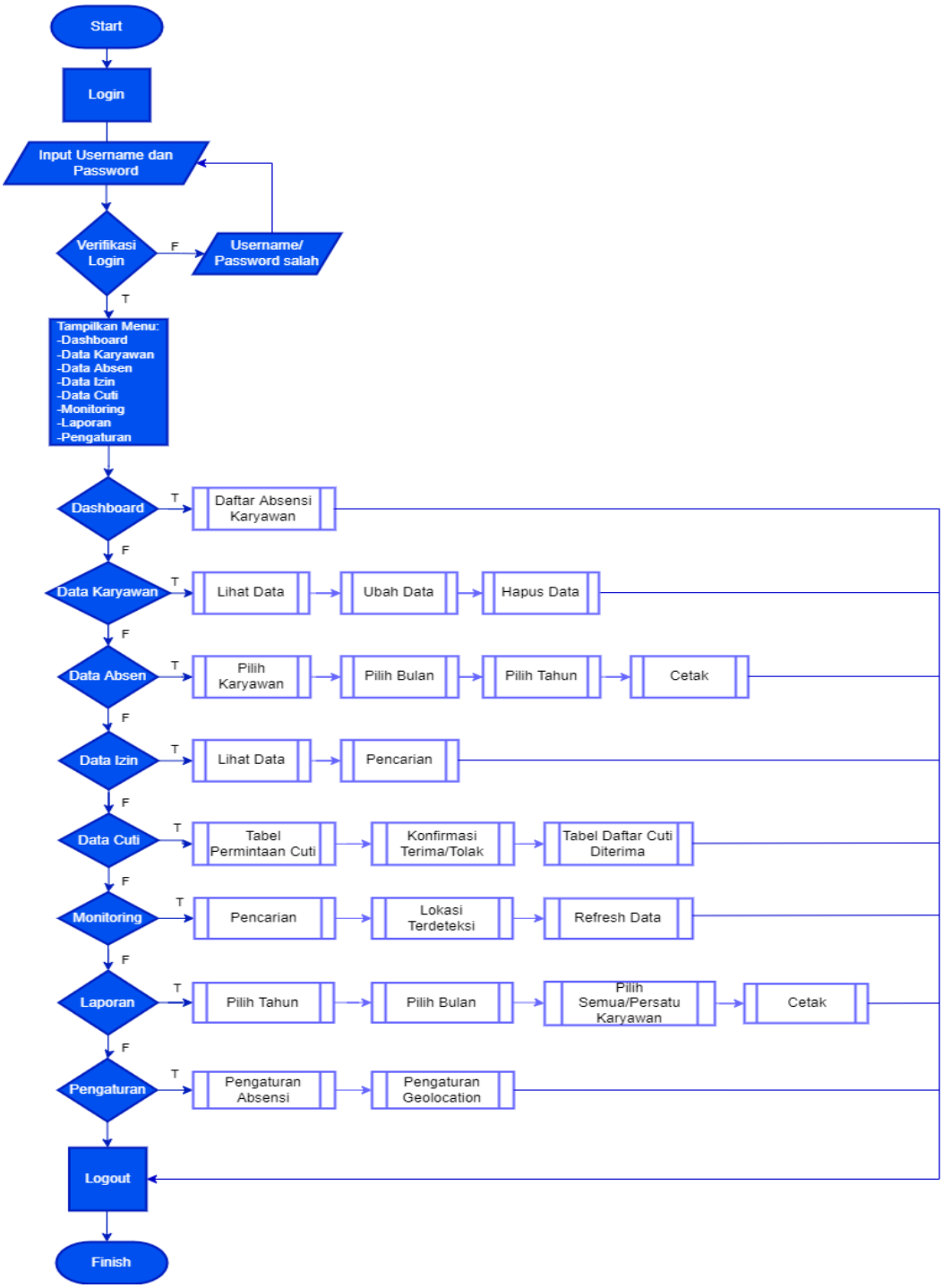
Path 6 = 1 – 2 – 3 – 4 – 6 – 7 – 11 – 15 – 19 – 20 – 21 – 22 – 23

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	E-1
1		1																						0
2			1																					0
3				1																				0
4					1	1																		1
5			1																					0
6							1																	0
7								1			1													1
8									1															0
9										1														0
10																						1		0
11												1			1									1
12													1											0
13														1										0
14																						1		0
15																1			1					1
16																	1							0
17																		1						0
18																						1		0
19																				1		1		1
20																					1			0
21																						1		0
22																							1	0
23																								
Sum=(E+1)																							5+1=6	

Gambar 4.53 Matriks pada Aplikasi

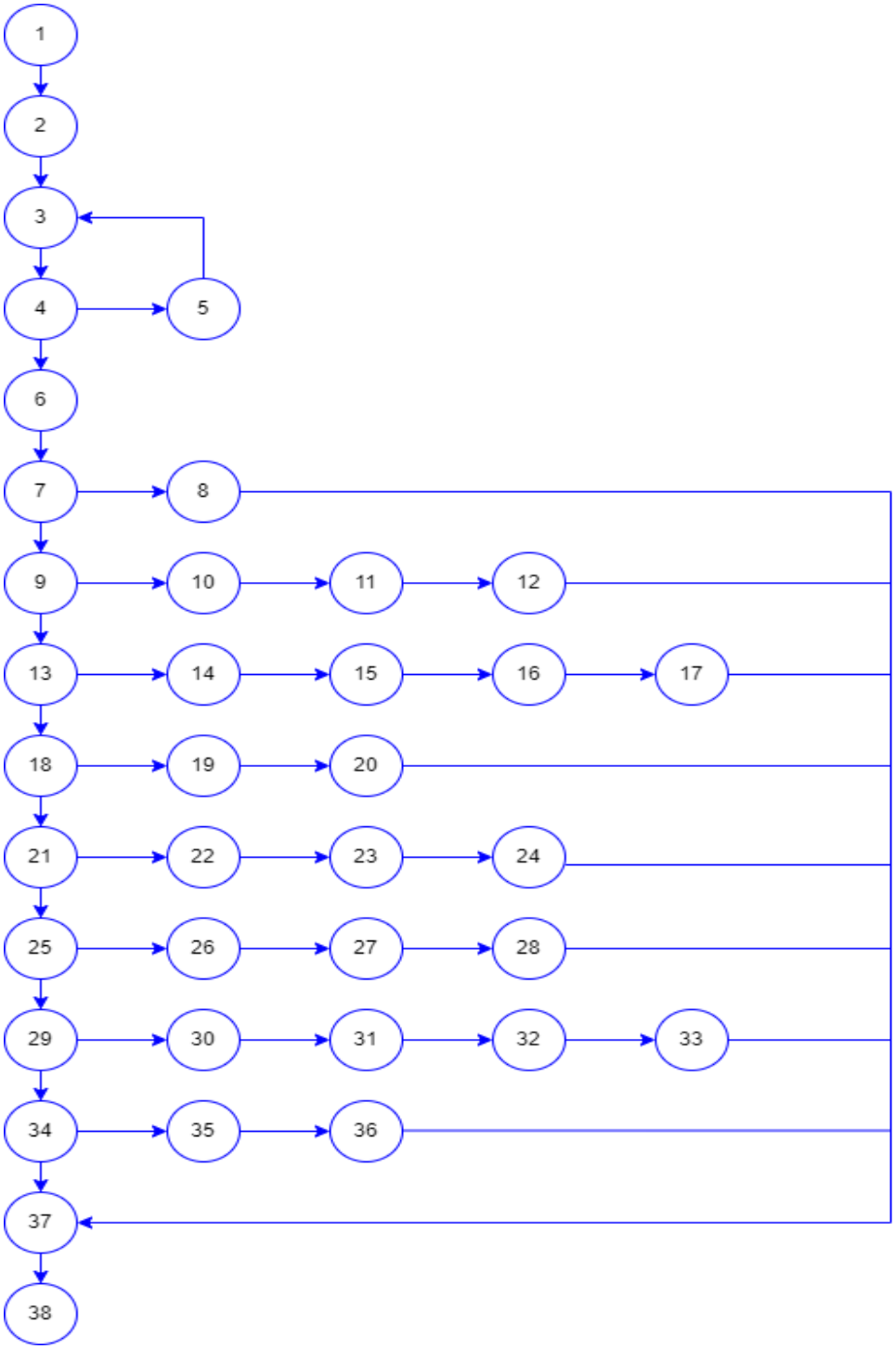
2) Halaman Web

a. *Flowchart* Web



Gambar 4.54 Flowchart Web

b. *Flowgraph* Web



Gambar 4.55 Flowgraph Web

Dari gambar *flowgraph* diatas dapat dilakukan *proses* perhitungan sebagai berikut:

(1) Mengitung *Cyclomatic Complexcity* V(G) dari *Edge* dan *Node* Dengan Rumus : V (G) = E – N + 2

N (*node*) = 38

E (*edge*) = 46

P(*Predikat node*)= 9

Penyelesaian : V(G) = E – N +2

= 46 – 38 + 2

= 10

Predikat = P + 1

= 9 + 1 = 1

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2

(3) Independent *Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3– 4 – 6 – 7 – 9 – 13 – 18 – 21 – 25 – 29 – 34 – 37 – 38

Path 2 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 9 – 13 – 18 – 21 – 25 – 29 – 34 – 37 – 38

Path 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 37 – 38

Path 4 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 12 – 37 – 38

Path 5 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 13 – 14 – 15 – 16 – 17 – 37 – 38

Path 6 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 13 – 18 – 19 – 20 – 37 – 38

Path 7 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 13 – 18 – 21 – 22 – 23 – 24 – 37 – 38

Path 8 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 13 – 18 – 21 – 25 – 26 – 27 – 28 – 37 – 38

Path 9 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 13 – 18 – 21 – 25 – 29 – 30 – 31 – 32 – 33 – 37 – 38

Path 10 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 13 – 18 – 21 – 25 – 29 – 34 – 35 – 36 – 37 – 38

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	E-1
1		1																																						0
2			1																																					0
3				1																																			0
4					1	1																																	1
5			1																																				0
6							1																																0
7								1	1																														1
8																																					1		0
9										1			1																										1
10											1																												0
11												1																											0
12																																					1		0
13														1				1																					1
14															1																									0
15																1																								0
16																	1																							0
17																																					1		0
18																			1		1																		1
19																				1																				0
20																																					1		0
21																						1			1														1
22																							1																0
23																								1															0
24																																					1		0
25																										1			1										1
26																											1												0
27																												1											0
28																																					1		0
29																													1					1					1
30																														1									0
31																															1								0
32																																1							0
33																																					1		0
34																																		1		1	1		1
35																																			1				0
36																																					1		0
37																																						1	0
38																																							
Sum=(E+1)																																							9+1=10

Gambar 4.56 Matriks pada Web

c) Pengujian Akurasi

Sistem pendeteksi lokasi pada aplikasi ini diuji secara manual, dengan cara pengguna/*user* berdiri di satu titik lokasi sebuah perusahaan, kemudian mengecek

latitude tempat *user* tersebut berdiri pada aplikasi absensi. Nilai latitude yang terderteksi oleh aplikasi absensi dari lokasi tempat user berdiri, selanjutnya di cek ke akuratannya di google Maps. Pengujian tersebut dilakukan secara berulang di beberapa titik lokasi dan dengan menggunakan beberapa jenis perangkat *android mobile* yang berbeda. Dari hasil pengujian tersebut didapatkan hasil 7 dari 10 jenis perangkat *android mobile* yang hasil pengujiannya akurat. Perangkat *android mobile* yang digunakan untuk pengujian dan hasil dari pengujian dapat dilihat pada tabel berikut:

Tabel 4.7 Spesifikasi Perangkat *Android Mobile*

Spesifikasi			Akurasi
Merk / Pemilik	**Processor**	**Ram**	
Vivo Y21 / Fina	2,3 GHz Octa-core	4,00+1,00 GB	Akurat
aomi Redmi 8 / Muhsyafir	2x2.0 GHz Octa-core	4,00 GB	Akurat
Redmi Note 9 /	MTK Helio G85 Octa-core	4,00 GB	Akurat
ppo A3S / Nyrma	8 Core 64-Bit	2,00 GB	Tidak Akurat (30M)
dmi Note 5 / Wiwi	napdragon 636, Octa-core	4,00 GB	Akurat
dmi Note 9 Pro / Akmal	Octa-core 2x2.3 GHz Kryo 465 Gold	8,00 GB	Tidak Akurat (10M)

ppo A12 /Akram	Octa-core 2.3GHz	4,00 GB	Akurat
Spesifikasi			**Akurasi**
Merk / Pemilik	**Processor**	**Ram**	
msung J1 Ace/ Tante Nani	Quad-core 1.2 GHz	768 MB	Tidak Akurat (800M)
vo Y20s (G)/ Bapak Fahri	2,0 GHz Octa-core	4,00 GB	Akurat
dmi Note 8 Pro / Fani	Mediatek Helio G90T (12nm)	6,00 GB	Akurat

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, Dihasilkannya sebuah Aplikasi Absensi Karyawan Menggunakan *Geolocation* dan *Finger Print* Berbasis *Android*. Aplikasi yang dibuat sudah diuji dengan menggunakan dua teknik pengujian yaitu pengujian *Black box* dan pengujian *White box* dan di uji Keakuratannya secara manual. Maka dapat ditarik kesimpulan sebagai berikut :

1. Didalam aplikasi ini terdapat fitur absen datang dan absen pulang yang dapat di akses ketika pengguna/ *user* berada dalam lokasi perusahaan yang telah ditentukan.
2. Aplikasi ini juga menyediakan fitur untuk memonitoring lokasi karyawan yang telah melakukan absen datang, sehingga perusahaan dapat mengetahui apabila karyawan meninggalkan lokasi perusahaan pada jam kerja tanpa konfirmasi. Fitur ini disertai keterangan berupa nama dan longitude latitude dari lokasi karyawan.
3. Aplikasi ini juga mempunyai fitur pengajuan cuti dan izin yang dapat di akses tidak hanya dalam lokasi perusahaan melainkan dapat di akses di seluruh area.
4. Proses absensi yang diterapkan dalam aplikasi ini adalah menggunakan *system geolocation* dan *finger print* agar tingkat kesalahan dalam melakukan absensi dapat diminimalisir dan memberikan pelayanan lebih baik.

5. Aplikasi absensi ini dapat memberikan kemudahan dalam melakukan pendataan kehadiran karena penggunaannya yang bersifat *user friendly*.
6. Aplikasi absensi menggunakan *Firebase Realtime* sebagai pengolahan datanya yang dirancang dengan bahasa pemrograman *JavaScript* dan *NodeJS* pada *Android Flutter*, serta membutuhkan koneksi internet pada saat menjalankannya.
7. Keakuratan pendeteksi lokasi pada aplikasi ini di uji secara manual dengan hasil 7 dari 10 perangkat *Android Mobile* mendapatkan hasil akurat tinggi dan 3 lainnya mendapatkan hasil keakuratan yang lebih rendah. Disimpulkan bahwa semakin tinggi spesifikasi suatu perangkat *android mobile*, semakin tinggi pula kemungkinan tingkat akurasi yang di hasilkan.

B. Saran-saran

Aplikasi Absensi Karyawan Menggunakan *Geolocation* dan *Finger Print* Berbasis *Android* adalah aplikasi absensi berbasis *android* pertama yang dirancang di PT. Bosowa Berlian Motor Parepare. Oleh karena itu diperlukan pengembangan kearah yang lebih baik untuk meningkatkan keefektifan penggunaan aplikasi. Adapun saran saran terhadap pengembangan aplikasi adalah sebagai berikut.

1. Untuk pengembangan selanjutnya dapat ditambah misalnya dibuat fitur chat pada aplikasi agar *admin* dapat berkomunikasi antar pengguna.
2. Keakuratan pendeteksi lokasi juga dapat diberi perhatian lebih agar semakin meningkatnya tingkat akurasi di berbagai jenis perangkat *Android.*

3. Aplikasi ini hanya mendeteksi GPS Handphone karyawan saja, sehingga masih memungkikan adanya ide-ide kecurangan. Untuk pengembang selanjutnya agar memberi perhatian lebih untuk contoh masalah tersebut.
4. Hendaknya sistem yang penulis rancang dapat dikembangkan sehingga dalam pengolahan datanya dapat lebih dijelaskan secara mendetail, agar pengolahan data dan pembuatan laporan dapat dilakukan lebih efektif dan efesien.
5. Sistem aplikasi yang telah dikembangkan penulis sebaiknya dipelihara dengan sebaik-baiknya dan di perbaharui sesuai dengan kebutuhan yang ada serta melakukan *backup* data pada jangka waktu tertentu untuk menghindari kemungkinan data hilang atau rusak.

Demikian saran yang dapat penulis berikan, semoga saran tersebut bisa dijadikan sebagai bahan masukan yang dapat bermanfaat bagi penulis khususnya dan bagi pengembang pada umumnya.

DAFTAR PUSTAKA

Arifianto, Teguh. 2011. *Membuat Interface Aplikasi Android Lebih Keren dengan LWUIT*. Yogyakarta: Andi Publisher.

Adi Nugroho. 2010. *Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP*. Andi. Yogyakarta

Adi Nugroho, 2005, *Konsep Pengembangan Sistem Basis Data Informatika,*

Diana, (2017). *"Perancangan Sistem Informasi Absensi Guru Dan Siswa Berbasis Web Di Sweet School Batam"*. Skripsi : Sekolah Tinggi Ilmu Komputer (STMIK) Gici, Batam.

Dhani Wibawa, (2016). *"Aplikasi Presensi Kepegawaian Kelurahan Menggunakan Teknologi Fingerprint (Studi kasus di Kantor Kelurahan Sindumartani Ngemplak Sleman Yogyakarta)"*. Skripsi : STMIK Akakom Yogyakarta.

Deka M Wildan, 2014. *"Tensorflow". Avaible at: https://www.tensorflow.org/ [Accessed 29 Oktober 2021]*.

Ely Mulyadi, Andis Trihariprasetya, I Gede Wiryawan, (2020). "*Penerapan Sistem Presensi Mobile dengan Menggunakan Sensor GPS (Klinik Pratama X di Jember)".* Skripsi : Universitas Pendidikan Ganesha, Bali.

Hermawan S, Stephanus. 2011. *"Mudah Membuat Aplikasi Android".* Yogyakarta : Andi Offset.

Hartono, Jogiyanto. 1999. *Pengenalan Komputer*. Andi:Yogyakarta.

Isaca, 2016. Chelsey Juarez, Geolocation: A Pathway to Identification for Deceased Undocumented Border Crossers.: University of California. Santa Cruz.

Jogiyanto Hartono, 2005. *Analisis dan Desain Sistem Informasi. Pendekatan terstruktur teori dan praktis aplikasi bisnis*. Andi. Yogyakarta.

KBBI, 2015. Kamus Besar Bahasa Indonesia (KBBI). [Online] Available at: http://kbbi.web.id/di [Diakses 21 Juni 2020].

Kadir, A. 2008. *Dasar Pemograman dan Implementasi Database Relational*.

Lardinois, F. (2015). Microsoft Launches Visual Studio Code, a Free CrossPlatform Code Editor for OS X, Linux and Windows. [Diakses pada tanggal 02 Oktober 2021]

Madcoms. 2012. *Manajemen Sistem Jaringan Komputer.Ed I*. Yogyakarta: Andi, Madiun: Madcoms.

Madcoms. 2012. Aplikasi Chatting menggunakan Laravel dan Vue js. [Diperoleh 10 juni 2021]

Munawar. (2005), *Pemodelan Visual dengan UML*, Graha Ilmu,

Yogyakarta, 17-100. Nazruddin Safaat H, 2011, *Android (Pemograman*

Aplikasi Mobile Smartphone dan

Tablet PC Berbasis Android). Informatika, Bandung

Nazruddin, Safaat H. 2012. (Edisi Revisi) *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika, Bandung.

Putri Dwi Intan Ra'uufi, dan Yanuar Sinatra. 2016. *Rancangan Bangun Aplikasi Delivery Order Berbasis Android Di Ageng Bakery Group Bliatar*.

Pressman, Roger, S. 2012. *Rekayasa Perangkat Lunak.Pendekatan Praktisi*. Edisi 7.

Purnomosidi D.P, 2013, Buku Cloud Node.js, https://github.com/bpdp/buku-cloud- nodejs. [Diakses pada tanggal 02 Oktober 2021]

Pengertian Finger Print dan fungsinya. 20 November 2019, https://www.kiosbarcode.com/blog/pengertian-fingerprint-dan-fungsinya/.

Setiawan dan Kurniawan. (2015). *Perancangan Sistem Absensi Kehadiran Perkuliahan dengan Menggunakan Radio Frequency Identification* (RFId).

Sholiq. 2006. *Pemodelan Sistem Informasi Berorientasi Objek dengan UML.Graha Ilmu*. Yogyakarta.

Sugiyono. 2001. *Metode Penelitian kuantitatif kualitatif dan R&D*. Bandung: Alfabeta.

Universitas Muhammadiyah Parepare. (2021). *Panduan Akademik 2020-2021*, Parepare : UMPAR Press.

Vivah. (2017). THE BEGINNER'S GUIDE: Understanding Node.js & Express.js fundamentals.Retrieved from Medium: https://medium.com/@LindaVivah/the- beginners-guide-understandingnode-js-express-js. [Diakses pada tanggal 02 Oktober 2021]

Widyatmoko, Nur, 2012. "Pemanfaatan Geolocation dan Haversine Formula dalam Perancangan Sistem Informasi Geografis (GIS) (Studi Kasus : Pariwisata Kabupaten Semarang)". Repository Digital Library UKSW.

Wikipedia, 23 Desember 2009, *UML dan Rational Rose*, *http://wikipedia.com*