

# PENGEMBANGAN APLIKASI MOBILE PRESENSI PEGAWAI BERBASIS GPS DAN GEOCODING MENGGUNAKAN METODE PENGUKURAN JARAK HAVERSINE

## *DEVELOPMENT OF EMPLOYEE PRESENCE MOBILE APPLICATION BASED ON GPS AND GEOCODING USING HAVERSINE DISTANCE MEASUREMENT METHOD*

Guna Dermawan<sup>1</sup>

<sup>1,2,3</sup>Teknik Informatika/Fakultas Ilmu komputer/Universitas Dian Nuswantoro  
E-mail: <sup>1</sup>11201811538@mhs.dinus.ac.id

### **Abstrak**

Presensi adalah sebuah tanda kehadiran seseorang yang menandakan bahwa orang tersebut telah melakukan presensi dimana presensi itu nantinya dapat di jadikan sebuah laporan ataupun pengambilan sebuah kebijakan baik di instansi Pendidikan, perusahaan ataupun instansi jenis lainnya. Ada berbagai jenis presensi yang lazim di temukan mulai dari yang menggunakan sistem konvensional ataupun yang berbasis komputerisasi. Di dalam instansi pemerintahan presensi berperan penting sebagai laporan kinerja pegawai dan sebagai parameter dalam pengambilan kebijakan pemimpin kepada bawahannya, tetapi masih banyak ditemukan instansi yang melakukan presensi pegawainya menggunakan cara konvensional dengan cara menuliskannya secara manual pada lembaran kertas, hal ini dapat menimbulkan masalah dikemudian hari jika ada kerusakan media kertas yang dijadikannya sebagai media presensi, oleh sebab itu diperlukan sebuah sistem penanganan yang menyediakan kemudahan dan kenyamanan saat melakukan presensi yaitu dengan merancang aplikasi presensi berbasis android dengan memanfaatkan GPS atau global positioning system dan geocoding sebagai alat untuk membantu melakukan presensi. GPS adalah sistem navigasi berbasis satelit sedangkan geocoding adalah mekanisme untuk melakukan translasi bahasa manusia dan akan diterjemahkan menjadi sebuah koordinat peta yaitu berupa latitude dan longitude. Sedangkan untuk membatasi jarak lokasi presensi dan ponsel pengguna, maka digunakan metode haversine sebagai algoritma utama dalam melakukan kalkulasi jarak secara akurat.

Kata kunci: android, GPS, geocoding, presensi mobile, haversine

### **Abstract**

*Presence is a sign of someone's presence which indicates that the person has made a presence where the presence can later be made into a report or making a policy either in educational institutions, companies or other types of institutions. There are various types of attendance that are commonly found, ranging from those using conventional systems or computer-based ones. In government agencies the presence plays an important role as an employee performance report and as a parameter in making policy decisions for leaders to their subordinates, but there are still many agencies that carry out employee attendance using conventional methods by writing them down manually on sheets of paper, this can cause problems in the future if there is damage to the paper media which is used as*

*presence media, therefore we need a handling system that provides convenience and comfort when doing attendance, namely by designing an android-based presence application by utilizing GPS or global positioning system and geocoding as a tool to help make attendance. GPS is a satellite-based navigation system while geocoding is a mechanism for translating human language and will be translated into a map coordinate in the form of latitude and longitude. Meanwhile, to limit the distance between the presence and the user's cell phone, the haversine method is used as the main algorithm in calculating the distance accurately.*

*Keywords: android, GPS, geocoding, mobile presence, haversine*

## 1. PENDAHULUAN

Laporan presensi merupakan hal yang sangat penting dalam berbagai instansi. presensi sebagai penilaian kinerja dan kedisiplinan pegawai ataupun karyawannya[1][2], begitu juga presensi yang ada di kantor balai desa Warureja kabupaten Tegal ini, presensi yang di gunakan masih menggunakan cara konvensional, yaitu dengan menggunakan buku presensi di mana setiap pegawai akan melakukan presensi dengan menuliskannya secara manual yang nantinya akan di rekap oleh bagian administrasi pegawai, namun dengan presensi yang menggunakan cara konvensional ini menimbulkan berbagai problematika, diantaranya adalah manipulasi presensi dimana setiap pegawai dapat melakukan presensi pada jam yang tidak seharusnya, kemungkinan presensi palsu juga sangat mungkin terjadi, kedua hal di atas tentunya menjadi hal yang krusial mengingat presensi dapat dijadikan pengambilan kebijakan oleh atasan dalam hal ini adalah kepala desa, untuk itu perlu di buat sistem presensi yang dapat mengatasi problematika diatas yaitu dengan membuat sistem presensi menggunakan GPS dan Geocoding yang berbasis android untuk memastikan akurasi dan kedisiplinan para pegawai. Dalam penelitian ini, setiap presensi pegawai akan dapat dilakukan jika perangkat pegawai tersebut berada pada jarak tertentu dari kantor, jika tidak maka pegawai tersebut tidak dapat melakukan presensi. Data presensi yang disediakan sistem ini lengkap dengan tanggal dan jam pada saat pegawai melakukan presensi.

Problematika presensi bukanlah hal yang baru, telah banyak penelitian yang mengungkap tentang permasalahan presensi, khususnya penerapannya pada perangkat android untuk dapat dijadikan alternatif alat presensi diantaranya adalah penelitian yang dilakukan oleh J. Christian dan H. Nasrullah, dengan memanfaatkan RFID atau *Radio Frequenc Identification* menghasilkan sebuah mesin pembaca kartu untuk presensi. Namun terdapat beberapa kelemahan dalam pengembangan sistem presensi, diantaranya adalah masalah pada validasi pemilik identitas asli dari kartu tersebut serta karena sistem ini menggunakan koneksi *server-client*, maka jika terjadi putus jaringan secara otomatis akan melumpuhkan kinerja dari sistem presensi ini.[3]

Penelitian mengenai presensi berbasis mobile juga pernah dilakukan oleh D. Supriatna dan E. Junianto, dimana dalam penelitian ini menghasilkan aplikasi yang dapat melakukan presensi dengan dua metode, yaitu menggunakan GPS serta menggunakan *fingerprint*, namun terdapat beberapa kelemahan dalam penelitian ini, diantaranya adalah data yang didapatkan pada saat pengguna melakukan presensi menggunakan *fingerprint* data yang dihasilkan belum dapat tersimpan kedalam *database* aplikasi [4]. Penelitian lain juga pernah diimplementasikan pada bidang Kesehatan yang dilakukan oleh E. Mulyadi, A. Trihariprasetya, dan I. G. Wiryawan dimana dihasilkan sistem presensi berbasis mobile yang dapat terintegrasi dengan sistem informasi yang telah tersedia, selain itu sistem presensi berbasis mobile ini juga dapat bekerja 24 jam untuk memenuhi kebutuhan pegawai dalam proses *shifting* sesuai dengan jadwal yang tersedia [5].

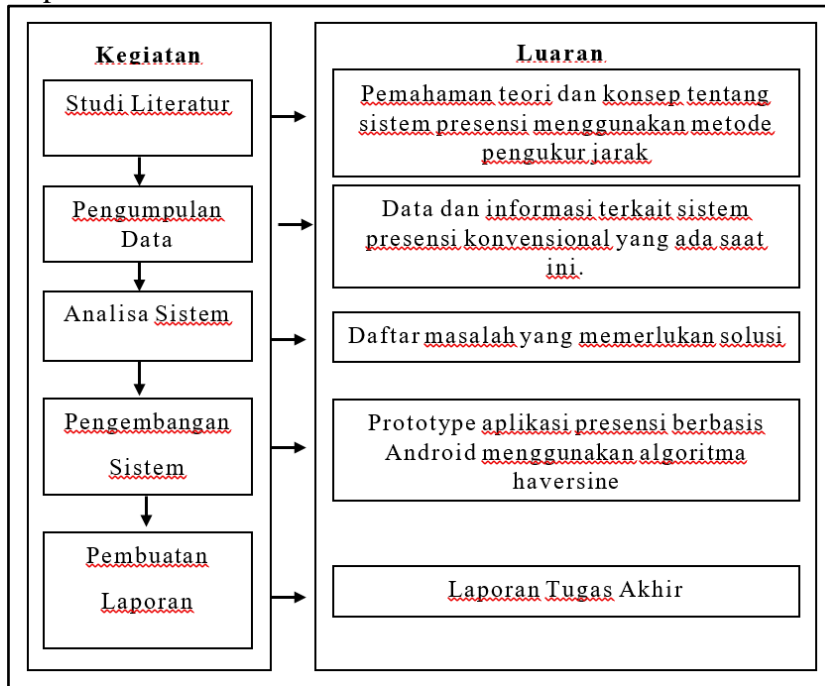
Berdasarkan uraian penelitian diatas, terdapat beberapa penelitian yang telah menggunakan *smartphone* sebagai alat presensi, diantaranya menggunakan GPS untuk melakukan deteksi lokasi serta *fingerprint* sebagai alternatif dalam melakukan presensi, namun ketika menggunakan GPS secara objek tunggal pengambilan posisi, potensi akurasi yang kurang tepat atau tidak akurat[6], oleh karena itu dalam penelitian ini digunakan bantuan geocoder yang telah tertanam pada sistem android dimana geocoder itu akan mentranslasikan koordinat berupa latitude dan longitude menjadi sebuah alamat yang dapat dibaca serta dapat diketahui lokasinya[2], geocoder ini akan digunakan untuk mengambil koordinat kantor yang akan disimpan dalam variabel untuk dilakukan pengukuran jarak dengan lokasi terkini dari *smartphone* pengguna. Untuk mengatasi kecurangan yang mungkin terjadi, maka perlu dilakukan pembatasan area atau pengukuran jarak untuk melakukan presensi[7], metode yang akan digunakan untuk melakukan pengukuran jarak antara kantor dengan lokasi pengguna adalah metode haversine. Haversine adalah metode pengukuran yang menghitung jarak antara titik lokasi dipermukaan bumi dengan menggunakan garis lintang atau longitude dan garis bujur atau latitude sebagai nilai dari inputan variabel [8]. Dengan adanya aplikasi ini diharapkan pegawai dapat dengan mudah dalam melakukan presensi pada setiap hari kerja, juga pihak kantor akan dengan mudah membuat sebuah kebijakan terkait dengan kinerja pegawai dari perspektif presensi karena pegawai melakukan presensi secara terjadwal setiap jam kerja yang telah ditetapkan.

## 2. METODOLOGI PENELITIAN

Sebelum membuat sistem presensi, perlu diadakan pemodelan tentang sistem atau aplikasi yang akan dikembangkan, detail dari pemodelan akan

*Table 1. Tabel Metodologi penelitian*

dijelaskan pada tabel 3 berikut ini:



### 2.1 Studi Literatur

Dalam mengembangkan sistem presensi, perlu diimplementasikan kalkulasi yang akurat dalam penentuan jarak dan lokasi yang akan digunakan untuk melakukan presensi, untuk itu, dalam penelitian ini, digunakan metode pengukuran jarak haversine, metode haversine merupakan metode pengukuran jarak yang paling akurat jika dikomparasikan dengan metode pengukuran jarak lain, seperti manhatan dan Euclidean[9]. Dengan demikian diharapkan aplikasi presensi yang akan dibuat dapat secara positif membantu pengguna yang dalam hal ini adalah pegawai di balai desa Warureja serta dapat digunakan sebagai bahan acuan mengenai kedisiplinan pegawai dalam melakukan presensi.

### 2.2 Pengumpulan Data

Metode pengumpulan data menggunakan beberapa sumber data, diantaranya adalah sebagai berikut:

#### 2.2.1 Observasi

Pengumpulan data dengan cara observasi dilakukan dengan cara pengamatan langsung di lapangan. Dengan begitu, pemahaman kebutuhan antara subjek dan objek menjadi jelas dan mengerucut pada satu bidang penelitian. Dalam

penelitian ini, digunakan observasi dengan cara wawancara kepada pihak terkait dalam hal ini adalah sekretariat desa Warureja Kabupaten Tegal, melakukan pengamatan bangunan instansi untuk menentukan jarak pengukuran presensi yang tepat serta mencoba beberapa aplikasi presensi yang telah tersedia secara publik.

### **2.2.2 Studi Pustaka**

Studi Pustaka merupakan langkah awal yang digunakan untuk melakukan komparasi terhadap penelitian terkait yang telah ada sebelumnya, hasil dari komparasi ini adalah data perbandingan yang dapat digunakan untuk melakukan inovasi terhadap penelitian yang sedang dilakukan, untuk mendapatkan studi pustaka yang baik, ada beberapa sumber yang bisa dijadikan acuan mencari sumber penelitian, seperti dari jurnal penelitian, paper, internet dan lain sebagainya. Adapun pembahasan detail terkait studi pustaka terdapat di bab 2.

### **2.3 Analisa Sistem**

Problematisa presensi bukanlah hal baru di dalam suatu instansi, mengingat presensi dapat dijadikan bahan pengambilan keputusan kepala kantor bagi karyawannya ataupun sebagai tolak ukur kedisiplinan pegawai dalam menghadiri jam kantor sesuai dengan waktu yang ditentukan.

Hal diatas juga terjadi di kantor kepala desa Warureja Kabupaten Tegal, presensi para pegawai di kantor tersebut masih menggunakan cara konvensional, yaitu dengan menuliskannya pada buku presensi, kemudian akan dilakukan rekapitulasi oleh bagian administrasi pegawai, namun dengan menggunakan cara konvensional, aksi kecurangan dalam presensi masih bisa dilakukan, misalnya dengan menitipkan presensi pada teman ataupun menuliskannya pada jam yang tidak sesuai.

Oleh karena itu, perlu dibuat sistem presensi yang dapat mengatasi masalah yang telah disebutkan, yaitu dengan membuat aplikasi presensi berbasis android dengan menggunakan fitur geocoder dan algoritma haversine untuk mendapatkan akurasi yang tepat pada saat pegawai melakukan presensi, aplikasi ini akan menyediakan fitur presensi dimana setiap pegawai diharuskan membuat akun terlebih dahulu sebelum bisa melakukan presensi, aplikasi akan membaca waktu dan jarak tertentu dari perangkat yang digunakan untuk presensi, jika waktu dan jarak perangkat memenuhi persyaratan, maka presensi dapat dilakukan, dan aplikasi akan mencatatkan tanggal serta jam presensi pada tiap pegawai, namun jika dua kondisi diatas tidak terpenuhi, maka presensi akan ditolak oleh sistem.

#### **2.3.1 Analisa Kebutuhan Fungsional**

1. Aplikasi dapat melakukan aksi login sesuai dengan data pengguna yang tersimpan di dalam basis data.

2. Aplikasi dapat menambah pengguna baru atau *signup* sesuai dengan spesifikasi yang telah ditentukan.
3. Aplikasi dapat melakukan reset password pengguna dengan menggunakan email yang telah terdaftar di dalam basis data.
4. Aplikasi dapat melakukan proses presensi sesuai dengan spesifikasi yang telah ditentukan.
5. Aplikasi dapat menampilkan data hasil presensi yang telah tersimpan ke dalam basis data dalam bentuk list.
6. Aplikasi dapat menampilkan profile lengkap pengguna secara lengkap sesuai dengan spesifikasi yang telah ditentukan.
7. Aplikasi dapat mengganti profile, nama pengguna, melakukan verifikasi email, merubah email dan password pengguna sesuai dengan spesifikasi yang telah ditentukan.
8. Aplikasi dapat menyediakan fitur lapor *bug* dari pengguna.
9. Aplikasi dapat menampilkan detail tentang aplikasi kepada pengguna.
10. Aplikasi dapat melakukan logout pada tiap akun pengguna yang telah login.

### **2.3.2 Analisa Kebutuhan Non-fungsional**

1. Aplikasi memiliki antar muka atau *user interface* yang mudah dipahami oleh pengguna.
2. Aplikasi menyediakan hak akses pada tiap penggunanya.
3. Aplikasi dapat berjalan dengan baik di sistem operasi Android

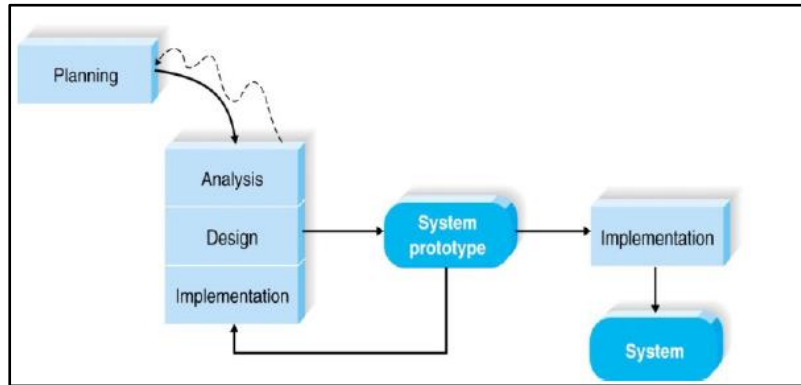
## **2.4 Pengembangan Sistem**

Beberapa perangkat pendukung dalam mengembangkan aplikasi ini adalah sebagai berikut:

### **2.4.1 Metode Pengembangan Sistem Prototype**

Pada penelitian ini, digunakan metode *prototype* sebagai penyelesaian tahapan pengembangan sistem, dimana dalam metode ini akan melibatkan pengguna secara *real time* dalam proses pengembangan perangkat lunak, feedback yang berasal dari pengguna akan digunakan dalam pengembangan aplikasi[10], detail metode prototyping adalah sebagai berikut:

Berikut adalah tahapan metode pengembangan perangkat lunak dengan menggunakan metode prototype:



*Gambar 1. metode prototype*

a. Planning

Pada tahapan ini, pengguna dalam hal ini adalah pegawai balai desa Warureja dan programmer aplikasi akan berdiskusi untuk menentukan kebutuhan perangkat lunak secara garis besar bagaimana sistem atau aplikasi akan dibuat, selain itu, dilakukan pula pengkajian jurnal ataupun penelitian yang berkorelasi dengan sistem aplikasi presensi berbasis android yang akan dikembangkan, seperti jenis algoritma untuk diimplementasikan dalam pembuatan aplikasi. Untuk mencapai hasil yang maksimal, ditentukan juga waktu pengembangan aplikasi beserta aktivitas dan gambaran hasil akhir dari aplikasi presensi yang akan dikembangkan di balai desa Warureja Kabupaten Tegal.

b. Analysis

Tahapan analisa akan dilakukan identifikasi kebutuhan perangkat lunak dan perangkat keras dalam pengembangan aplikasi. Analisa kebutuhan meliputi kebutuhan perangkat lunak dan perangkat keras yang akan digunakan dalam pengembangan aplikasi presensi menggunakan algoritma haversine berbasis android.

Sedangkan untuk pemenuhan kebutuhan algoritma pendukung, telah dilakukan studi untuk menetapkan algoritma yang tepat dalam pengembangan aplikasi presensi, khususnya dalam akurasi pengukuran jarak, dan algoritma yang akan digunakan adalah algoritma haversine, algoritma ini dipilih karena menghasilkan akurasi, waktu pemrosesan dan ketepatan aplikasi paling baik diantara algoritma [9].

c. Design

Tahapan desain adalah tahapan iterasi antara developer dan pengguna, pada tahapan ini, desain dari UI (*user interface*), basis data aplikasi akan dibuat, desain dari aplikasi presensi di desa Warureja adalah berupa *wireframe* termasuk diagram pendukungnya seperti ERD, activity diagram, class diagram serta sequence

diagram. Database yang akan digunakan adalah *database* firebase, meliputi autentikasi untuk menyimpan informasi mengenai pengguna dan *storage database* yang digunakan untuk menyimpan foto profile pengguna dan *realtime database* untuk menyimpan histori presensi pengguna. Tahap desain akan dilakukan iterasi antara pengguna dan developer untuk mendapatkan desain aplikasi yang sesuai dengan kebutuhan.

d. Implementation

Pembuatan prototype dilakukan berdasarkan data yang didapatkan dari tahap desain, Pada tahapan ini, programmer akan mulai melakukan pembuatan aplikasi menggunakan android studio dan bahasa pemrograman kotlin sebagai bahasa pendukungnya. Bagian aplikasi yang dibuat meliputi tampilan *splash screen*, login dan signup, fitur lupa password, bagian aplikasi ini akan pertama muncul ketika pengguna membuka aplikasi, terdapat inputan berupa email dan password, ketika pengguna telah memiliki akun, maka akan bisa melakukan login kedalam menu utama aplikasi, namun ketika tidak memiliki akun, pengguna bisa membuatnya pada menu signup dengan memasukkan email dan password yang valid sesuai dengan ketentuan aplikasi, selanjutnya, ketika pengguna lupa password akun yang telah dibuat, maka pengguna bisa menggunakan fitur lupa password dan aplikasi akan mengirimkan tautan untuk melakukan reset password secara mandiri, sedangkan bagian utama dari aplikasi presensi ini meliputi menu beranda, presensi, histori presensi dan menu profil pengguna, di menu beranda, terdapat beberapa tautan terkait seperti lapor gubernut, lapor bupati informasi Covid19 dan beberapa tautan lainnya, pada menu presensi, terdapat tombol untuk aplikasi melakukan kalkulasi jarak dan beberapa kondisi yang memungkinkan pengguna untuk presensi di area yang telah diijinkan, sedangkan pada menu histori, terdapat list user yang telah melakukan presensi dilengkapi tanggal dan waktu presensi, dan pada menu profile terdapat pengaturan untuk konfigurasi akun pengguna, seperti ganti foto profil, username, email, verifikasi email, lapor bug, info aplikasi dan tombol logout untuk keluar dari aplikasi. Tahapan implementasi akan dilakukan iterasi sesuai dengan kebutuhan pengguna.

e. System prototype

Tahapan prototyping disini akan dilanjutkan dengan melakukan pengujian pada perangkat lunak atau aplikasi yang telah disesuaikan dengan kebutuhan pengguna, ada dua jenis pengujian yang akan dilakukan, yaitu *black box testing* dan *white box testing*, untuk memastikan kepuasan pengguna yang dalam hal ini adalah pegawai balai desa Warureja, juga dilakukan kuesioner tentang aplikasi yang telah dikembangkan oleh pegawai balai desa Warureja Kabupaten Tegal.

## 2.5 Pembuatan Laporan

Sebagai tahapan akhir dari penelitian ini, maka akan dibuat laporan penelitian berupa laporan berupa skripsi sebagai tugas akhir di program studi



Teknik informatika. Laporan ini juga akan dipublikasikan sebagai bahan acuan pada penelitian berikutnya serta sebagai bahan pertimbangan untuk pembuatan aplikasi presensi yang lebih kompleks.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Rancangan Sistem

##### 3.1.1 Planing

Pada tahapan ini, pengguna dan programmer aplikasi akan berdiskusi untuk menentukan kebutuhan perangkat lunak secara garis besar bagaimana sistem atau aplikasi akan dibuat, dalam pengumpulan kebutuhan perangkat lunak, akan dikumpulkan data berupa hasil wawancara dengan klien, referensi ilmiah serta beberapa sampel aplikasi yang telah ada sebelumnya sebagai bahan acuan dalam pengembangan perangkat lunak.

Struktur rincian kerja pada pembuatan aplikasi presensi menggunakan algoritma haversine pada android adalah sebagai berikut:

*Table 2. Tabel WBS (Work Breakdown Structure)*

WBS	TUGAS PROYEK	DURASI	TANGGAL MULAI	TANGGAL SELESAI
<b>1.0</b>	<b>Planning dan Analisa proyek</b>	<b>2 hari</b>	<b>22 November 2021</b>	<b>23 November 2021</b>
1.1	Perumusan masalah	1 Hari	22 November 2021	23 November 2021
1.2	Memahami kinerja sistem	1 Hari	22 November 2021	23 November 2021
1.3	Analisa hasil	1 Hari	23 November 2021	23 November 2021
1.4	Pengumpulan referensi	1 Hari	23 November 2021	23 November 2021
<b>2.0</b>	<b>UI Desain</b>	<b>3 Hari</b>	<b>24 November 2021</b>	<b>26 November 2021</b>
2.1	Pembuatan wireframe aplikasi	1 Hari	24 November 2021	24 November 2021
2.2	Implementasi ui di Android studio	2 Hari	25 November 2021	26 November 2021
<b>3.0</b>	<b>Koding / Penulisan kode aplikasi</b>	<b>15 Hari</b>	<b>27 November 2021</b>	<b>11 Desember 2021</b>
3.1	Konfigurasi basis data	2 Hari	27 November 2021	28 November 2021

3.2	Membuat fungsi login dan sign up aplikasi	4 Hari	29 November 2021	2 Desember 2021
3.3	Manajemen user profile	3 Hari	3 Desember 2021	5 Desember 2021
3.4	Membuat fungsi presentasi	6 Hari	6 Desember 2021	11 Desember 2021
<b>4.0</b>	<b>Testing dan implementasi</b>	<b>3 Hari</b>	<b>12 Desember 2021</b>	<b>14 Desember 2021</b>
4.1	Debugging	2 Hari	12 Desember 2021	13 Desember 2021
4.2	Implementasi	1 Hari	14 Desember 2021	15 Desember 2021
<b>5.0</b>	<b>Pemeliharaan</b>	<b>2 Hari</b>	<b>16 Desember 2021</b>	<b>17 Desember 2021</b>

### 3.1.2 Analysis

Tahapan analisa akan dilakukan identifikasi kebutuhan perangkat lunak dan perangkat keras dalam pengembangan aplikasi.

Kebutuhan perangkat keras atau hardware dalam pengembangan aplikasi presensi berbasis android ini adalah perangkat laptop dengan menggunakan spesifikasi sebagai berikut:

1. Processor INTEL CORE I5 10300H 10th
2. RAM sebesar 16GB DDR4
3. VGA NVIDIA GForce 1650 4GB
4. SSD NVme 500GB

Kebutuhan perangkat lunak atau software dalam pengembangan aplikasi ini adalah sebagai berikut:

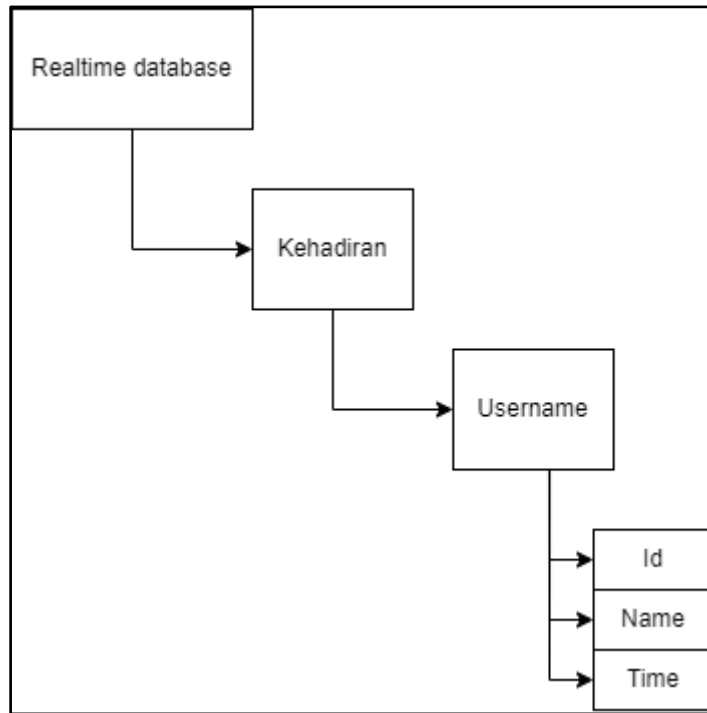
1. Android Studio
2. Github
3. Console Firebase Google

### 3.1.3 Design

Perancangan arsitektur perangkat lunak bertujuan untuk memberikan gambaran, perencanaan dan sketsa dalam pembuatan aplikasi sebelum menjadi kesatuan aplikasi yang utuh, berikut ini merupakan rancangan aplikasi presensi mobile menggunakan algoritma pengukur jarak haversine pada android:

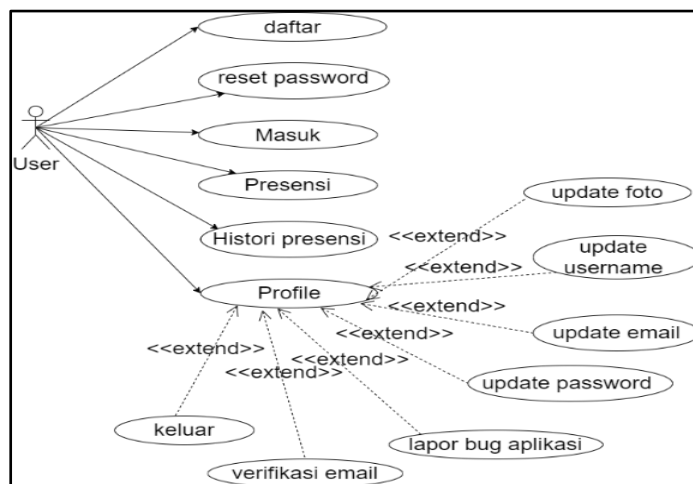
a. Hirarki basis data Firebase

Karena basis data firebase merupakan basis data bertipe NoSql, jadi tidak terdapat relasi pada tiap datanya, untuk menggantikan relasi antar data, firebase menggunakan hirarki JSON, jadi setiap data yang masuk kedalam basis data akan berurutan membentuk hirarki data.[11]



Gambar 2. Hirarki NoSql firebae aplikasi pada menu presensi

b. Use Case Diagram



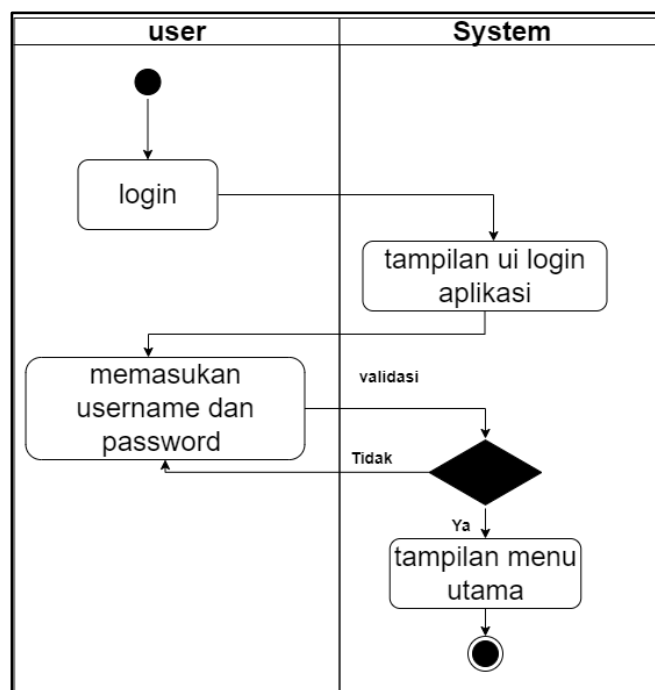
Gambar 3. Use Case diagram aplikasi presensi

Merupakan diagram yang dibuat untuk melihat korelasi antara user dan system yang akan dibuat serta mendeskripsikan sebuah interaksi antara satu aktor atau lebih dengan system yang dibuat. Use case diagram aplikasi terdapat pada gambar nomor 7.

c. Activity Diagram

Activity diagram menggambarkan aliran atau proses kerja dari sebuah sistem atau menu yang terdapat didalam aplikasi, diagram aktivitas secara tepat menggambarkan aktivitas yang dilakukan oleh sistem, bukan yang dilakukan oleh pengguna ketika berinteraksi dengan sistem. Activity diagram dalam aplikasi presensi ini adalah sebagai berikut:

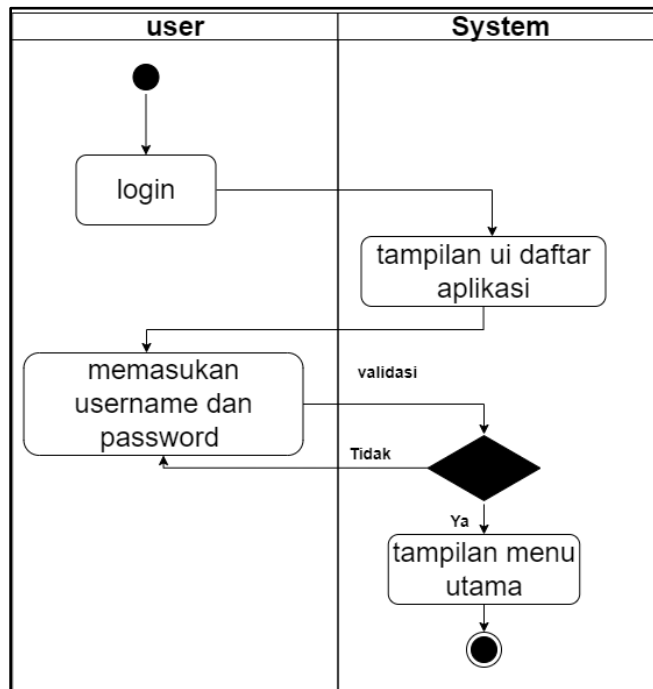
1. Activity diagram login



Gambar 4. Activity diagram login

Ketika sistem melakukan proses login, pertama pengguna akan diarahkan menuju halaman login dan mengisi password serta username, proses validasi akan dilakukan untuk mengecek username dan password pengguna terdaftar atau tidak didalam database, jika terdaftar maka akan diarahkan ke halaman dashboard, namun ketika gagal, akan muncul pesan error dan diarahkan ke halaman login aplikasi.

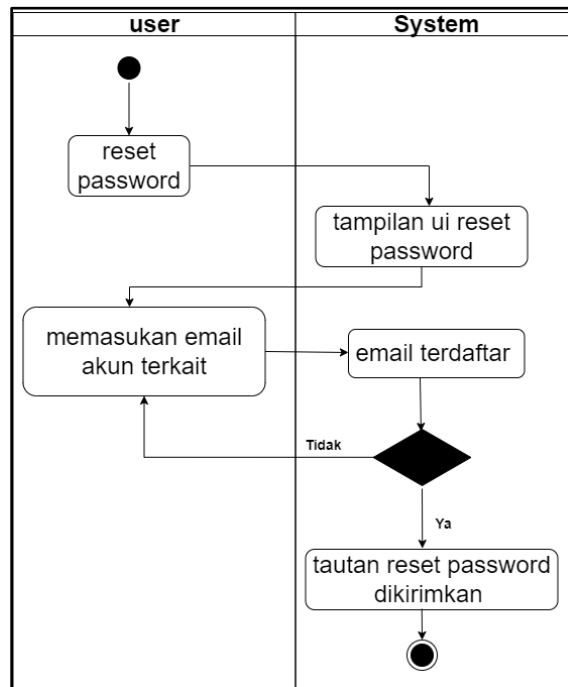
2. Activity diagram registrasi



Gambar 5. Activity diagram registrasi

Ketika sistem melakukan proses registrasi, pertama pengguna akan diarahkan menuju halaman registrasi dan mengisi password serta username, proses validasi akan dilakukan untuk mengecek username dan password pengguna terdaftar atau tidak didalam database, jika terdaftar maka registrasi dapat dilakukan akan diarahkan ke halaman dashboard, namun ketika gagal, akan muncul pesan error dan diarahkan ke halaman registrasi aplikasi.

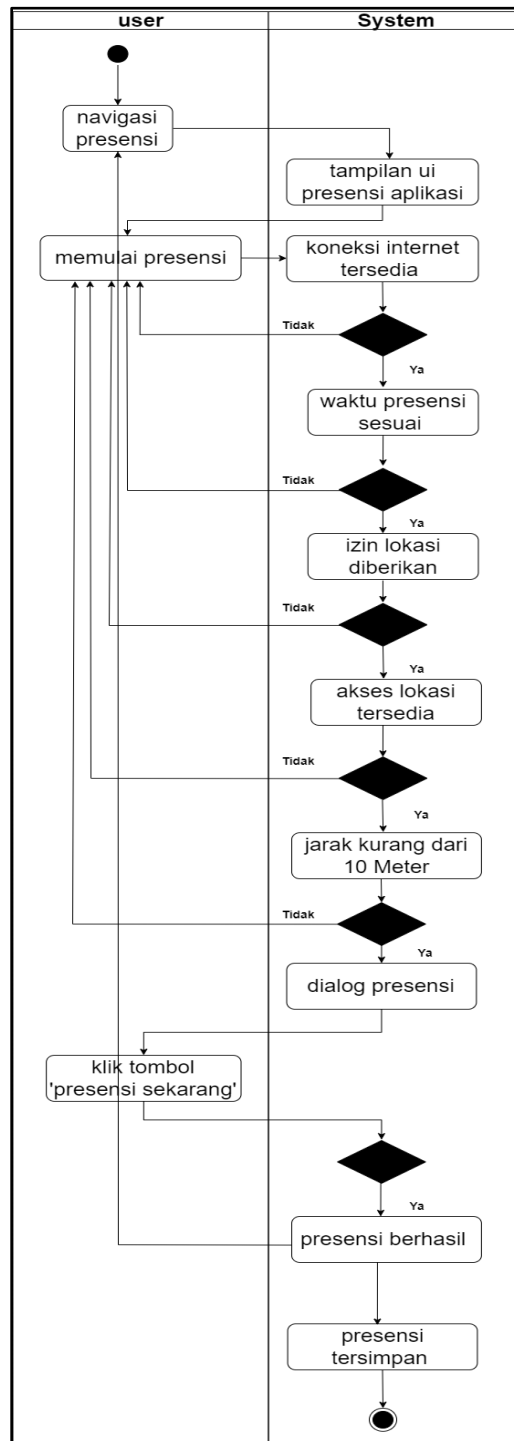
### 3. Activity reset password



Gambar 6. Activity diagram reset password

Reset password dapat dilakukan hanya ketika pengguna telah terdaftar atau memiliki akun, pengguna memasukkan email dan aplikasi secara otomatis mengecek email didalam database, ketika email terdaftar, aplikasi akan mengirimkan tautan untuk reset password, jika tidak, aplikasi akan mengirimkan pesan singkat berupa *toast* kepada pengguna berisikan informasi akun tidak terdaftar.

#### 4. activity presensi



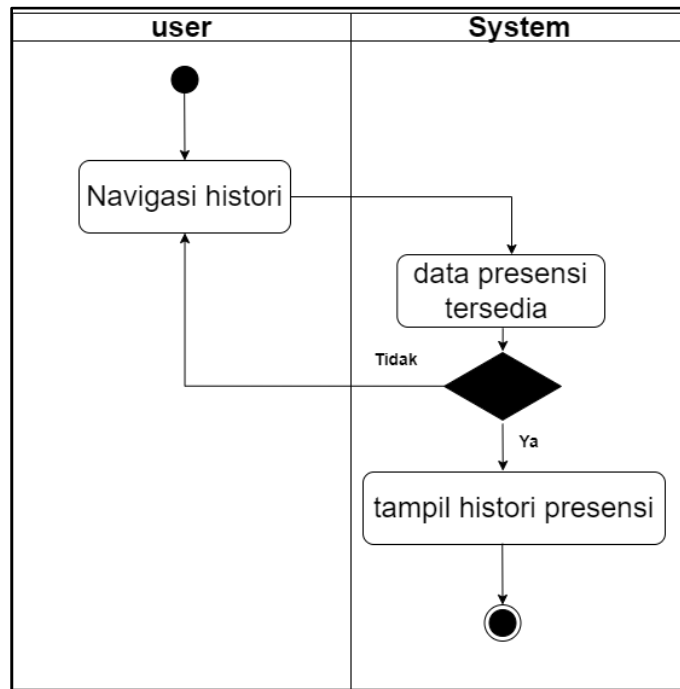
Gambar 7. Activity diagram menu presensi didalam aplikasi

Presensi aplikasi membutuhkan beberapa kondisi yang harus dipenuhi, diantaranya adalah koneksi internet, waktu presensi, *permission* dan jarak presensi, ketika semua kondisi terpenuhi, presensi pengguna dapat dilakukan, dialog presensi akan secara otomatis muncul dan pengguna cukup menekan tombol presensi



sekarang untuk presensi dan username akan digunakan untuk histori presensi didalam database.

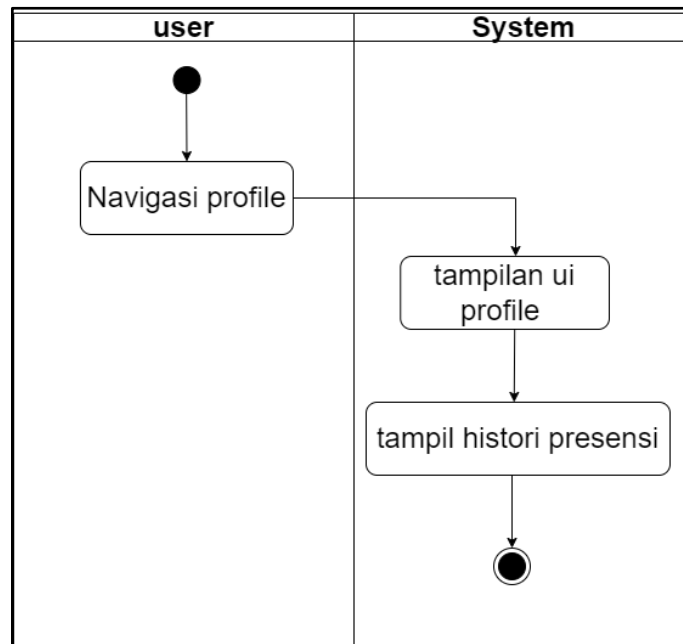
## 5. Activity history



Gambar 8. Actovity diagram menu history aplikasi presensi

Aplikasi akan menampilkan histori presensi tiap pengguna yang terdapat didalam database, Ketika pengguna mengakses menu histori pada aplikasi, aplikasi akan membaca informasi pengguna berupa nama dan waktu presensi yang dilakukan pengguna, tampilan akan dikemas menggunakan recyclerView untuk menjamin tampilan yang mulus dan dinamis.

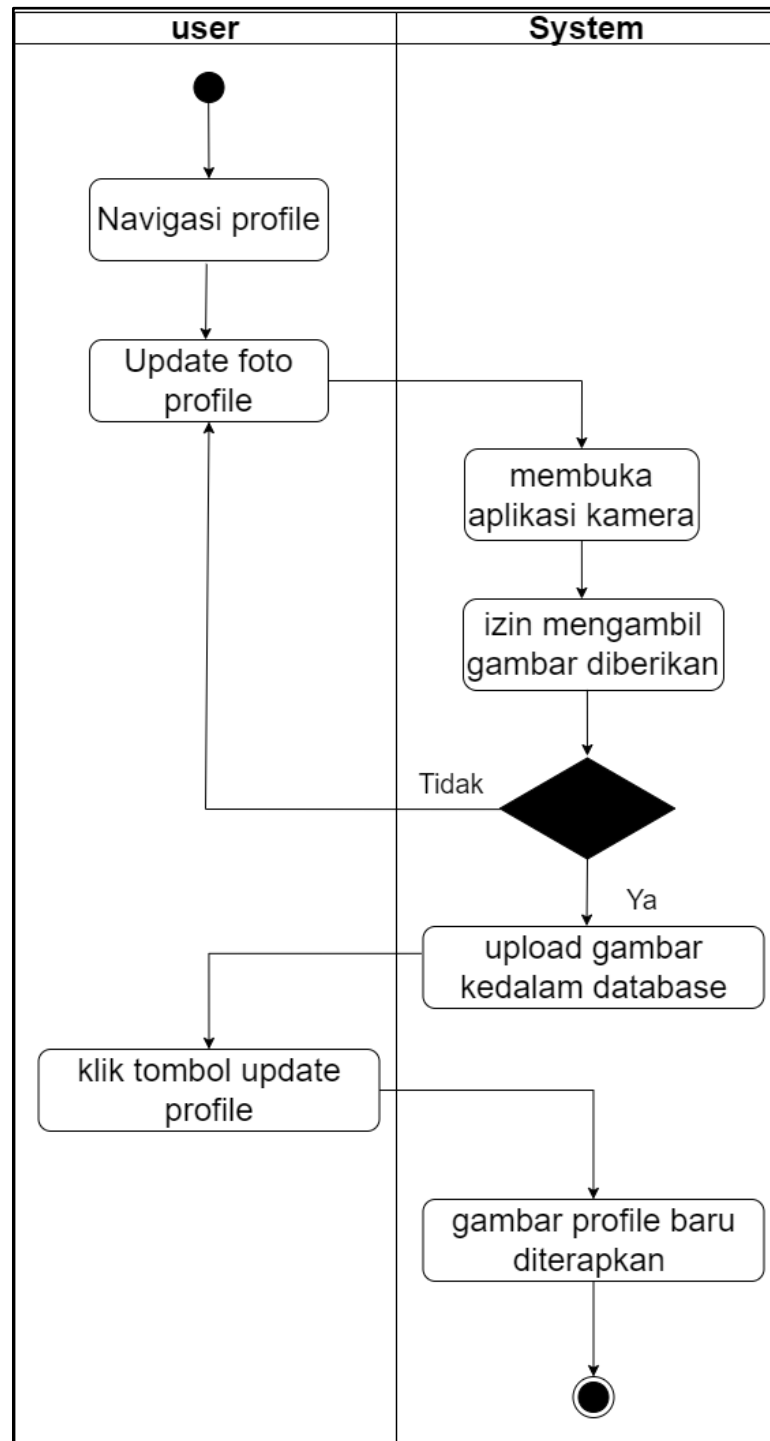
#### 6. Activity profile aplikasi



*Gambar 9. Activity diagram menu profile aplikasi*

Ketika pengguna membuka menu profil, aplikasi akan menampilkan beberapa opsi pengaturan akun pengguna, seperti ganti foto profile, password, email dan username.

#### 7. Activity update foto profile



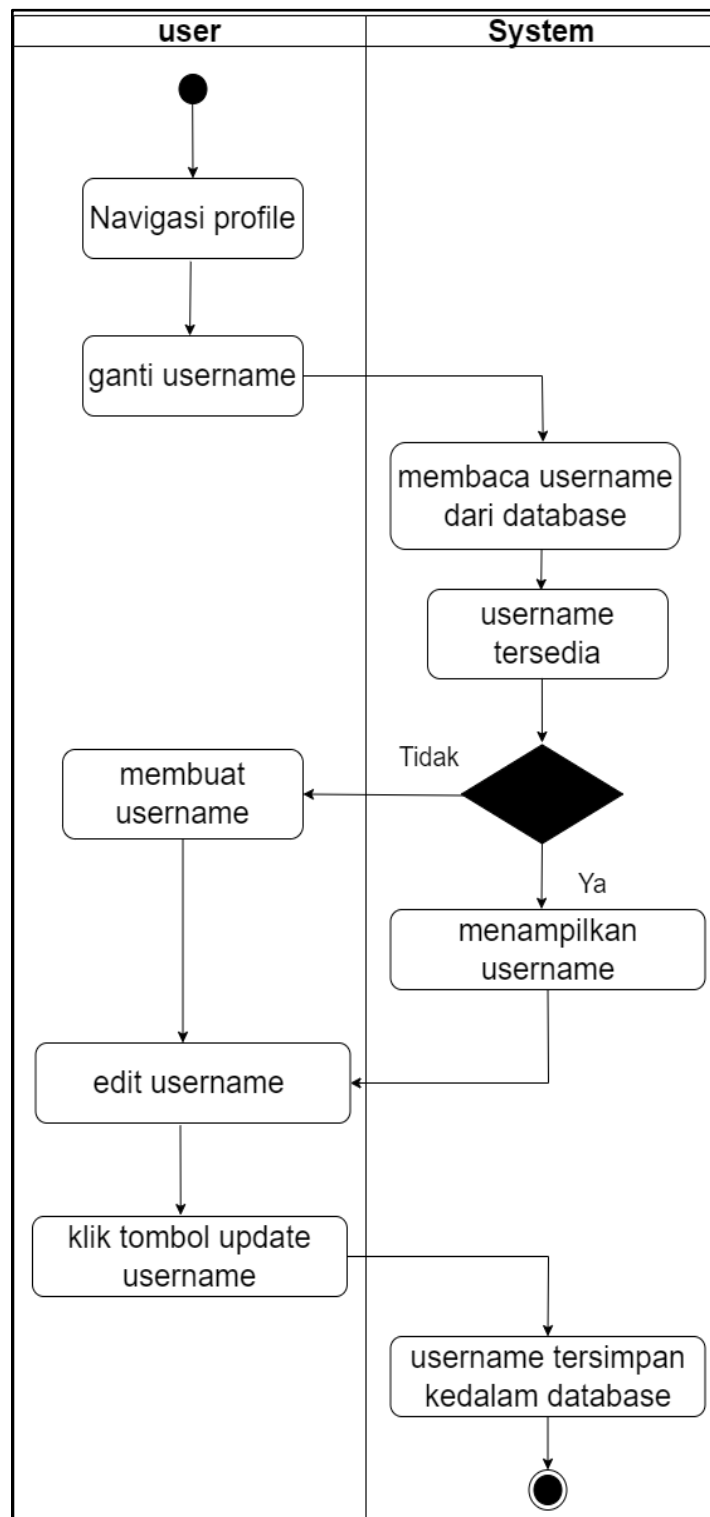
Gambar 10. Activity update foto profile

Aplikasi menyediakan fitur untuk mengatur foto profil pengguna, fitur ini dapat ditemukan didalam menu profile, pengguna cukup menekan foto profile dan aplikasi akan membuka kamera, pengguna dapat mengambil gambar dan mengunggahnya secara otomatis ke database, ketika update foto profile berhasil,

aplikasi akan menampilkan toast menandakan upload foto profile berhasil dilakukan.

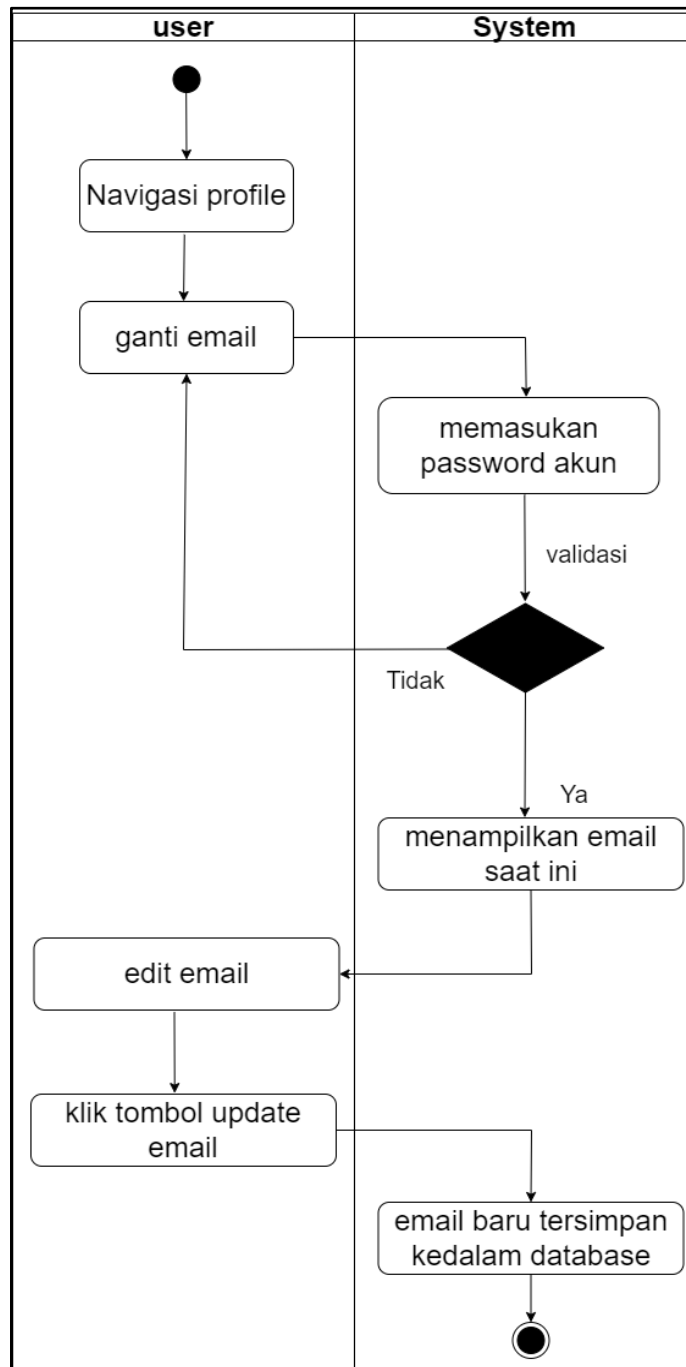
#### 8. Activity update username

Pengguna dapat mengubah username akun, fitur ganti username dapat ditemukan di menu profile aplikasi, pengguna cukup menekan tombol ganti username, aplikasi akan mengecek username yang ada didalam database, ketika tersedia, aplikasi akan menampilkanya ke edit text, namun jika tersedia, edit text akan kosong dan siap untuk menerima inputan username dari pengguna, ketika pengguna menekan tombol update username, maka username akan diset kedalam database dan aplikasi akan *redirect* kedalam menu profile.



Gambar 11. Activity diagram update username

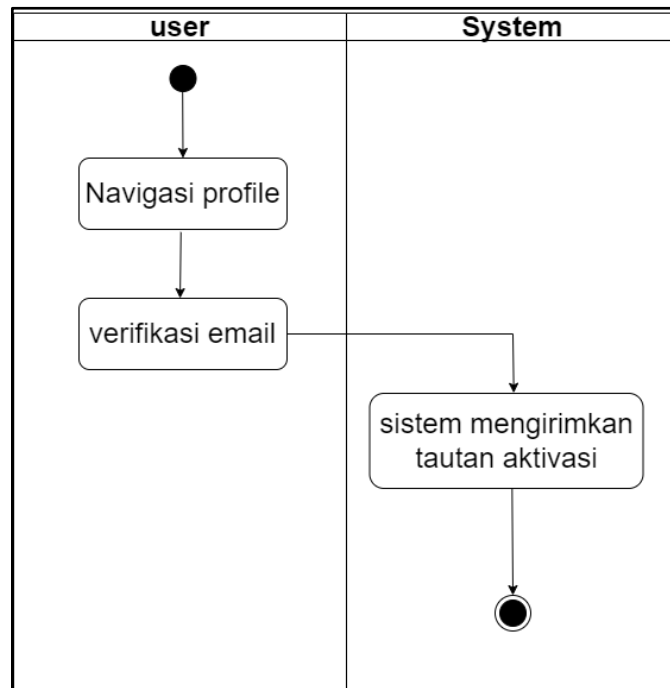
## 9. Activity ganti email



Gambar 12. Activity diagram ganti email

Pengguna dapat mengganti email mereka, fitur ganti email terdapat didalam menu profile, pengguna diharuskan untuk melakukan validasi password akun terkait, ketika password berhasil di verifikasi, aplikasi akan menampilkan email terkait dan pengguna dapat mengganti email, ketika berhasil mengganti email, aplikasi akan mengarahkan pengguna kedalam halaman profile dan membuat notifikasi bahwa password telah dirubah.

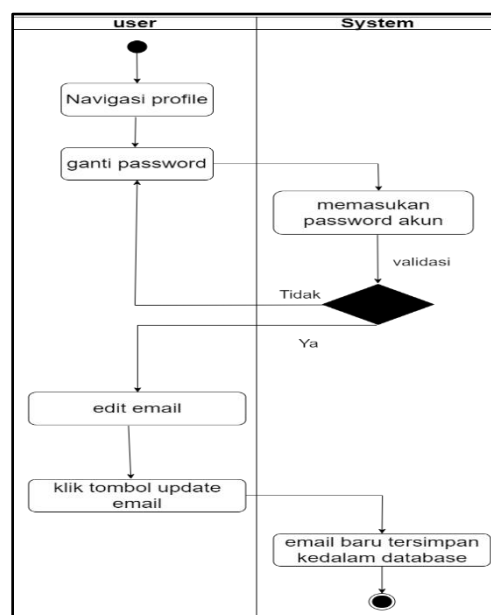
#### 10. Activity verifikasi email



*Gambar 13. Activity diagram verifikasi email*

Verifikasi email diperlukan untuk memvalidasi email pengguna, untuk melakukan verifikasi email, pengguna cukup menekan tombol verifikasi email di menu profile, dan aplikasi akan mengirimkan tautan ke email terkait, pengguna cukup melakukan klik pada tautan untuk melakukan verifikasi.

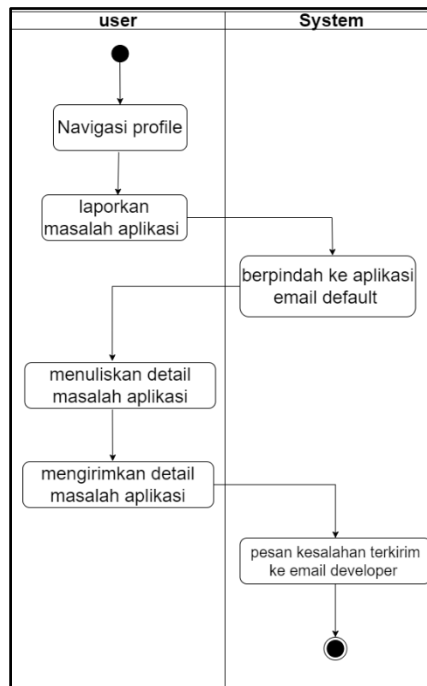
#### 11. Activity ganti password



*Gambar 14. Activity diagram ganti password*

Ganti password dapat dilakukan ketika pengguna berhasil melakukan validasi password akun, pengguna harus memasukkan password dua kali pada kolom inputan, ketika password baru berhasil dibuat, aplikasi akan mengarahkan ke halaman profile dan notifikasi akan dibuat secara otomatis untuk memberitahukan kepada pengguna jika password akun telah diubah.

## 12. Activity lapor masalah pada aplikasi (bug)

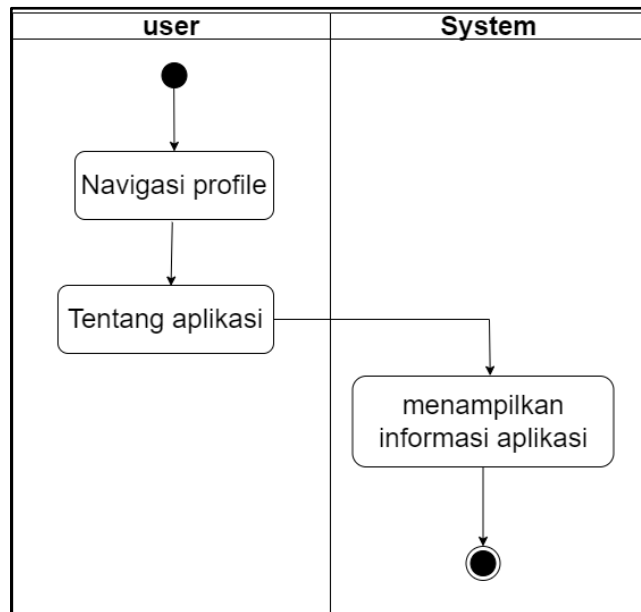


*Gambar 15. Activity diagram lapor masalah pada aplikasi*

Laporkan masalah aplikasi dapat ditemukan di menu profile, ketika pengguna melakukan aksi klik pada tombol laporkan masalah aplikasi, sistem akan mengarahkan ke aplikasi email default, pengguna bisa menuliskan keluhan dan akan dikirimkan ke email developer aplikasi.



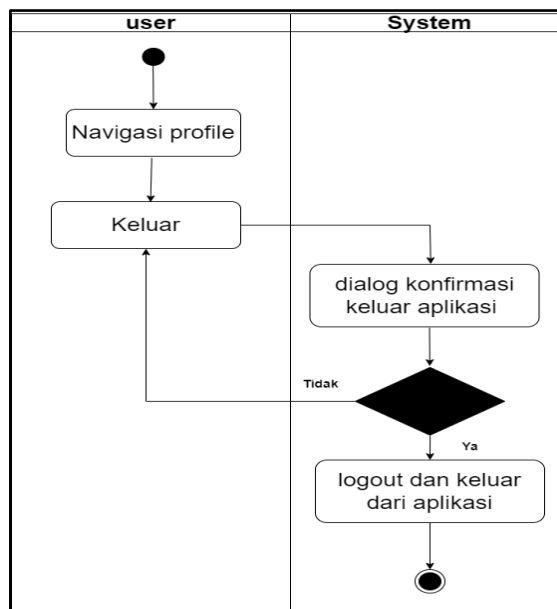
### 13. Activity tentang aplikasi



*Gambar 16. Activity diagram tentang aplikasi*

Tentang aplikasi menunjukkan informasi dasar aplikasi, seperti info penggunaan singkat aplikasi, versi aplikasi dan info developer aplikasi.

### 14. Activity keluar aplikasi



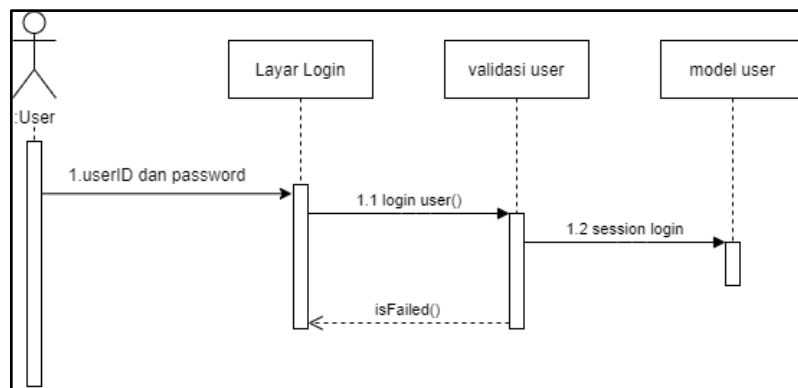
*Gambar 17. Activity diagram keluar aplikasi*

Ketika pengguna menekan tombol logout atau keluar di menu profile, aplikasi akan secara otomatis menghapus sesi login pengguna dan mengarahkan pengguna ke halaman login aplikasi

#### d. Sequence Diagram

Secara umum, sequence diagram digunakan untuk menggambarkan interaksi yang terjadi antara objek dan sistem yang digambarkan dalam bentuk pesan terhadap waktu. Sequence diagram terdiri dari dimensi vertikal untuk menggambarkan waktu dan dimensi horizontal untuk menggambarkan obyek-objek yang terkait, penggunaan sequence diagram dapat merepresentasikan langkah dari suatu kegiatan pada sistem untuk dapat menghasilkan output tertentu [12].

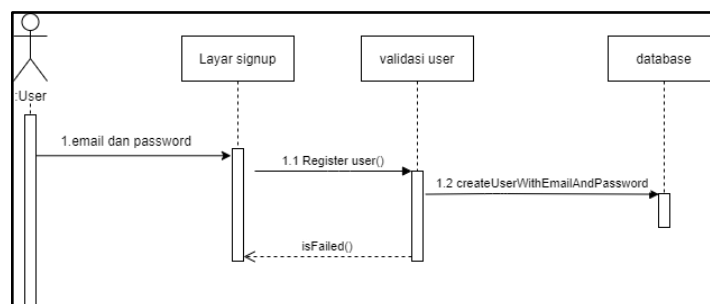
##### 1. Sequence diagram login



Gambar 18. Sequence diagram login

Proses login user dapat digambarkan pada gambar nomor 21, ketika user memasukkan inputan berupa userID dalam bentuk email pengguna dan password akun terkait, ketika user melakukan aksi klik pada tombol login, maka sistem akan melakukan validasi dan menjalankan fungsi login user(), ketika validasi berhasil, maka sistem akan membuat session baru berdasarkan database yang tersimpan dengan menggunakan model user, namun ketika validasi gagal, sistem akan mengembalikan *callback* kedalam layar login pengguna.

##### 2. Sequence diagram Signup

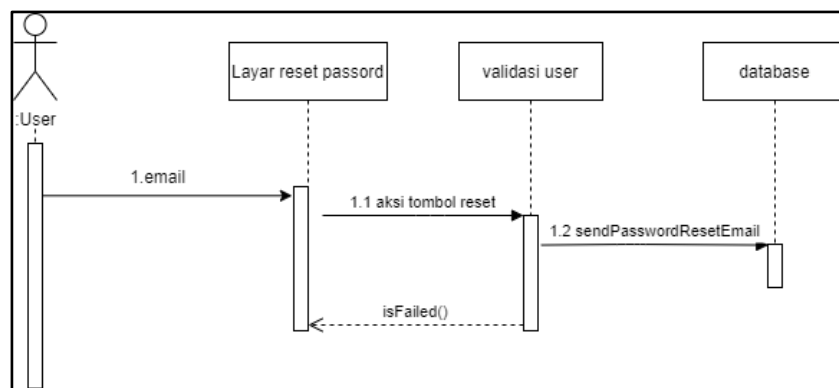


Gambar 19. Sequence diagram signup

Proses registrasi/signup user terdapat pada gambar nomor 22, ketika user mengisi inputan di layar Signup aplikasi dan menekan tombol register, maka sistem

akan melakukan validasi inputan yang terdiri dari email dan password, ketika validasi berhasil, user akan dialihkan menuju halaman beranda aplikasi dan data user akan tersimpan kedalam database dengan melakukan eksekusi fungsi `createUserWithEmailAndPassword` dengan parameter email dan password, namun ketika validasi aplikasi gagal, sistem akan memberikan *callback* kepada user mengenai error yang terjadi.

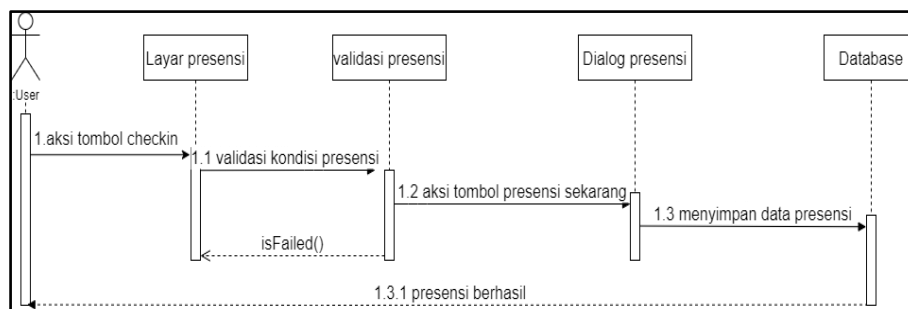
### 3. Sequence diagram reset password



Gambar 20. Sequence diagram reset password

Reset password dilakukan dengan melakukan validasi inputan user bertipe email yang dilakukan oleh sistem, ketika email terdaftar didalam database, maka fungsi `sendPasswordResetEmail()` akan dijalankan dan email terkait akan menerima tautan untuk reset email akun.

### 4. Sequence diagram presensi

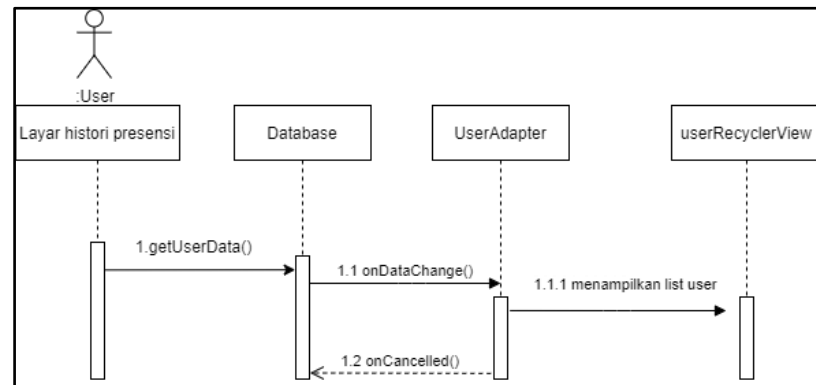


Gambar 21. Sequence diagram presensi

Proses presensi dimulai ketika user melakukan aksi klik pada button di layar presensi, setelah button diklik, akan ada beberapa kondisi yang harus terpenuhi sebelum presensi dapat dilakukan, kondisi itu antara lain cek koneksi internet, waktu presensi yang diizinkan, cek perizinan aplikasi, cek kondisi fitur GPS pada perangkat android dan terakhir adalah jarak presensi terpenuhi, untuk memperoleh hasil kalkulasi jarak yang sesuai, maka digunakan algoritma haversine.

Setelah semua kondisi terpenuhi, maka aplikasi akan menampilkan dialog dimana user bisa melakukan presensi dengan melakukan aksi pada tombol “presensi sekarang” dan secara otomatis username dari pengguna akan dijadikan bukti presensi, presensi dikategorikan berhasil ketika sistem mengembalikan *callback successful* direpresentasikan dengan *toast* berwarna hijau.

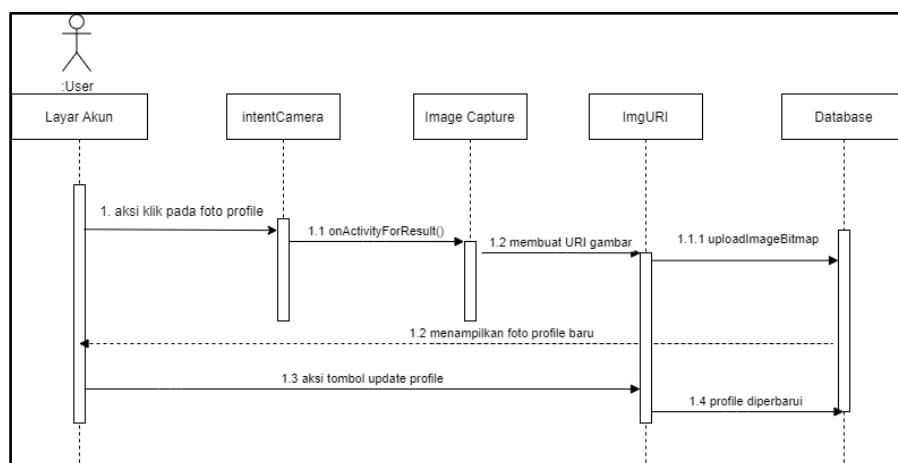
#### 5. Sequence diagram histori presensi



Gambar 22. Sequence diagram histori presensi

Ketika user berada pada bagian menu histori presensi, aplikasi akan menampilkan histori presensi yang tersedia dengan melakukan koneksi ke database, jika data tersedia dan koneksi berhasil, data akan ditampilkan dalam bentuk list objek *recyclerView*, namun ketika data tidak tersedia dan koneksi gagal, aplikasi akan menampilkan data tidak tersedia berupa gambar dan *log error* dalam bentuk *toast*.

#### 6. Sequence diagram update foto profil

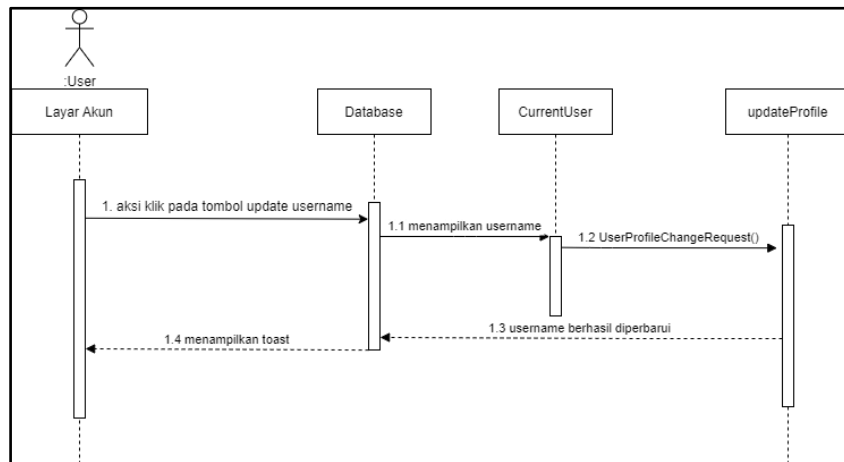


Gambar 23. Sequence diagram update foto profil

User dapat melakukan update foto profil dengan melakukan aksi klik pada foto profil di menu akun, aplikasi akan membuka kamera dan user dapat mengambil

gambar menggunakan kamera depan ataupun kamera belakang, setelah gambar diambil, aplikasi akan menampilkan gambar terbaru hasil pengambilan gambar, ketika user melakukan aksi klik pada tombol update profile, maka gambar terbaru akan menggantikan gambar lama di dalam database dan akan ditampilkan gambar baru pada aplikasi dibagian menu akun.

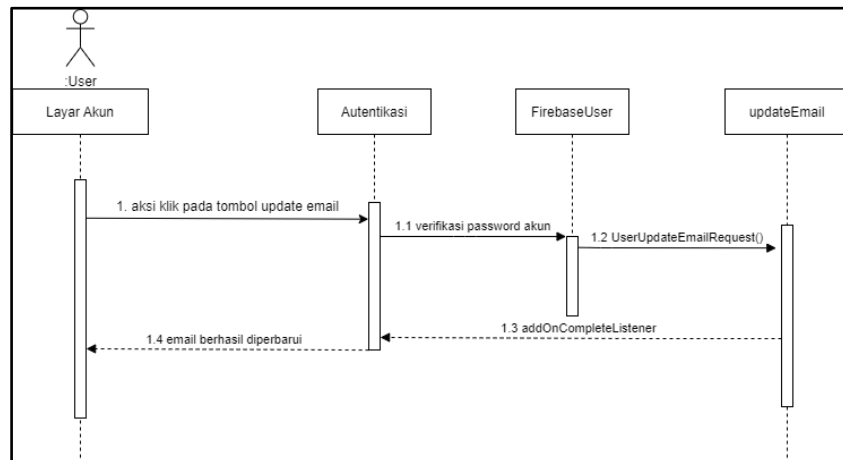
#### 7. Sequence diagram update username



Gambar 24. Sequence diagram update username

Untuk dapat melakukan update username, pengguna harus menekan tombol update username yang terdapat pada menu akun aplikasi, aplikasi akan membaca username berdasarkan *current user* yang sedang login, ketika username ditemukan didalam database, maka akan ditampilkan kepada pengguna, namun ketika tidak ada, maka kolom username akan kosong, untuk membuat username baru, pengguna harus memasukkan kedalam kolom update username dan memberikan aksi klik pada tombol update username, maka username terbaru akan tersimpan kedalam database dan aplikasi akan membuat *toast* yang menandakan update username berhasil.

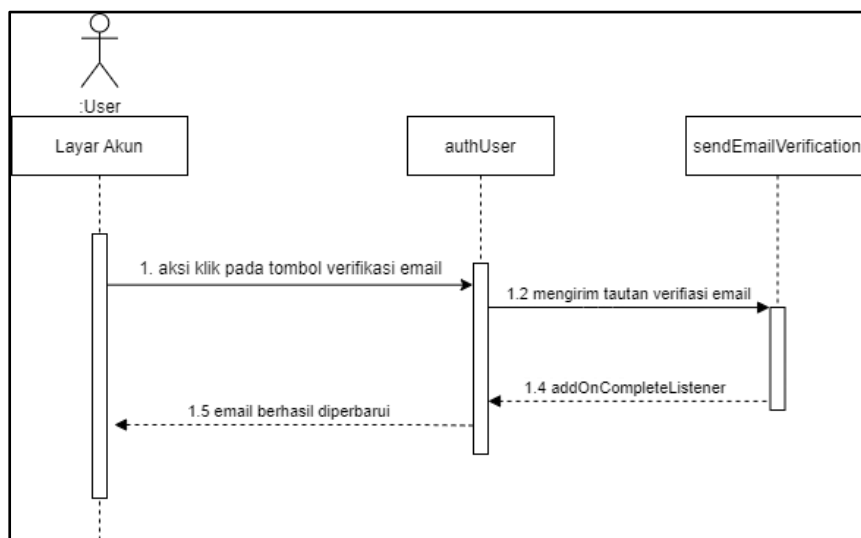
#### 8. Sequence diagram ganti email



Gambar 25. Sequence Diagram update email

Ketika pengguna akan mengganti email pada akun terkait, pengguna harus melakukan autentikasi password akun, hal ini ditujukan untuk keamanan akun pengguna, setelah autentikasi password berhasil, aplikasi akan menampilkan email yang terkait dengan akun dan pengguna bisa menggantinya jika diperlukan, ketika tombol update email ditekan, maka aplikasi akan mengirimkan permintaan update email kedalam database dan respons dari database akan ditangkap aplikasi dan memunculkan toast kepada user ketika update email berhasil dilakukan.

#### 9. Sequence diagram verifikasi email

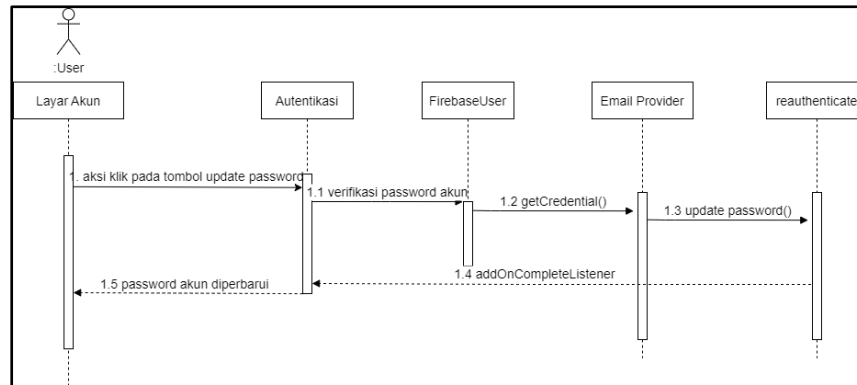


Gambar 26. Sequence diagram verifikasi email

Verifikasi email diperlukan untuk memastikan email yang digunakan user adalah email yang aktif, selain daripada itu, verifikasi email dapat membantu user ketika lupa password, sehingga tautan reset password yang dikirimkan aplikasi akan terkirim dan user dapat melakukan reset password, untuk melakukan verifikasi

email, pengguna cukup memberikan aksi klik pada tombol verifikasi email, dan aplikasi akan mengirimkan tautan ke email akun terkait, ketika tautan itu diklik, maka verifikasi email berhasil.

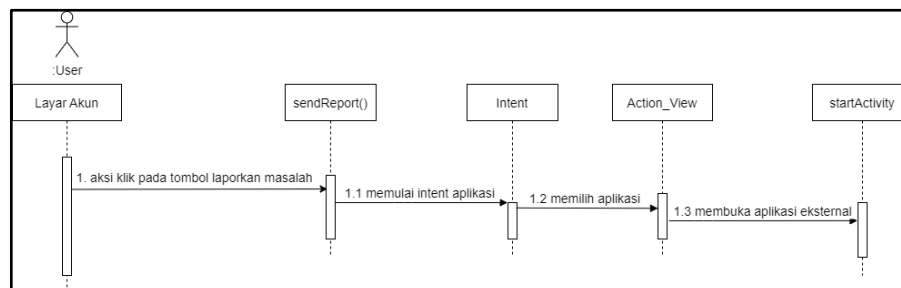
#### 10. Sequence diagram update password



Gambar 27. Sequence diagram update password

Update password dapat dilakukan ketika pengguna telah *login* kedalam aplikasi, ketika pengguna memutuskan untuk mengganti password lama ke password baru, pengguna dapat melakukannya secara cepat pada menu akun, tombol ganti password akan memandu pengguna memperbarui password, langkah pertama aplikasi akan memastikan pemilik akun atau biasa dikenal dengan autentikasi, dengan memasukkan password yang sesuai dengan akun, pengguna akan dengan mudah melewati proses ini, tabapan setelah autentikasi adalah memasukkan password baru, password harus terdiri dari 8 karakter dan harus sesuai pada kolom konfirmasi, sebagai informasi tambahan, kolom konfirmasi ditujukan untuk user melakukan input ulang password baru, ketika kolom password baru dan kolom password telah sesuai, maka aplikasi akan melakukan *reauthenticate* dengan email provider dan password pengguna telah diperbarui.

#### 11. Sequence diagram laporkan masalah aplikasi



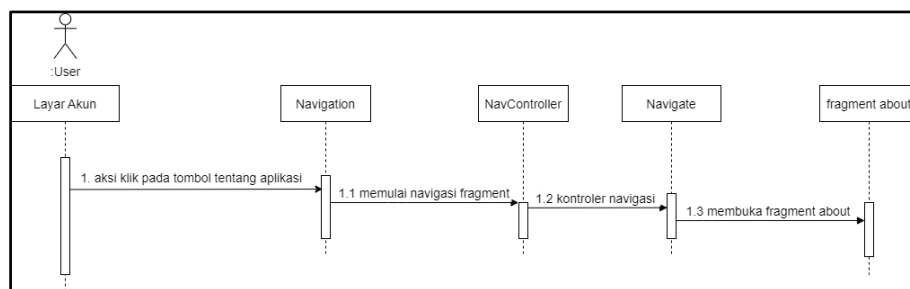
Gambar 28. Sequence diagram laporkan masalah aplikasi

Ketika terjadi masalah pada aplikasi atau biasa dikenal dengan *bug*, pengguna dapat melakukan *report* terkait masalah yang terjadi pada aplikasi, hal



ini diperlukan sebagai masukan kepada pengembang aplikasi mengenai masalah pada aplikasi mereka, pengguna dapat menemukan fitur ini pada menu akun, ketika tombol laporkan masalah aplikasi diklik, maka sistem android akan memilih aplikasi eksternal yang mendukung subjek berupa email, ketika ditemukan, pengguna akan diarahkan kedalam aplikasi tersebut dan menuliskan detail masalah yang terjadi pada aplikasi.

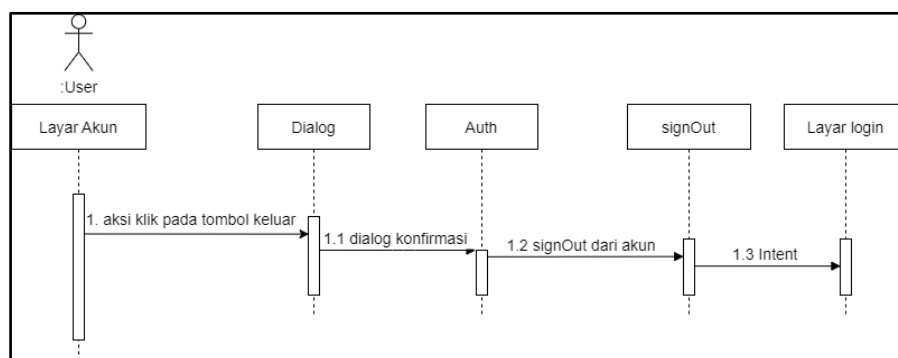
## 12. Sequence diagram tentang aplikasi



Gambar 29. Sequence diagram tentang aplikasi

Tentang aplikasi berisi informasi dasar mengenai aplikasi, seperti versi aplikasi, developer dan info pendukung lainnya, pengguna dapat membaca tentang aplikasi yang terdapat pada menu akun dengan melakukan aksi klik pada tombol tentang aplikasi, aplikasi akan melakukan navigasi dan mengarahkan pengguna kedalam fragment yang sesuai yaitu *fragment aboutApp*.

## 13. Sequence diagram logout



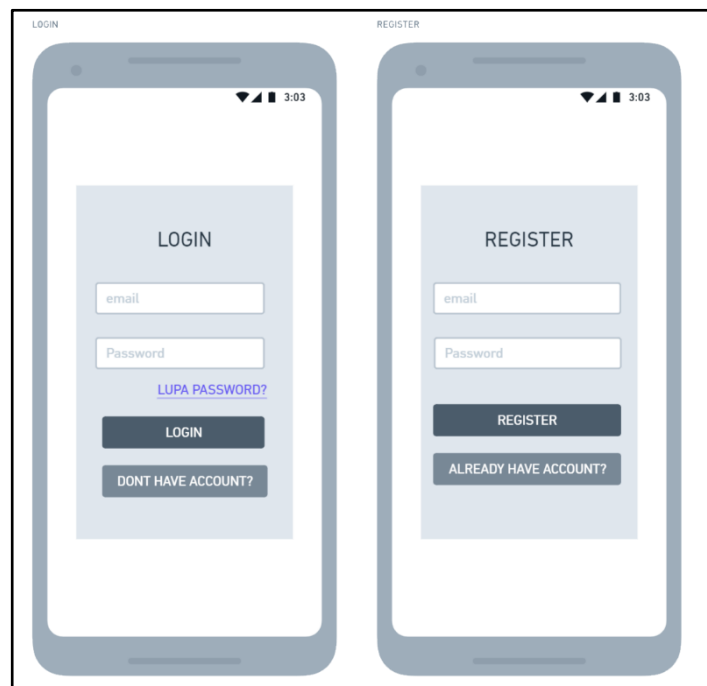
Gambar 30. Sequence diagram logout

Ketika pengguna memutuskan untuk keluar dari akun terkait atau biasa dikenal dengan *logout*, aplikasi akan menampilkan dialog konfirmasi sebagai pengaman sederhana bilamana pengguna sebenarnya tidak menginginkan hal tersebut, ketika pengguna memutuskan untuk tetap keluar dari akun terkait, aplikasi akan memanggil fungsi signout yang terdapat pada objek firebase, dan pengguna akan diarahkan ke halaman login aplikasi.

e. Wireframe aplikasi

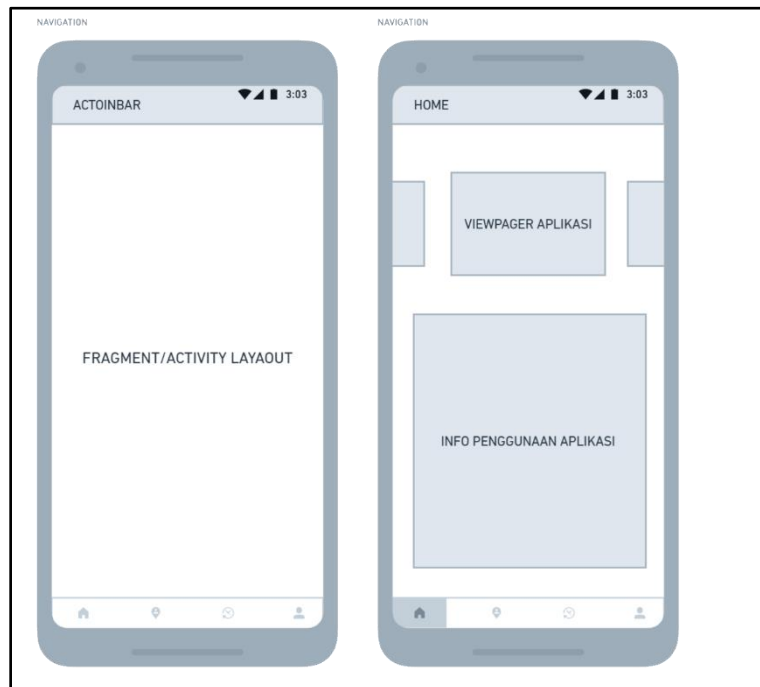
Sebagai tahapan iterasi dalam pengembangan aplikasi, maka tahapan selanjutnya setelah pembuatan UML adalah *wireframe*. *Wireframe* adalah kerangka awal dalam pembuatan sebuah aplikasi, baik itu berbasis website maupun mobile , tahapan pembuatan wireframe perlu dilakukan terlebih dahulu sebelum pembuatan UI (*user interface*) dan diimplementasikan kedalam aplikasi [13].

1. Wireframe login dan signup



*Gambar 31. Wireframe login dan signup aplikasi presensi*

2. Wireframe beranda aplikasi



*Gambar 32. Wireframe halaman beranda aplikasi*

### 3. Wireframe halaman dan histori presensi



*Gambar 33. Wireframe halaman presensi dan histori presensi*

### 4. Wireframe halaman profil pengguna



Gambar 34. Wireframe halaman profil pengguna aplikasi presensi

f. Antar muka pengguna (*User Interface*)

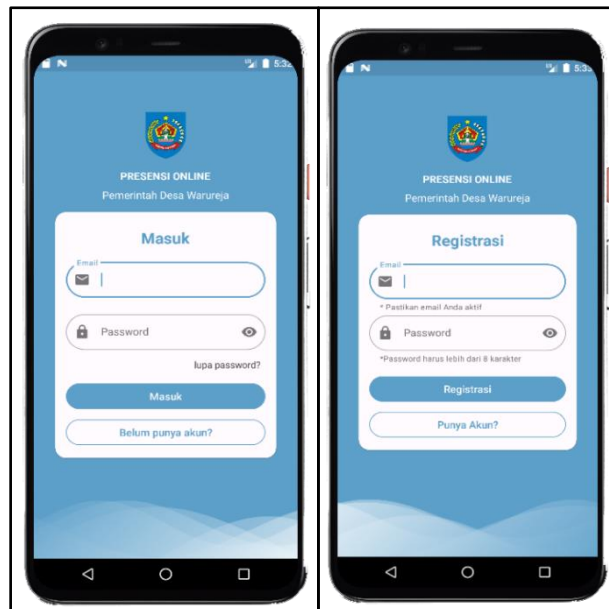
Antar muka pengguna atau biasa disebut dengan *User Interface* adalah tahapan interaksi yang terjadi antara pengguna dan aplikasi dengan suatu perintah didalam aplikasi, seperti memasukkan data dan penggunaan konten didalam aplikasi[13]. Untuk menghasilkan antar muka pengguna yang ramah penggunaan, maka dalam pengembangan aplikasi presensi ini akan digunakan komponen *material design* sebagai acuan utama dalam menyusun antar muka pengguna disetiap bagian aplikasi.

1. Antar muka pengguna *splash screen*



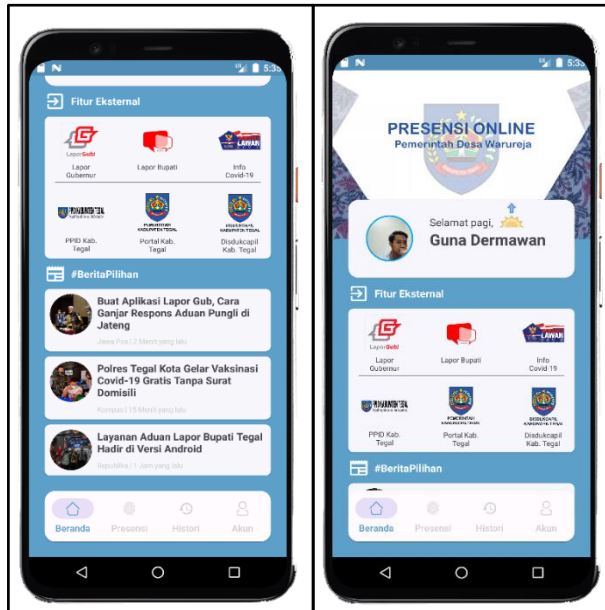
*Gambar 35. Antar muka pengguna splash screen aplikasi*

## 2. Antar muka pengguna login dan signup



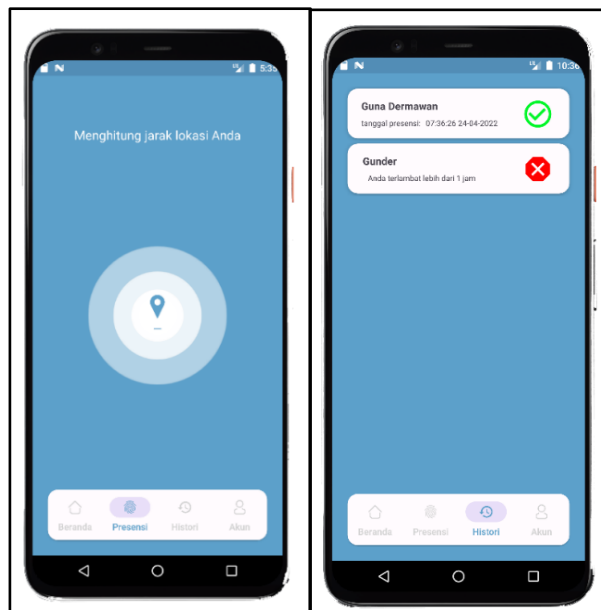
*Gambar 36. Antar muka pengguna login dan signup*

## 3. Antar muka pengguna halaman beranda



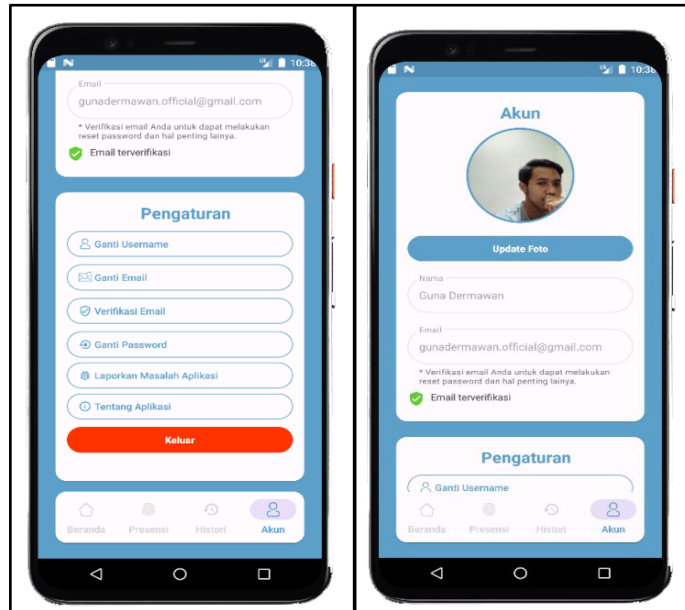
*Gambar 37. Antar muka pengguna halaman beranda aplikasi*

4. Antar muka pengguna halaman presensi dan histori presensi



*Gambar 38. Antar muka halaman presensi dan histori presensi aplikasi*

5. Antar muka pengguna halaman profil pengguna



*Gambar 39. Antar muka pengguna halaman profil aplikasi*

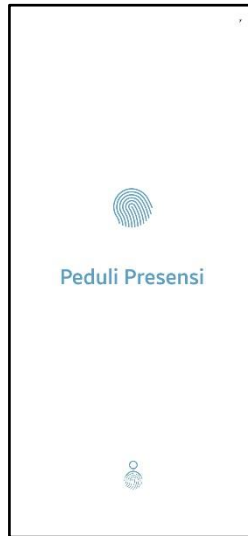
### 3.2 Implementasi dan Testing

Tahapan implementasi dilakukan agar aplikasi dapat berjalan dengan baik sesuai dengan rancangan sistem pada tahap sebelumnya, dan testing diperlukan untuk memastikan semua fungsi yang terdapat didalam aplikasi dapat berjalan dengan baik. Dalam tahap ini dilakukan pengkodean menggunakan bahasa pemrograman kotlin dan menggunakan IDE android studio sebagai lingkungan pengembangan aplikasi, sedangkan untuk penyimpanan data aplikasi, digunakan layanan firebase dari google untuk mengakomodasi penyimpanan hasil presensi, akun pengguna dan foto profile didalam database firebase.

#### 4.2.1 Implementasi

Implementasi berisikan aplikasi yang telah terinstal didalam perangkat android sebenarnya, implementasi menerapkan semua tahapan wireframe dan desain secara keseluruhan aplikasi pada tahap desain rancangan sistem.

1. Implementasi desain *splash screen*



*Gambar 40. Implementasi desain splash screen*

Splash screen adalah tampilan yang pertama kali muncul ketika aplikasi dibuka, splash screen pada gambar 43 terdapat 3 atribut, yaitu teks bertuliskan nama aplikasi, dan dua gambar dengan satu gambar animasi, dan satu gambar icon aplikasi, semua atribut pada splash screen menggunakan warna yang seragam untuk memberikan pengalaman yang profesional ketika pengguna membuka aplikasi.

## 2. Implementasi desain login



*Gambar 41. Implementasi desain login aplikasi*

Tampilan login aplikasi terdapat beberapa atribut, seperti logo dari pemerintah kabupaten Tegal, teks untuk judul login dan kolom inputan teks beserta tombol untuk membuat aksi login aplikasi. Pada layar login, pengguna dapat memasukkan inputan bertipe email dan teks untuk password,



ketika pengguna menekan tombol masuk, aplikasi akan melakukan validasi inputan sebelum mengarahkan pengguna kedalam menu utama aplikasi.

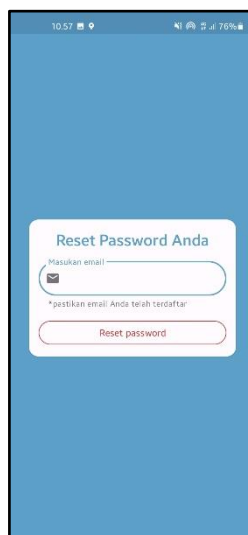
### 3. Implementasi desain signup



*Gambar 42. Implementasi desain signup aplikasi*

Tampilan *signup* berisikan atribut yang hampir sama dengan tampilan login, seperti teks untuk judul, logo instansi, kolom inputan pengguna dan tombol aksi aplikasi registrasi.

### 4. Implementasi desain lupa password



*Gambar 43. Implementasi desain lupa password aplikasi*

Implementasi desain pada layar lupa password terdiri dari kolom inputan pengguna berupa email, teks sebagai judul dan tombol untuk memulai aksi aplikasi, terdapat juga ikon email sebagai informasi tambahan kepada pengguna.

## 5. Implementasi desain beranda



*Gambar 44. Implementasi desain beranda aplikasi*

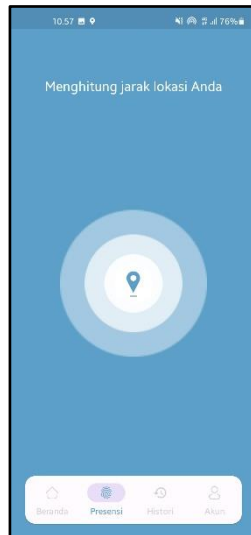
Layar beranda adalah tampilan awal ketika pengguna berhasil login ataupun signup, didalam tampilan beranda terdapat banner atau gambar yang menunjang tampilan aplikasi, banner juga mempertegas jenis aplikasi yang sedang digunakan yaitu aplikasi presensi online pada kantor balai desa Warureja Kab. Tegal.

Aplikasi juga akan memberikan sambutan sederhana kepada pengguna, disertai dengan username dan foto profil pada tiap akun yang terkait, ketika pertama kali registrasi, maka username dan foto profile yang ditampilkan adalah tampilan *default* dan pengguna dapat menggantinya pada menu akun.

Untuk menunjang kebutuhan pengguna, aplikasi juga dilengkapi dengan fitur eksternal berupa tautan yang akan mengarahkan pengguna kedalam tautan tertentu, ikon yang terdapat pada tautan eksternal akan membantu pengguna memilih tautan yang tepat dan sesuai dengan kebutuhan.

Bagian paling bawah pada layar beranda adalah list berita, berisikan tautan yang diwakili dengan gambar, teks judul dan kalimat utama berita, serta waktu publikasi berita, namun untuk pengembangan, fitur list berita ini hanya bersifat *dummy* dan perlu dilakukan pengembangan lanjutan untuk membuat fitur ini menjadi dinamis menggunakan bantuan API.

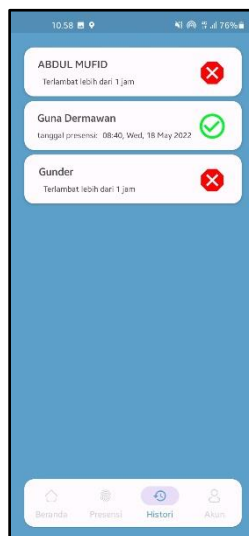
## 6. Implementasi desain halaman presensi



*Gambar 45. Implementasi desain halaman presensi*

Halaman presensi menitikberatkan pada fungsionalitas, disini adalah fungsi utama dari aplikasi, terdapat tiga atribut utama yaitu teks sebagai judul dan info halaman, gambar yang digunakan sebagai tombol presensi dan animasi yang menstimulasikan penghitungan jarak.

#### 7. Implementasi desain halaman histori presensi

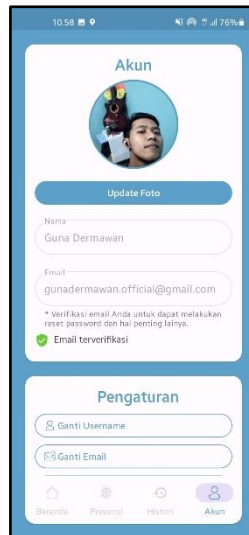


*Gambar 46. Implementasi desain halaman histori presensi*

Halaman histori presensi berfungsi untuk menampilkan daftar presensi yang telah dilakukan oleh setiap pengguna, baik presensi yang tepat waktu ditandai dengan ikon berwarna hijau, maupun presensi yang ditandai sebagai terlambat yang ditandai dengan ikon berwarna merah, terdapat

atribut lainya, yaitu teks yang digunakan untuk menampilkan usernane dan tanggal presensi dilakukan, semua daftar disajikan dalam bentuk daftar atau list dengan bantuan *recyclerview* pada android.

#### 8. Implementasi desain halaman akun



Gambar 47. Implementasi desain halaman akun aplikasi

Halaman akun pengguna terdiri dari beberapa informasi akun pengguna, seperti foto profil, username, dan email akun terkait, terdapat juga atribut teks yang digunakan untuk judul dan informasi halaman, selain hal diatas, halaman akun juga memuat atribut tombol sebagai fungsi tambahan manajemen pengguna seperti ganti password, email, username, *report bug*, dan keluar atau *logout* dari akun terkait.

#### 4.2.2 Testing

Pada tahapan ini, dilakukan pengujian terhadap sistem aplikasi presensi berbasis android, hal ini diperlukan untuk mengetahui kinerja aplikasi secara keseluruhan serta memastikan semua fungsi yang terdapat didalam aplikasi dapat berjalan dengan baik. Teknik pengujian yang digunakan adalah teknik pengujian menggunakan metode Blackbox, metode ini akan menguji semua proses inputan yang dilakukan oleh pengguna dan respon aplikasi terhadap inputan tersebut.

Table 3. Pengujian Blackbox

No	User Interface	Input	Output	Hasil Uji
<b>Splashscreen</b>				
1.	Membuka aplikasi presensi	-	Menampilka n	Komponen aplikasi berjalan dengan baik

			splashscreen aplikasi	
<b>Login</b>				
1.	Login dengan email dan password akun dalam keadaan kosong	Email: - Password:-	Menampilka n pesan error pada kolom inputan email	Komponen aplikasi berjalan dengan baik
2.	Login dengan password dalam keadaan kosong	Email: valid Password:-	Menampilka n pesan error pada kolom inputan password	Komponen aplikasi berjalan dengan baik
3.	Login dengan email dalam keadaan kosong	Email: - Password: valid	Menampilka n pesan error pada kolom inputan email	Komponen aplikasi dapat berjalan dengan baik
4.	Login dengan email dan password dalam keadaan salah	Email: invalid Password: invalid	Menampilka n toast singkat akun tidak terdaftar	Komponen aplikasi berjalan dengan baik
5.	Login dengan email benar dan password salah	Email:validp assword: invalid	Menampilka n toast singkat password salah`	Komponen aplikasi berjalan dengan baik
6.	Login dengan email salah dan password benar	Email:invalid Password: valid	Menampilka n toast singkat akunn pengguna tidak ditemukan	Komponen aplikasi berjalan dengan baik
7.	Masuk kedalam halaman lupa password	Klik tombol lupa password	Berpindah kedalam halaman lupa password	Komponen aplikasi berjalan dengan baik
8.	Masuk kedalam halaman signup	Klik tombol belum punya akun	Berpindah kedalam halaman signup	Komponen aplikasi berjalan dengan baik
<b>Lupa password</b>				
1.	Reset password dengan kolom inputan email dalam keadaan kosong	Email: -	Menampilka n pesan email tidak boleh kosong	Komponen aplikasi berjalan dengan baik

			pada kolom inputan	
2.	Reset password dengan email tidak terdaftar	Email: invalid	Menampilkan pesan email tidak terdaftar	Komponen aplikasi berjalan dengan baik
3.	Reset password dengan email terdaftar	Email: valid	Mengirimkan tautan reset password ke email terkait	Komponen aplikasi berjalan dengan baik
<b>Signup</b>				
1.	Signup dengan email dan password akun dalam keadaan kosong	Email: - Password: -	Menampilkan pesan error pada kolom inputan email	Komponen aplikasi berjalan dengan baik
2.	Signup dengan password dalam keadaan kosong	Email: valid Password: -	Menampilkan pesan error pada kolom inputan password	Komponen aplikasi berjalan dengan baik
3.	Signup dengan email dalam keadaan kosong	Email: - Password: valid	Menampilkan pesan error pada kolom inputan email	Komponen aplikasi dapat berjalan dengan baik
4.	Signup dengan email dan password dalam keadaan terdaftar	Email: invalid Password: invalid	Menampilkan toast singkat akun telah terdaftar	Komponen aplikasi berjalan dengan baik
5.	Masuk kedalam halaman login	Klik tombol punya akun	Berpindah kedalam halaman login	Komponen aplikasi berjalan dengan baik
<b>Beranda</b>				
1.	Melakukan aksi klik pada ikon fitur eksternal lapor gubernur	Klik ikon lapor gubernur	Membuka tautan <a href="http://laporgub.jatengprov.go.id">laporgub.jatengprov.go.id</a>	Komponen aplikasi berjalan dengan baik
2.	Melakukan aksi klik pada ikon fitur eksternal lapor bupati	Klik ikon lapor gubernur	Membuka aplikasi playstore untuk mengunduh aplikasi lapor bupati tegal	Komponen aplikasi berjalan dengan baik

3.	Melakukan aksi klik pada ikon fitur eksternal info covid-19	Klik ikon info covid-19	Membuka tautan covid19.tegal.kab.go.id	Komponen aplikasi berjalan dengan baik
4.	Melakukan aksi klik pada ikon fitur eksternal PPID Kab. Tegal	Klik ikon PPID Kab. Tegal	Membuka tautan ppid.tegalkab.go.id	Komponen aplikasi berjalan dengan baik
5.	Melakukan aksi klik pada ikon fitur eksternal Portal Kab. Tegal	Klik ikon Portal Kab. Tegal	Membuka tautan utama.tegalkab.go.id	Komponen aplikasi berjalan dengan baik
6.	Melakukan aksi klik pada ikon fitur eksternal Disdukcapil Kab. Tegal	Klik ikon Disdukcapil Kab. Tegal	Membuka tautan disdukcapil.tegalkab.go.id	Komponen aplikasi berjalan dengan baik
7.	Melakukan aksi klik pada daftar berita	Klik salah satu list daftar berita	Membuka tautan berita terkait	Komponen aplikasi berjalan dengan baik
<b>Presensi</b>				
1.	Melakukan presensi tanpa koneksi internet	Klik tombol presensi	Memunculkan toast singkat koneksi internet tidak tersedia	Komponen aplikasi dapat berjalan dengan baik
2.	Melakukan presensi diluar waktu yang telah ditentukan	Klik tombol presensi	Muncul dialog presensi ditolak	Komponen aplikasi dapat berjalan dengan baik
3.	Melakukan presensi tanpa izin aplikasi	Klik tombol presensi	Muncul permintaan akses lokasi oleh sistem android	Komponen aplikasi dapat berjalan dengan baik
4.	Melakukan presensi dengan status GPS ponsel mati	Klik tombol presensi	Muncul dialog untuk mengaktifkan GPS ponsel	Komponen aplikasi dapat berjalan dengan baik
5.	Melakukan presensi diluar jarak yang telah ditentukan	Klik tombol presensi	Muncul dialog untuk mendekat ke area presensi.	Komponen aplikasi dapat berjalan dengan baik
6.	Melakukan presensi dengan koneksi internet tersedia,	Klik tombol presensi	Muncul dialog presensi	Komponen aplikasi dapat

	dalam waktu presensi, izin aplikasi, GPS menyala dan dalam lingkup jarak presensi yang diperbolehkan		dapat dilakukan	berjalan dengan baik
7.	Melakukan presensi pada dialog presensi	Klik tombol presensi sekarang	Muncul toast presensi berhasil	Komponen aplikasi dapat berjalan dengan baik
<b>Histori presensi</b>				
1.	Membuka menu presensi aplikasi ketika data dalam database tersedia	Klik navigasi presensi	Muncul data presensi dalam bentuk list yang berisi username, tanggal presensi serta ikon presensi	Komponen aplikasi dapat berjalan dengan baik
2.	Membuka menu presensi aplikasi ketika data dalam database tidak tersedia	Klik navigasi presensi	Muncul gambar yang menandakan bahwa data tidak tersedia	Komponen aplikasi dapat berjalan dengan baik
<b>Foto profil</b>				
1.	Mengganti foto profil pada menu akun aplikasi	Aksi klik pada foto profile	Proses unggah dan mengganti foto profil	Komponen aplikasi dapat berjalan dengan baik
2.	Memperbarui foto profil pada menu akun aplikasi	Aksi klik pada tombol update foto profil	Muncul toast foto profil berhasil diperbarui	Komponen aplikasi dapat berjalan dengan baik
<b>Ganti username</b>				
1.	Ganti username pada menu akun	Aksi klik pada tombol ganti username	Muncul kolom inputan username	Komponen aplikasi berjalan dengan baik
2.	Ganti username dengan kolom inputan dalam keadaan kosong	Username: -	Muncul error pada kolom inputan username	Komponen aplikasi berjalan dengan baik
3.	Ganti username dengan kolom	Username: valid	Muncul toast singkat berisi informasi	Komponen aplikasi berjalan dengan baik



	inputan dalam keadaan terisi		update username	
<b>Ganti email</b>				
1.	Ganti email dengan kolom inputan autentikasi dalam keadaan kosong	Password:-	Muncul error kolom harus di isi pada kolom inputan password	Komponen aplikasi dapat berjalan dengan baik
2.	Ganti email dengan kolom inputan autentikasi dalam keadaan salah	Password: invalid	Muncul error password salah pada kolom inputan password	Komponen aplikasi dapat berjalan dengan baik
3.	Ganti email dengan kolom inputan autentikasi dalam keadaan benar	Password: valid	Muncul kolom inputan mengganti email baru	Komponen aplikasi dapat berjalan dengan baik
4.	Ganti email dengan kolom inputan email dalam keadaan kosong	Email: -	Muncul error email tidak boleh kosong pada kolom inputan email	Komponen aplikasi dapat berjalan dengan baik
5.	Ganti email dengan kolom inputan email disini bukan bertipe email	Email: invalid	Muncul error email tidak valid pada kolom inputan email	Komponen aplikasi dapat berjalan dengan baik
6.	Ganti email dengan kolom inputan email diisi dengan tipe inputan email	Email: valid	Muncul toast singkat ketika email berhasil diganti	Komponen aplikasi dapat berjalan dengan baik
<b>Verifikasi email</b>				
1.	Verifikasi email dengan aksi klik pada tombol verifikasi email	Aksi klik pada tombol verifikasi email	Muncul toast verifikasi email dan mengirim tautan ke email terkait	Komponen aplikasi dapat berjalan dengan baik
<b>Ganti password</b>				
1.	Ganti password dengan kolom inputan autentikasi	Password:-	Muncul error kolom harus di isi pada	Komponen aplikasi dapat

	dalam keadaan kosong		kolom inputan password	berjalan dengan baik
2.	Ganti password dengan kolom inputan autentikasi dalam keadaan salah	Password: invalid	Muncul error password salah pada kolom inputan password	Komponen aplikasi dapat berjalan dengan baik
3.	Ganti password dengan kolom inputan autentikasi dalam keadaan benar	Password: valid	Muncul kolom inputan mengganti password baru	Komponen aplikasi dapat berjalan dengan baik
4.	Ganti password dengan kolom input password dalam keadaan kosong	Password: - konfirmasi password: -	Muncul error password tidak boleh kosong	Komponen aplikasi berjalan dengan baik
5.	Ganti password dengan jumlah karakter kurang dari 8	Password: invalid (< 8 karakter) Konfirmasi password: invalid (< 8 karakter)	Muncul error password kurang dari 8 karakter	Komponen aplikasi dapat berjalan dengan baik
6.	Ganti password dengan kolom inputan password dan konfirmasi password tidak sama	Password: invalid (tidak sama) Konfirmasi password: invalid (tidak sama)	Muncul error password tidak sama	Komponen aplikasi dapat berjalan dengan baik
7.	Ganti password dengan kolom inputan password dan konfirmasi password diisi dengan karakter sama serta dengan jumlah karakter > 8 karakter	Password: valid Konfirmasi password: valid	Muncul toast informasi ubah password dan berpindah ke halaman login aplikasi	Komponen aplikasi dapat berjalan dengan baik
<b>Laporkan masalah aplikasi</b>				
1.	Aksi klik pada tombol laporkan masalah aplikasi	Aksi klik pada tombol laporkan	Berpindah ke aplikasi email	Komponen aplikasi dapat

		masalah aplikasi	didalam sistem android	berjalan dengan baik
<b>Tentang aplikasi</b>				
1.	Aksi klik pada tombol tentang aplikasi	Aksi klik pada tombol tentang aplikasi	Berpindah kedalam halaman tentang aplikasi	Komponen aplikasi dapat berjalan dengan baik
<b>Keluar/Logout</b>				
1.	Aksi klik pada tombol keluar	Aksi klik pada tombol keluar aplikasi	Muncul dialog konfirmasi keluar aplikasi?	Komponen aplikasi dapat berjalan dengan baik
2.	Aksi klik pada tombol “ya” dialog konfirmasi keluar aplikasi	Aksi klik pada tombol “ya” dialog konfirmasi keluar aplikasi	Logout dari akun terkait dan berpindah kedalam halaman login aplikasi	Komponen aplikasi dapat berjalan dengan baik
3.	Aksi klik pada tombol “tidak” dialog konfirmasi keluar aplikasi	Aksi klik pada tombol “tidak” dialog konfirmasi keluar aplikasi	Dialog menghilang dan tetap pada menu akun aplikasi	Komponen aplikasi dapat berjalan dengan baik

Dibawah ini adalah pengujian Whitebox yang dilakukan pada aplikasi, pengujian Whitebox diperlukan untuk menguji lebih dekat tentang detail dari perangkat lunak dengan pendekatan *logical path* pada aplikasi yang akan dilakukan pengujian.

a. Pengujian Whitebox presensi aplikasi

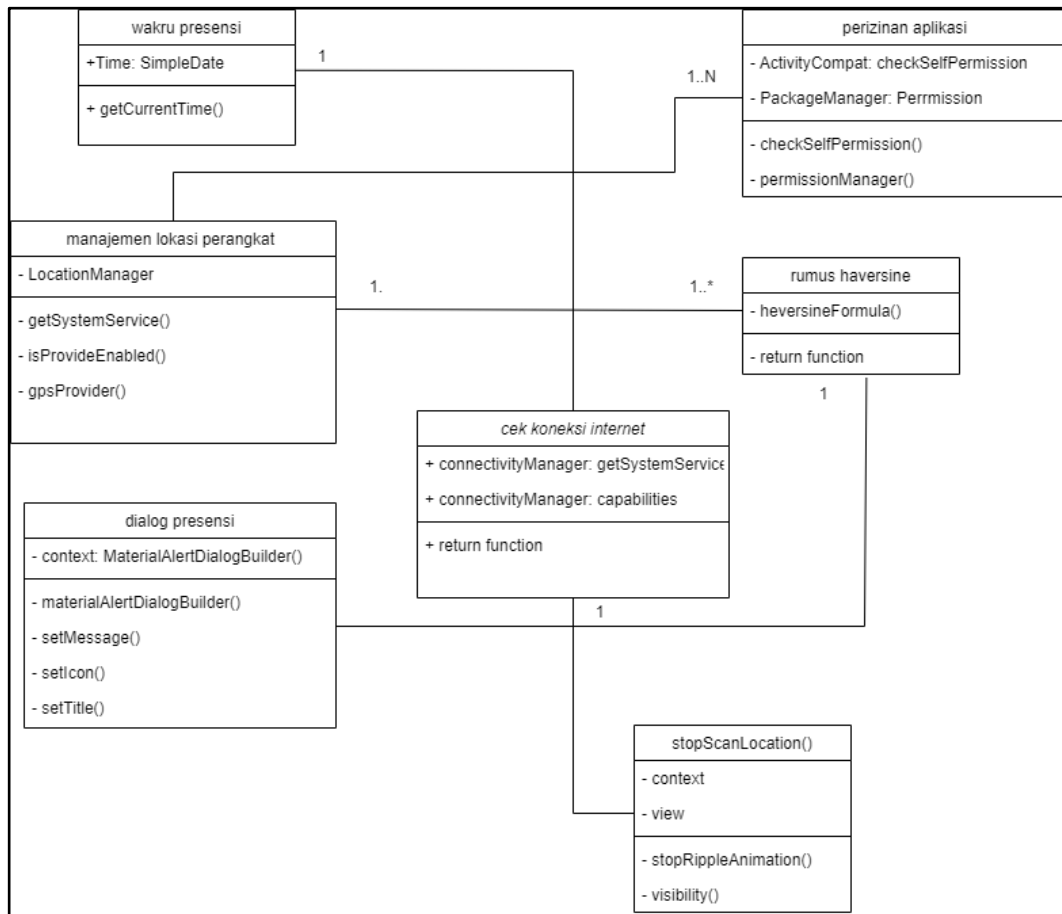
Kasus yang akan diuji dalam pengujian Whitebox testing pada gambar nomor 51 adalah pengujian sistem presensi aplikasi. Gambar 51 berisikan baris kode yang akan di eksekusi oleh aplikasi khusus untuk menangani fitur presensi dan pengukuran jarak menggunakan metode haversine.

```

1 private fun getLastLocation() {
2     if (FunctionLibrary.checkConnection(requireContext())) {
3         if (FunctionLibrary.timeAttendance() || FunctionLibrary.timeAttendanceLate()) {
4             if (checkPermission()) {
5                 if (isLocationEnabled()) {
6                     val locationCallback = object : LocationCallback() {
7                         override fun onLocationResult(locationResult: LocationResult) {
8                             super.onLocationResult(locationResult)
9                             val location = locationResult.lastLocation
10                            val currentLat = location!!.latitude
11                            val currentLong = location.longitude
12                            val destinationLat = getAddress()[0].latitude
13                            val destinationLong = getAddress()[0].longitude
14                            lifecycleScope.launch(Dispatchers.Default) {
15                                val distance = FunctionLibrary.calculateDistance(
16                                    currentLat, currentLong, destinationLat, destinationLong
17                                ) * 1000
18                                Log.d(TAG, "$TAG_RESULT - $distance")
19                                withContext(Dispatchers.Main) {
20                                    if (distance < MEASURING_DISTANCE) {
21                                        showDialog()
22                                        FunctionLibrary.toast(
23                                            context as Activity,
24                                            TOAST_SUCCESS,
25                                            LOCATION_FOUND,
26                                            MotionToastStyle.SUCCESS,
27                                            MotionToast.GRAVITY_BOTTOM,
28                                            MotionToast.LONG_DURATION,
29                                            ResourcesCompat.getFont(
30                                                context as Activity,
31                                                R.font.helvicabold
32                                            )
33                                        )
34                                    } else {
35                                        simpleDialog(
36                                            OUT_OF_RANGE,
37                                            OUT_OF_RANGE_MESSAGE
38                                        )
39                                        binding.tvCheckIn.visibility = View.VISIBLE
40                                    }
41                                }
42                            }
43                            fusedLocationProviderClient?.removeLocationUpdates(this)
44                            stopScanLocation()
45                        }
46                    }
47                    lifecycleScope.launch(Dispatchers.IO) {
48                        fusedLocationProviderClient?.requestLocationUpdates(
49                            locationRequest,
50                            locationCallback,
51                            Looper.getMainLooper()
52                        )
53                    }
54                } else {
55                    simpleDialog(
56                        GPS_STATUS,
57                        GPS_MESSAGE
58                    )
59                    stopScanLocation()
60                }
61            } else {
62                stopScanLocation()
63                requestPermission()
64            }
65        } else {
66            stopScanLocation()
67            simpleDialog(
68                ATTENDANCE_DENIED,
69                ATTENDANCE_TIME
70            )
71        }
72    } else {
73        FunctionLibrary.toast(
74            context as Activity,
75            TOAST_ERROR,
76            PERMISSION_INTERNET,
77            MotionToastStyle.ERROR,
78            MotionToast.GRAVITY_BOTTOM,
79            MotionToast.LONG_DURATION,
80            ResourcesCompat.getFont(context as Activity, R.font.helvicabold)
81        )
82        stopScanLocation()
83    }
84 }

```

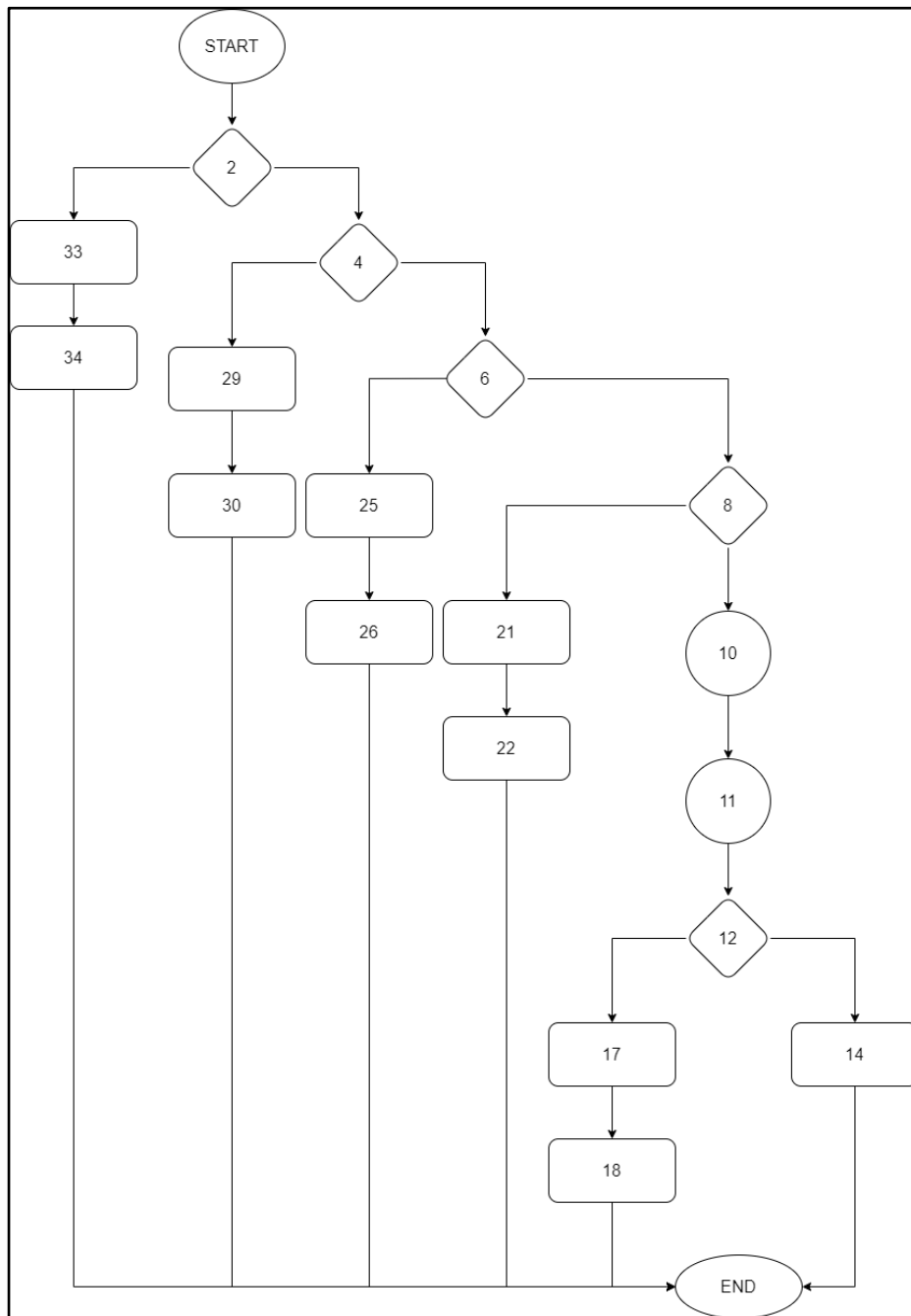
*Gambar 48. Pengujian Whitebox sistem presensi aplikasi*



*Gambar 49. Class diagram fungsi presensi aplikasi*

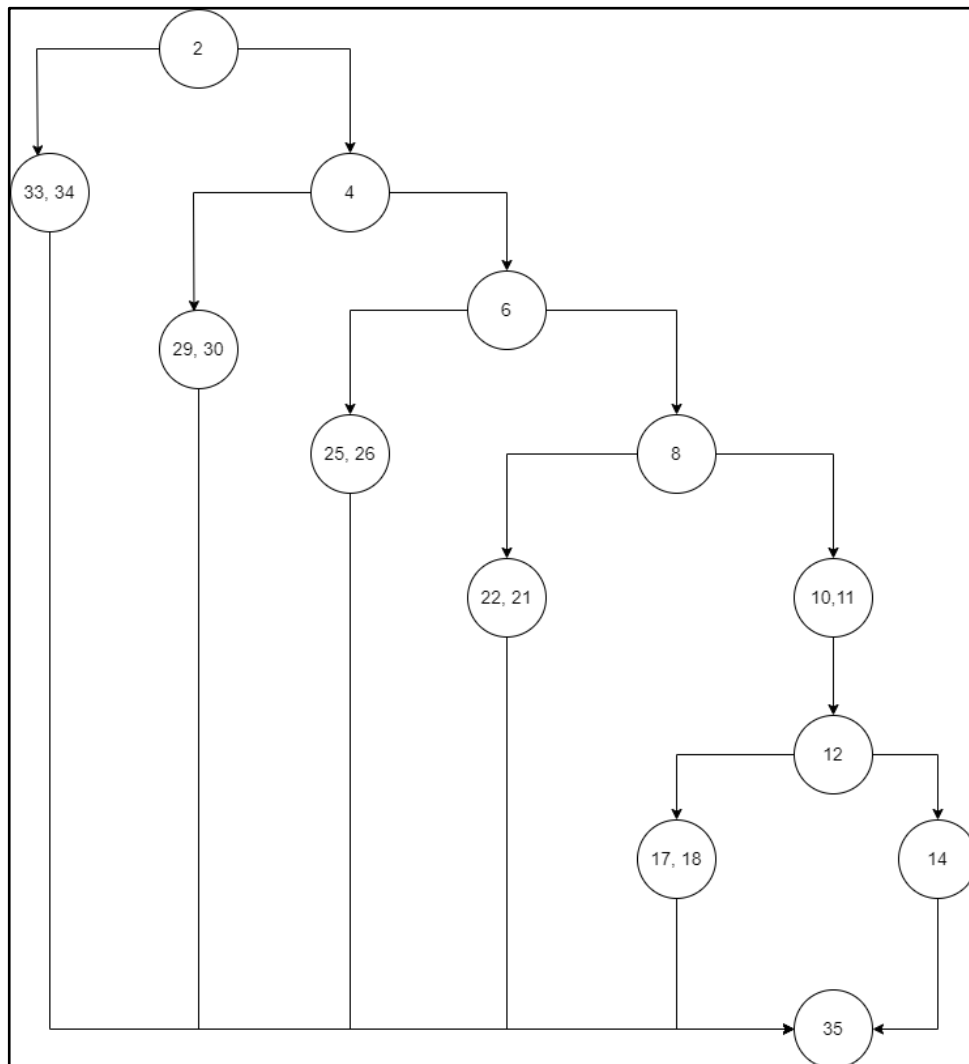
Pada gambar 52, class diagram akan menggambarkan atribut yang digunakan aplikasi untuk membuat fitur presensi dan pengukuran jarak, class diagram diatas juga merupakan fungsi yang digunakan dan keterkaitan antar fungsi yang terjadi.

Flow chart dari pengujian Whitebox pada gambar 52 adalah sebagai berikut:



*Gambar 50. Flowchart whitebox testing*

Sedangkan untuk flowgraph dari pengujian Whitebox terdapat pada gambar 54 dibawah ini



*Gambar 51. Flowgraph whiteBox testing sistem presensi*

Dengan adanya flowgraph, maka Cyclomatic Complexity dapat dilakukan dengan rumus sebagai berikut:

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 35 - 12 + 2 \\
 &= 25
 \end{aligned}$$

Dengan begitu, jumlah region dapat ditentukan dengan jumlah region keseluruhan adalah 25. Untuk menentukan independent path, dapat melihat flowgraph yang ada digambar 53, independent graph yang terbentuk berdasarkan flowgraph adalah sebagai berikut:

Path 1: 2, 33, 34, 35

Path 2: 2, 4, 29, 30, 35

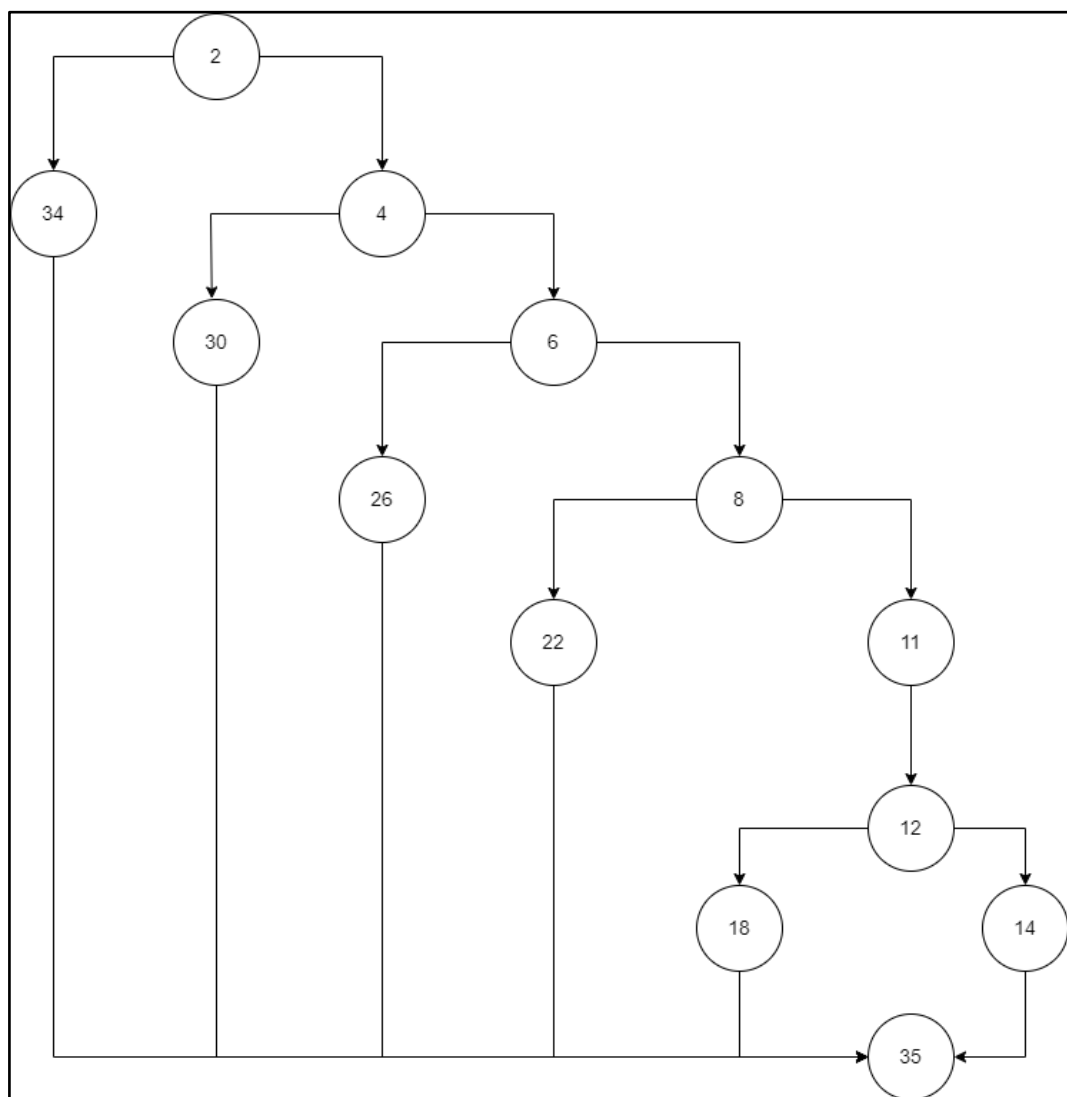
Path 3: 2, 4, 6, 25, 26, 35

Path 4: 2, 4, 6, 8, 22, 21, 35

Path 5: 2, 4, 6, 8, 10, 11, 12, 17, 18, 35

Path 6: 2, 4, 6, 8, 10, 11, 12, 14, 35

Untuk mendapatkan gambaran yang lebih jelas tentang graph matriks, maka perlu dibuat konversi dari flowgraph seperti berikut:



*Gambar 52. Konversi flowgraph Whitebox testing*

Tahapan selanjutnya dalam menentukan predicate node adalah membuat tabel matriks sesuai dengan konversi graph pada gambar nomor 54, detail dari tabel matriks adalah sebagai berikut:



Pada tabel lampiran nomor 1, didapatkan predicate node dengan jumlah 5 node, dengan begitu, predicate node dapat dihitung dengan rumus:

$$\begin{aligned} V(G) &= P + 1 \\ &= 5 + 1 \\ &= 6 \end{aligned}$$

Dengan predicate node yang terdeteksi adalah matriks 2, 4, 6, 8, 12

### 3.3 Hasil Penelitian

Tahapan selanjutnya setelah implementasi aplikasi adalah uji coba system, untuk mendapatkan hasil uji coba yang akurat sesuai dengan metode yang telah ada sebelumnya, maka uji coba aplikasi akan menggunakan metode *System Usability Scale* (SUS), secara singkat, metode usability merupakan sebuah analisa yang bersifat kualitatif atau berorientasi pada kualitas pada objek yang diujikan, dengan metode usability ini akan menentukan seberapa mudah antarmuka aplikasi ketika digunakan oleh pengguna, salah satu indikator antarmuka aplikasi mudah digunakan antara lain adalah ketika semua fungsi yang terdapat didalam aplikasi efektif, memuaskan dan defisien ketika digunakan, konklusi dari usability testing adalah menentukan atau melakukan evaluasi antarmuka aplikasi apakah telah memenuhi kebutuhan pengguna atau belum memenuhi dan perlu diadakan perbaikan [14] dan [15].

Table 4. Kuisioner System Usability Scale (SUS)

No	Pertanyaan	Rata-rata	Kategori
1	Saya merasa akan sering menggunakan aplikasi peduliPresensi	4.05	setuju
2	Saya menilai bahwa aplikasi peduliPresensi tidak perlu serumit ini	2.4	Tidak setuju
3	Saya merasa aplikasi peduliPresensi mudah untuk digunakan	4.2	setuju
4	Saya membutuhkan pendampingan teknis dalam menggunakan aplikasi ini	2.8	Netral
5	Saya menilai fitur yang terdapat pada aplikasi PeduliPresensi dapat berjalan dengan baik	2.8	Netral
6	Saya menilai banyak hal pada aplikasi PeduliPresensi yang tidak konsisten	2.4	Tidak setuju
7	Saya merasa bahwa orang lain dapat mempelajari aplikasi PeduliPresensi dengan cepat	4.15	Setuju
8	Saya menilai aplikasi PeduliPresensi ssangat rumit ketika digunakan	2.2	Tidak setuju

9	Saya tidak menemui hambatan dalam penggunaan aplikasi	3.85	Setuju
10	Saya perlu untuk beradaptasi dalam menggunakan aplikasi	2.2	Tidak setuju

### 3.4 Pembahasan

Pembahasan hasil penelitian dilakukan dengan melakukan kalkulasi nilai yang terdapat pada tiap kuisioner yang telah di isi oleh pengguna.

*Table 5. Penilaian kuisioner SUS*

No	Jawaban	Skor
1	Sangat tidak setuju	1
2	Tidak setuju	2
3	Ragu-ragu	3
4	Setuju	4
5	Sangat setuju	5

Aturan perhitungan jawaban pada kuisioner System Usability Scale (SUS) adalah sebagai berikut:

1. Setiap pertanyaan yang bernomor ganjil, penilaian dilakukan dengan melakukan operasi pengurangan terhadap skor yang didapat dari pengguna dengan angka 1.
2. Setiap pertanyaan yang bernomor genap, penilaian dilakukan dengan melakukan operasi pengurangan angka 5 terhadap skor yang didapat dari pengguna.
3. Sedangkan untuk mendapatkan hasil akhir dari SUS didapatkan dari hasil penjumlahan skor setiap pertanyaan dan dikalikan dengan 2.5

Untuk mendapatkan skor akhir dari SUS, maka dilakukan perhitungan dengan rumus

$$\bar{x} = \frac{\sum x}{n}$$

$\bar{x}$  = skor rata-rata  
 $\sum x$  = jumlah skor SUS  
 $n$  = jumlah responden

*Gambar 53. Rumus skor rata-rata SUS*

Dengan begitu, semua data yang didapatkan dari hasil kuisioner akan dimasukan dalam tabel dan dilakukan kalkulasi sesuai dengan ketentuan dari metode SUS.

*Table 6. Tabel data asli responden*

No	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Responden 1	4	2	4	3	5	4	4	2	5	2
2	Responden 2	3	3	4	4	4	3	3	2	2	1
3	Responden 3	4	3	4	4	5	2	4	2	4	3
4	Responden 4	3	3	4	4	5	3	4	3	4	2
5	Responden 5	4	3	3	4	3	4	2	4	3	2
6	Responden 6	3	4	5	3	4	3	5	2	5	3
7	Responden 7	4	3	4	3	4	2	5	3	5	1
8	Responden 8	4	2	4	2	4	2	4	2	4	3
9	Responden 9	4	4	4	4	4	3	4	3	3	3
10	Responden 10	4	3	4	2	4	4	4	2	4	2
11	Responden 11	4	2	5	4	4	2	3	3	2	2
12	Responden 12	4	2	5	3	4	2	4	2	4	2
13	Responden 13	4	1	3	3	5	2	5	2	4	3
14	Responden 14	5	2	4	2	5	2	5	1	3	3
15	Responden 15	5	2	5	2	4	2	4	2	5	3
16	Responden 16	4	2	4	2	5	2	5	2	4	2
17	Responden 17	4	2	4	2	4	2	4	2	4	2
18	Responden 18	5	1	5	1	4	1	4	1	4	1
19	Responden 19	4	2	4	2	5	2	5	2	4	2
20	Responden 20	5	2	5	2	5	1	5	2	4	2

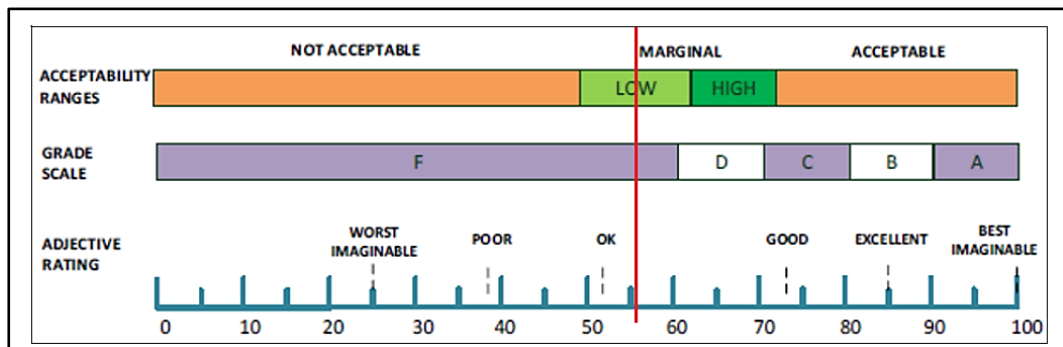
Data yang didapatkan pada tabel 10 selanjutnya akan dilakukan perhitungan dengan rumus SUS dan hasilnya akan ditampilkan pada tabel 11.

*Table 7. Tabel hasil perhitungan dengan rumus SUS*

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Nilai (Jml x 2.5)
3	3	3	2	4	1	3	3	4	3	29	73
2	2	3	1	3	2	2	3	1	4	23	58
3	2	3	1	4	3	3	3	3	2	27	68
2	2	3	1	4	2	3	2	3	3	25	63
3	2	2	1	2	1	1	1	2	3	18	45
2	1	4	2	3	2	4	3	4	2	27	68
3	2	3	2	3	3	4	2	4	4	30	75
3	3	3	3	3	3	3	3	3	2	29	73
3	1	3	1	3	2	3	2	2	2	22	55
3	2	3	3	3	1	3	3	3	3	27	68

3	3	4	1	3	3	2	2	1	3	25	63
3	3	4	2	3	3	3	3	3	3	30	75
3	4	2	2	4	3	4	3	3	2	30	75
4	3	3	3	4	3	4	4	2	2	32	80
4	3	4	3	3	3	3	3	4	2	32	80
3	3	3	3	4	3	4	3	3	3	32	80
3	3	3	3	3	3	3	3	3	3	30	75
4	4	4	4	3	4	3	4	3	4	37	93
<b>Skor rata-rata (Hasil Akhir)</b>											<b>72</b>

Berdasarkan akhir dari tabel 11, didapatkan skor rata-rata nilai 72, untuk mendapatkan konklusi yang lebih jelas, maka dibutuhkan analisa skor SUS.



Gambar 54. tingkat penerimaan metode SUS

Dari gambar 54 dan hasil akhir yang diperoleh, maka metode SUS pada aplikasi peduli presensi mendapatkan tingkat penerimaan “MARGINAL HIGH”, dengan adjective rating “GOOD” [16].

<b>Grade</b>	<b>SUS</b>	<b>Percentile range</b>
<i>A+</i>	84.1 – 100	96 – 100
<i>A</i>	80.8 – 84.0	90 – 95
<i>A-</i>	78.9 – 80.7	85 – 89
<i>B+</i>	77.2 – 78.8	80 – 84
<i>B</i>	74.1 – 77.1	70 – 79
<i>B-</i>	72.6 – 74.0	65 – 69
<i>C+</i>	71.1 – 72.5	60 – 64
<i>C</i>	65.0 – 71.0	41 – 59
<i>C-</i>	62.7 – 64.9	35 – 40
<i>D</i>	51.7 – 62.6	15 – 34
<i>F</i>	0 – 51.6	0 – 14

*Gambar 55. Curved Grade Scale metode SUS*

Sedangkan untuk CGS atau Curved Grade Scale, aplikasi peduli presensi mendapatkan grade B dengan percentile range 72. Berdasarkan penilaian menggunakan metode SUS, maka pengembangan dan perbaikan pada sisi fungsionalitas aplikasi perlu untuk ditingkatkan, selain itu karena aplikasi ini terbilang baru pada lingkungan kantor balai desa Warureja, perlu juga dibuat sosialisasi mengenai penggunaan aplikasi baik berupa video tutorial ataupun *manual book* tentang tata cara penggunaan dan aplikasi bekerja.

## 4. KESIMPULAN DAN SARAN

### 4.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan penulis terkait dengan Pengembangan aplikasi presensi pegawai berbasis Android menggunakan algoritma pengukur jarak haversine di kantor Balai Desa Warureja Kab. Tegal, maka dapat diperoleh kesimpulan sebagai berikut:

1. Sistem presensi berbasis Android pada balai desa Warureja telah selesai dibangun.
2. Metode yang digunakan untuk mengukur tingkat penerimaan pengguna terhadap aplikasi PeduliPresensi yaitu System Usability Scale (SUS). Hasil survei menunjukkan aplikasi PeduliPresensi mendapatkan tingkat penerimaan dengan skala Marginal High dan adjective rating Good
3. Metode pengukur jarak yang digunakan dalam aplikasi PeduliPresensi adalah metode *haversine*. Dengan menggunakan metode *haversine*, karyawan harus benar-benar berada di area tertentu di sekitar kantor dan dalam waktu yang telah diizinkan.
4. Dengan dikembangkan aplikasi PeduliPresensi, diharapkan kedisiplinan karyawan terkait presensi dapat ditingkatkan.

### 4.2 Saran

Berikut merupakan saran yang dapat penulis sampaikan untuk pengembangan aplikasi pada penelitian selanjutnya, detail saran adalah sebagai berikut:

1. Penggunaan firebase sebagai database utama aplikasi akan sangat mempercepat proses pembuatan presensi karena firebase sendiri merupakan database berbasis NoSql, sehingga skalabilitas serta performa aplikasi dapat terjaga secara optimal, namun menggabungkannya dengan database SQL akan jauh lebih menguntungkan terutama pada saat pengelolaan data yang terdapat didalam database untuk keperluan konversi data, menggunakan database SQL akan jauh lebih mudah.
2. Menambahkan fitur presensi saat kehadiran pegawai merupakan hal yang wajar, namun akan lebih baik jika aplikasi dilengkapi dengan presensi pulang pegawai dan izin cuti, dengan begitu aplikasi akan semakin kompleks dan memudahkan pengguna serta pihak kantor dalam melakukan pemantauan kinerja karyawan terlebih dalam hal kehadiran kerja.

3. Melakukan survei terkait kebutuhan pengguna terhadap aplikasi PeduliPresensi, guna menampung masukan-masukan dari pengguna agar mendapatkan tingkat penerimaan 'Acceptable' dari metode System Usability Scale (SUS).

## DAFTAR PUSTAKA

- [1] A. Rahmatulloh and R. Gunawan, "Sistem Presensi Berbasis Sidik Jari untuk Peningkatan Disiplin Kehadiran Aparat Desa," *JATI EMAS (Jurnal Apl. Tek. dan Pengabd. Masyarakat)*, vol. 3, no. 1, p. 14, 2019, doi: 10.36339/je.v3i1.183.
- [2] N. Qois and Y. Jumaryadi, "Implementasi Location Based Service pada Sistem Informasi Kehadiran Pegawai Berbasis Android," vol. 10, pp. 550–561, 2021.
- [3] J. Christian and H. Nasrullah, "PEMANFAATAN RADIO FREQUENCY IDENTIFICATION (RFID) UNTUK SISTEM ABSENSI PEGAWAI," *Budi Luhur Inf. Technol.*, vol. 10, no. 1, Jan. 2018, Accessed: Oct. 28, 2021. [Online]. Available: <https://journal.budiluhur.ac.id/index.php/bit/article/view/509>.
- [4] D. Supriatna and E. Junianto, "Aplikasi Presensi Pegawai Memanfaatkan Teknologi Fingerprint dan Global Positioning System (GPS) Pada Android," *e Pros. Tek. Inform.*, vol. 1, no. 1, pp. 269–280, 2020.
- [5] E. Mulyadi, A. Trihariprasetya, and I. G. Wiryawan, "Penerapan Sistem Presensi Mobile Dengan Menggunakan Sensor Gps (Klinik Pratama X Di Jember)," *J. Nas. Pendidik. Tek. Inform.*, vol. 9, no. 1, p. 11, 2020, doi: 10.23887/janapati.v9i1.23174.
- [6] . I. A. T., "ANALISIS TINGKAT AKURASI GLOBAL POSITIONING SYSTEM SMARTPHONE DALAM MENENTUKAN TITIK LOKASI PADA GOOGLE MAP," *J. Tek. Elektro Univ. Tanjungpura*, vol. 1, no. 1, Jan. 2018, Accessed: Nov. 02, 2021. [Online]. Available: <https://jurnal.untan.ac.id/index.php/jteuntan/article/view/23426>.
- [7] A. P. ALDYA, "HAVERSINE FORMULA UNTUK MEMBATASI JARAK PADA APLIKASI PRESENSI ONLINE," *J. INSTEK (Informatika Sains dan Teknol.*, vol. 4, no. 2, pp. 171–180, Aug. 2019, doi: 10.24252/INSTEK.V4I2.10042.
- [8] A. Rahmatulloh, "IMPLEMENTASI FORMULA HAVERSINE DAN KOMUNIKASI DATA REAL-TIME MENGGUNAKAN WEBSOCKET DI SISTEM PENGAWASAN WARGA NEGARA ASING," *KLIK - Kumpul. J. ILMU Komput.*, vol. 6, no. 2, p. 143, 2019, doi: 10.20527/klik.v6i2.210.
- [9] P. Metode Perhitungan *et al.*, "PERBANDINGAN METODE PERHITUNGAN JARAK EUCLIDEAN, HAVERSINE, DAN MANHATTAN DALAM PENENTUAN POSISI KARYAWAN," *J. Tekno Insentif*, vol. 14, no. 2, pp. 69–77, Aug. 2020, doi: 10.36787/JTI.V14I2.270.
- [10] T. Sinta Peringkat *et al.*, "PEMANFAATAN APLIKASI MOBILE



BERBASIS IOS DALAM MENUNJANG USAHA WEDDING ORGANIZER,” *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 5, no. 2, pp. 251–258, Feb. 2020, doi: 10.33480/JITK.V5I2.1154.

- [11] “Membuat Struktur Database Anda | Firebase Documentation.” <https://firebase.google.com/docs/database/android/structure-data?hl=id> (accessed Jun. 29, 2022).
- [12] M. Pratama and V. I. Delianti, “RANCANG BANGUN APLIKASI PRESENSI DENGAN GLOBAL POSITIONING SYSTEM (GPS) BERBASIS ANDROID (Studi Kasus : PT. Perkebunan Nusantara VI Unit Usaha Kayu Aro),” *INCARE, Int. J. Educ. Resour.*, vol. 2, no. 2, pp. 141–154, Aug. 2021, Accessed: Dec. 11, 2021. [Online]. Available: <http://ejournal.ijshs.org/index.php/incare/article/view/251>.
- [13] I. R. Salamah and F. Ismawan, “PENERAPAN USER CENTERED DESIGN (UCD) PADA WIREFRAME DESAIN USER INTERFACE DAN USER EXPERIENCE APLIKASI SINOPSIS FILM,” vol. 7, no. 1, pp. 75–79, 2021.
- [14] R. Firmansyah, “Usability Testing Dengan Use Questionnaire Pada Aplikasi Sipolin Provinsi Jawa Barat,” *Swabumi*, vol. 6, no. 1, pp. 1–7, 2018, doi: 10.31294/swabumi.v6i1.3310.
- [15] Z. Sharfina and H. B. Santoso, “An Indonesian adaptation of the System Usability Scale (SUS),” *2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSYS 2016*, pp. 145–148, Mar. 2017, doi: 10.1109/ICACSYS.2016.7872776.
- [16] D. Supriyadi, S. Thyas Safitri, and D. Y. Kristiyanto, “Higher Education e-Learning Usability Analysis Using System Usability Scale,” *Int. J. Inf. Syst. Technol. Akreditasi*, vol. 4, no. 1, pp. 436–446, 2020.