

## AbsenLoc: Aplikasi Absensi *Mobile* Berbasis Lokasi

Jemy Agung Pribadi<sup>#1</sup>, Nina Setiyawati<sup>\*2</sup>

<sup>#</sup>Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana

Jl. Diponegoro No. 52 – 60, Salatiga, Indonesia 50711

<sup>1</sup>672014104@student.uksw.edu

<sup>\*</sup>Staf Pengajar Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana

Jl. Diponegoro No. 52 – 60, Salatiga, Indonesia 50711

<sup>2</sup>nina.setiyawati@uksw.edu

### Abstrak

Pada beberapa perusahaan, banyak acara diselenggarakan dan karyawan wajib untuk menghadirinya. Akan tetapi absensi manual ataupun menggunakan alat pembaca *fingerprnt* dirasa tidak efisien dikarenakan akan menghabiskan banyak waktu untuk mengantre. Untuk mengatasi hal tersebut pada penelitian ini dilakukan pembangunan aplikasi AbsenLoc yang diharapkan dapat melakukan absensi secara efektif dan efisien. Pembangunan AbsenLoc dilakukan menggunakan Android Studio dengan bahasa pemrograman Java dan Firestore sebagai *database*. Adapun alur pembangunan dilakukan berdasarkan pada metode *Research and Development*. AbsenLoc menggunakan layanan *Location Based Service* untuk memvalidasi kehadiran dalam suatu acara, di mana *Location Based Service* merupakan layanan yang digunakan untuk mengetahui posisi dari pengguna yang kemudian menggunakan informasi tersebut untuk menyediakan jasa dan aplikasi yang personal. Validasi AbsenLoc dilakukan dalam dua tahap. Pertama, dengan mencocokkan lokasi peserta dengan lokasi acara yang telah ditentukan penyelenggara. Kedua, dengan mencocokkan lokasi dari swafoto memanfaatkan kamera pada *smartphone*. Pengujian AbsenLoc dilakukan dengan dua metode yaitu *Blackbox testing* dan uji pengguna. Berdasarkan *Blackbox testing* didapatkan bahwa fungsi, masukan dan keluaran dari aplikasi AbsenLoc sudah sesuai dengan spesifikasi yang dibutuhkan, sedangkan berdasarkan pengujian terhadap pengguna, didapatkan bahwa aplikasi AbsenLoc dapat memberikan proses validasi absensi yang efektif, mengurangi waktu antrean dan membantu perusahaan dalam mengelola pengarsipan absensi.

**Kata kunci:** Absensi, *Mobile*, *Location Based Service*, Metode *Research and Development*, *Blackbox Testing*

## AbsenLoc: Location-Based Mobile Attendance Application

### Abstract

In some companies, many events are held and employees are obliged to attend. However, manual attendance or using a fingerprint reader is considered inefficient because it will spend a lot of time waiting in line. In order to overcome this problem, this research conducted the development of the AbsenLoc application which is expected to be able to perform attendance effectively and efficiently. AbsenLoc development is carried out using Android Studio with the Java programming language and Firestore as the database. The development flow is carried out based on the Research and Development method. AbsenLoc uses Location Based Service to validate attendance at an event, where Location Based Service is a service used to find out the position of the user who then uses the information to provide personalized services and applications. The AbsenLoc validation is carried out in two stages. First, by matching the participant's location with the event location determined by the organizer. Second, by matching the location of the selfie using the camera on a smartphone. The AbsenLoc test is carried out by two methods, namely Blackbox testing and user testing. Based on Blackbox testing, it was found that the functions, input and output of the AbsenLoc application were in accordance with the required specifications, while based on testing of users, it was found that the AbsenLoc application can provide an effective attendance validation process, reduce queue time and help companies manage attendance filing.

**Keywords:** Absenteeism, Mobile, Location Based Service, Research and Development method, Blackbox Testing

### I. PENDAHULUAN

*Location Based Service* (LBS) adalah sebuah layanan yang digunakan untuk mengetahui posisi dari pengguna [1], kemudian menggunakan informasi tersebut untuk

meyediakan jasa dan aplikasi yang personal [2]. Dalam penggunaan layanan ini, salah satu unsur utamanya adalah pengguna dapat menentukan lokasinya, melacak

gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan [3].

Pada penelitian ini LBS diterapkan sebagai layanan untuk memvalidasi kehadiran dalam suatu acara, yaitu aplikasi AbsenLoc. Pada beberapa perusahaan, banyak acara diselenggarakan dan karyawan wajib untuk menghadirinya. Akan tetapi absensi manual ataupun menggunakan alat pembaca *fingerprind* dirasa tidak efisien dikarenakan akan menghabiskan banyak waktu untuk mengantre. Dengan aplikasi AbsenLoc berbasis *mobile* Android, *platform* yang memiliki *market share global* paling tinggi beberapa tahun terakhir [4], peserta acara bisa langsung melakukan absensi tanpa perlu mengantre.

Validasi AbsenLoc dilakukan dalam dua tahap. Pertama, aplikasi memverifikasi lokasi *smartphone* dari peserta acara, di mana *Global Positioning System* (GPS) pada *smartphone* akan melacak posisi dan menyediakan pengaturan waktu pengguna [5]. Kedua, aplikasi memverifikasi lokasi peserta acara dari informasi lokasi swafoto yang harus dilakukan melalui aplikasi AbsenLoc dengan mengakses kamera *smartphone*. Adapun kedua verifikasi tersebut akan dicocokkan dengan jangkauan *latitude* dan *longitude* yang telah ditetapkan oleh penyelenggara acara. Jika lokasi peserta acara di luar jangkauan tersebut, maka absensi tidak akan terekam di AbsenLoc.

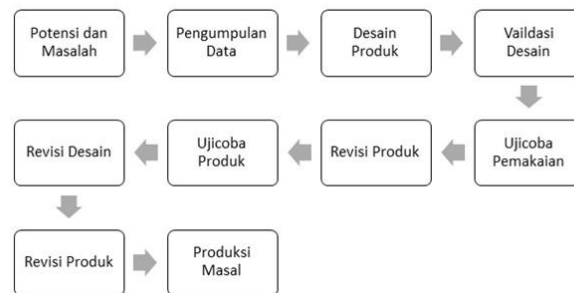
Selain proses validasi yang efektif, AbsenLoc juga memberikan laporan kehadiran yang lengkap. Hal ini akan membantu perusahaan dalam mengelola pengarsipan absensi di mana dapat menghasilkan informasi yang membantu dalam pengambilan keputusan.

Pada penelitian ini aplikasi AbsenLoc dibangun menggunakan Android Studio dan *Firestore* sebagai *database*. Android Studio merupakan *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA [6]. Sedangkan *Firestore* adalah *database* terbaru dari *Firebase* untuk pengembangan aplikasi seluler. *Database* ini melanjutkan keberhasilan *Realtime Database* dengan model data baru yang lebih intuitif [7]. *Google Maps API* adalah sebuah layanan (*service*) yang diberikan oleh *Google* kepada para pengguna untuk memanfaatkan *Google Maps* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi *enterprise* di dalam website [8]. Dalam kasus ini *Google maps API* digunakan untuk memunculkan peta dan marker.

## II. METODOLOGI

Pengembangan aplikasi dilakukan berdasarkan tahapan metode *Research and Development* (RnD) seperti terlihat pada Gambar 1.

Berdasarkan Gambar 1, pengembangan aplikasi dilakukan dalam beberapa tahapan seperti yang dikemukakan oleh Sugiyono [9], yaitu:



Gambar 1. Tahapan Metode *Research and Development* (RnD) [9]

1) *Potensi dan Masalah*: Pada tahap ini dilakukan identifikasi adanya potensi dan masalah. Potensi yang ada yaitu kebutuhan akan absensi kehadiran karyawan atau pegawai pada acara-acara yang diwajibkan. Akan tetapi sering muncul permasalahan yaitu tidak efisiennya absensi menggunakan perangkat pembaca *fingerprind*.

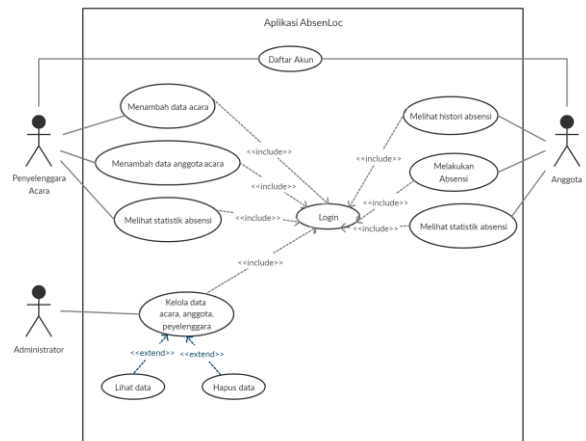
2) *Pengumpulan data*: Setelah potensi dan masalah dapat ditunjukkan secara faktual, langkah selanjutnya yaitu mengumpulkan berbagai informasi yang didapatkan studi literatur dan wawancara kepada pemangku kebijakan dari perusahaan. Dari wawancara yang dilakukan didapatkan bahwa banyak karyawan menggunakan *smartphone* android, maka dari itu direncanakan pembangunan aplikasi absensi berbasis android. Selain itu, dibutuhkan suatu mekanisme validasi lokasi karyawan ketika melakukan absensi agar karyawan hanya bisa melakukan absensi hanya dalam *range* lokasi yang telah ditentukan berdasarkan titik *longitude* dan *latitude*.

*Desain Produk*: Tahap ini adalah hasil akhir dari serangkaian penelitian awal yang dapat berupa rancangan kerja baru atau produk baru. Tahap ini diawali dengan perancangan aplikasi menggunakan UML (Unified Modelling Language) [10] untuk menghasilkan gambaran dasar dari aplikasi AbsenLoc. Adapun salah satu diagram yang dibangun adalah use case diagram yang menjelaskan secara visual konteks dari interaksi antara aktor dengan system [11]. Berikut adalah use case diagram aplikasi AbsenLoc yang dapat dilihat pada Gambar 2.

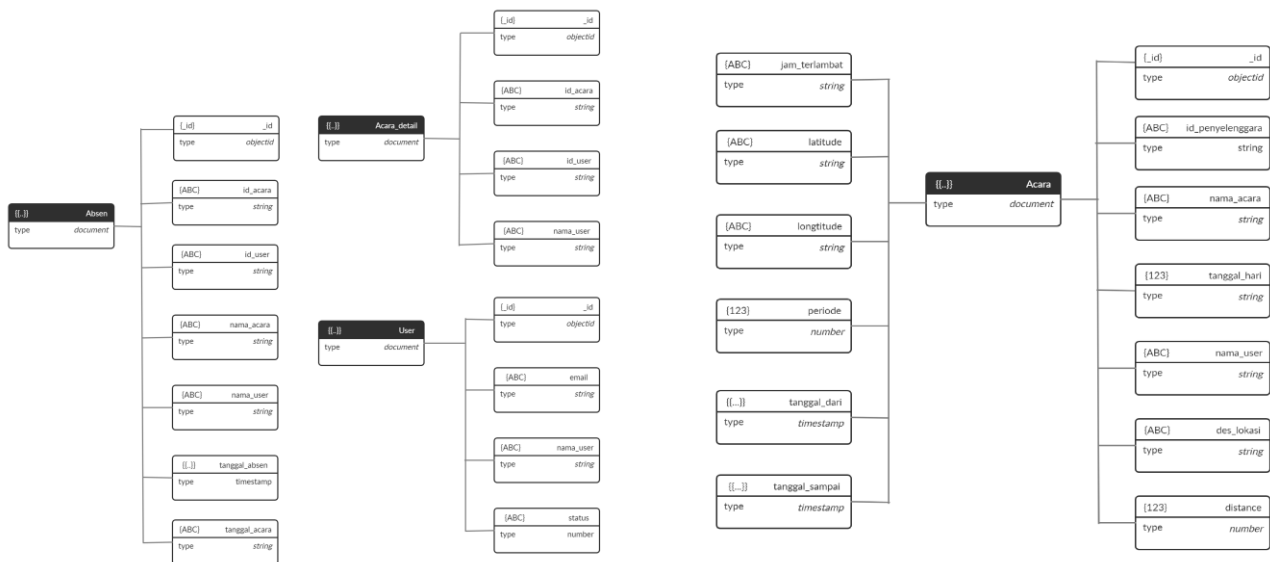
Gambar 2 menjelaskan proses bisnis aplikasi AbsenLoc yang terdiri dari 3 aktor yaitu administrator, penyelenggara acara dan anggota. Penyelenggara acara dapat melakukan pendaftaran akun dan harus melakukan proses *login* terlebih dahulu untuk dapat mengakses fungsi yang terdapat pada sistem, hal tersebut juga berlaku untuk aktor anggota. Penyelenggara dapat menambahkan data acara dengan menentukan lokasi acara kemudian mengisi *field* acara yang meliputi nama acara, deskripsi lokasi, tanggal acara, jam mulai, jam selesai, keterlambatan dan periode acara. Setelah acara berhasil ditambahkan, penyelenggara dapat menambahkan anggota yang akan didaftarkan pada setiap acara yang telah ditambahkan. Aktor anggota dapat melakukan absensi dengan waktu acara harus pada rentang yang telah ditentukan oleh penyelenggara acara dan lokasi anggota harus pada

jangkauan lokasi acara, anggota juga dapat melihat *history* absensi dan rekap data absensi pada setiap acaranya. Aktor Admin dapat melakukan pengelolaan data acara, anggota dan penyelenggara. Adapun desain *database* aplikasi AbsenLoc yang ada pada Gambar 3.

Gambar 3 merupakan desain *database* aplikasi AbsenLoc menggunakan data *modelling* NoSQL memanfaatkan *firestore* [12]. Pada data *modelling* NoSQL tabel berubah namanya menjadi *document* [13]. Dari Gambar 3 dapat dijelaskan bahwa terdapat empat tipe *document* yaitu *document user*, acara, acara detail dan absen. *Document user* digunakan untuk menyimpan data *user* yaitu *user* penyelenggara, anggota dan administrator. *Document* acara digunakan untuk menyimpan data acara yang diinputkan oleh penyelenggara. *Document* acara detail digunakan untuk menyimpan data anggota pada setiap acara. Sedangkan *document* absen digunakan untuk menyimpan data anggota yang sudah melakukan absensi.



Gambar 2. Use Case Diagram AbsenLoc



Gambar 3. Desain *database* Aplikasi AbsenLoc

3) *Validasi Desain*: Dalam proses ini desain yang sudah dibuat dievaluasi, akan tetapi hanya berdasarkan pemikiran rasional dan dilakukan secara internal.

4) *Revisi Desain Produk*: Berdasarkan validasi yang dilakukan pada tahap sebelumnya, didapatkan beberapa revisi seperti desain antar muka, dan jenis laporan yang diberikan. Dari revisi yang ada, produk dibangun menggunakan bahasa pemrograman *Java* pada *Android Studio* dan *Firestore* untuk *database*.

5) *Uji Coba Produk*: Pada tahap ini dilakukan uji coba aplikasi AbsenLoc untuk melihat apakah aplikasi sudah berjalan sesuai yang diharapkan.

6) *Revisi Produk*: Pada tahap ini dilakukan revisi jika pada tahap sebelumnya ditemukan hal-hal yang harus diubah atau ditambahkan pada aplikasi.

7) *Uji Coba Pemakaian*: Pada tahap ini dilakukan uji coba aplikasi AbsenLoc dalam kondisi yang

sesungguhnya, yaitu diuji pada beberapa acara. Dari tahap ini ditemukan kekurangan, yaitu tidak kompatibel pada beberapa *device*, sehingga beberapa anggota tidak dapat melakukan absensi.

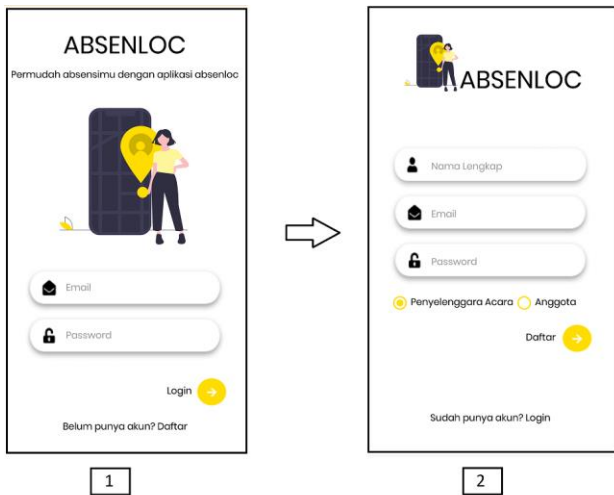
8) *Revisi Produk*: Pada tahap ini dilakukan revisi atau perbaikan dari kelemahan yang ditemukan pada uji coba pemakaian.

9) *Produksi Masal*: Aplikasi AbsenLoc tidak/belum diproduksi secara masal tetapi sudah diunggah di *Google Play Store*.

### III. HASIL DAN PEMBAHASAN

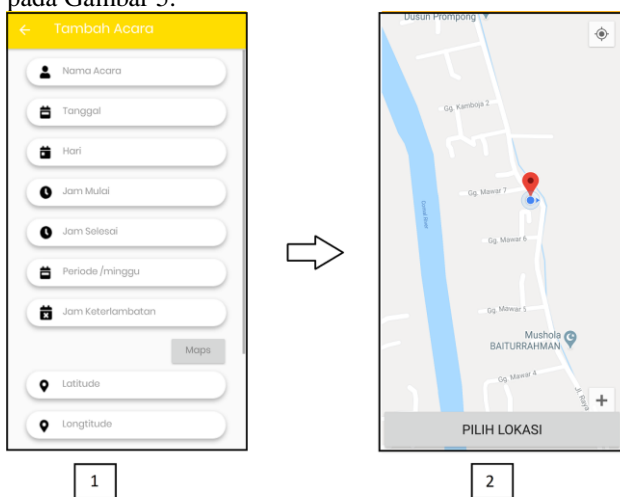
Hasil penelitian ini merupakan aplikasi absensi berbasis *Android* yang bisa digunakan oleh semua aktor. Terdapat 3 aktor yaitu penyelenggara, anggota, dan aktor administrator. Aplikasi AbsenLoc hanya dapat berjalan di *Sistem Operasi Android* dengan *Minimal* versi *Android*

adalah *Nugget*. Gambar 4 menunjukkan tampilan *login* dan pendaftaran.



Gambar 4. Tampilan *Login* (1) dan Pendaftaran (2)

Gambar 4 merupakan tahap pertama dalam menggunakan aplikasi AbsenLoc, penyelenggara dan anggota harus melakukan pendaftaran akun sedangkan untuk administrator tidak perlu melakukan pendaftaran karena akun administrator sudah terdaftar. Penyelenggara dan anggota harus mengisi *form* nama lengkap, email, *password* dan memilih *role* penyelenggara acara atau anggota, kemudian daftar. Setelah akun berhasil didaftarkan kemudian masuk ke halaman *login* dan memasukkan email dan *password* yang sudah didaftarkan. Setelah penyelenggara berhasil melakukan *login* langkah selanjutnya penyelenggara dapat menambahkan acaranya. Berikut adalah tampilan tambah acara yang dapat dilihat pada Gambar 5.



Gambar 5. Tampilan tambah acara (1) dan memilih lokasi acara (2)

Gambar 5 merupakan tahap untuk menambahkan acara yang dilakukan oleh penyelenggara dengan mengisi semua *field* yang ada yaitu nama acara, tanggal, hari, jam mulai, jam selesai, periode, jam keterlambatan, lokasi, jangkauan lokasi dan deskripsi lokasi. Terdapat tombol *maps* digunakan untuk menentukan lokasi acara dengan

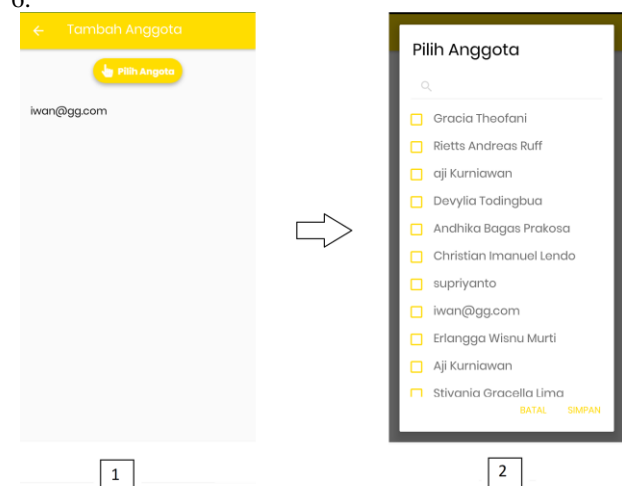
mengambil data *latitude* dan *longitude*. Berikut adalah kode program untuk tambah acara pada Kode Program 1.

KODE PROGRAM 1  
TAMBAH ACARA

```

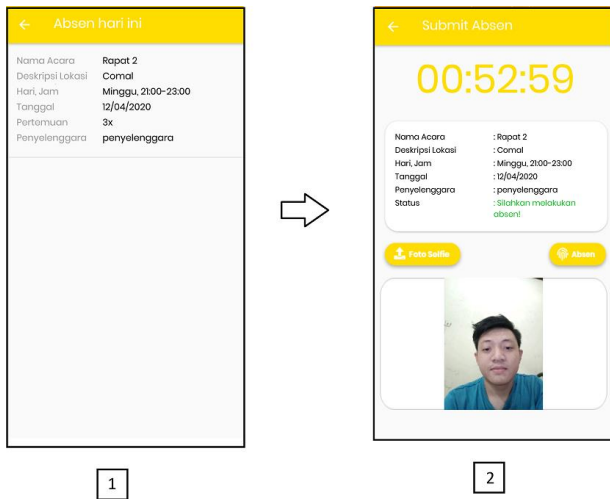
1. Map<String, Object> acara = new HashMap<>();
2. acara.put("nama_acara",
   String.valueOf(edtNama_acara.getText()));
3. acara.put("des_lokasi",
   String.valueOf(edtDes_lokasi.getText()));
4. acara.put("id_penyelenggara",
   mAuth.getCurrentUser().getUid());
5. acara.put("nama_user", nama_user);
6. acara.put("latitude", edtLatitude.getText().toString());
7. acara.put("longitude", edtLongitude.getText().toString());
8. acara.put("tanggal_dari", tgl_mulai);
9. acara.put("tanggal_sampai", tgl_sampai2);
10. acara.put("periode",
    Long.valueOf(String.valueOf(edtPeriode.getText())));
11. acara.put("tanggal_hari",
    String.valueOf(edtTanggal_hari.getText()));
12. acara.put("jam_terlambat",
    edtJam_terlambat.getText().toString());
13. acara.put("distance",
    Long.valueOf(edtDistance.getText().toString()));
14. db.collection("Acara")
15. .add(acara)
16. .addOnSuccessListener(new
    OnSuccessListener<DocumentReference>() {
17. @Override
18. public void onSuccess(DocumentReference
    documentReference) {
19. progressDialog.dismiss();
20. Toast.makeText(TambahAcara.this, "Acara berhasil di simpan!,
    silakan pilih anggota absensi pada menu (acara berlangsung ->
    tambah anggota)!",
21. Toast.LENGTH_LONG).show();
22. finish();
    
```

Kode Program 1 merupakan kode program yang ada di *activity class TambahAcaraActivity* yang berfungsi untuk menambahkan acara, pada baris 1 sampai baris 13 merupakan inialisasi variabel acara, kemudian pada baris 14 sampai baris 22 merupakan fungsi untuk menyimpan data kedalam *database firestore*. Setelah penyelenggara berhasil menambahkan acara langkah selanjutnya adalah penyelenggara menambahkan anggota pada acara yang telah ditambahkan. Berikut adalah tampilan tambah anggota yang dapat dilihat pada Gambar 6.



Gambar 6. Tampilan Tambah Anggota (1) Memilih Anggota (2)

Gambar 6 merupakan tahap untuk menambahkan anggota pada acara yang telah ditambahkan. Penyelenggara memilih anggota yang akan ikut dalam setiap acaranya. Setelah penyelenggara berhasil menambahkan anggota pada setiap acaranya, langkah selanjutnya anggota yang sudah terdaftar dalam acara dapat melakukan absensi. Berikut tampilan halaman absensi anggota yang dapat dilihat pada Gambar 7.



Gambar 7. Tampilan Acara Anggota (1) Submit Absen (2)

Gambar 7 merupakan proses anggota dalam melakukan absen. Anggota yang akan melakukan absensi harus melalui dua validasi: 1) memverifikasi lokasi *smartphone* dari anggota acara; 2) aplikasi memverifikasi lokasi anggota acara dari informasi lokasi swafoto yang harus dilakukan melalui aplikasi AbsenLoc dengan mengakses kamera *smartphone*. Adapun kedua verifikasi tersebut akan dicocokkan dengan jangkauan *latitude* dan *longitude* yang telah ditetapkan oleh penyelenggara acara. Jika lokasi anggota acara di luar jangkauan tersebut, maka absensi tidak akan terekam di AbsenLoc. Berikut adalah kode program untuk proses absensi pada Kode Program 2.

KODE PROGRAM 2  
PROSES ABSENSI

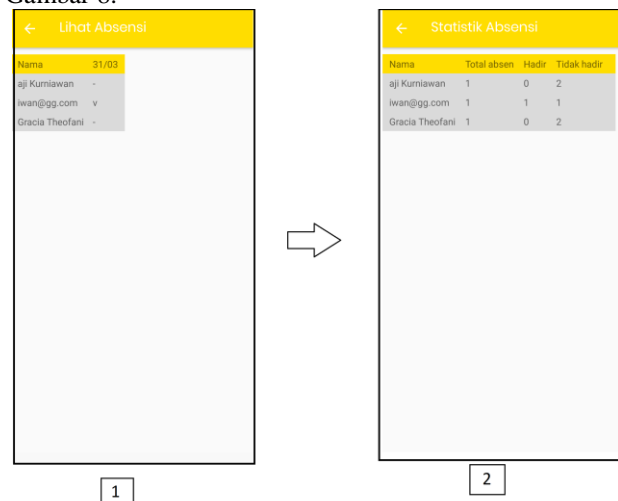
```

1. float lat_a = Float.valueOf(lat);
2. float lng_a = Float.valueOf(longe);
3. float lat_b = (float) mLastLocation.getLatitude();
4. float lng_b = (float) mLastLocation.getLongitude();
5. double earthRadius = 3958.75;
6. double latDiff = Math.toRadians(lat_b-lat_a);
7. double lngDiff = Math.toRadians(lng_b-lng_a);
8. double a = Math.sin(latDiff /2) * Math.sin(latDiff /2) +
9. Math.cos(Math.toRadians(lat_a)) *
  Math.cos(Math.toRadians(lat_b))
10. Math.sin(lngDiff /2) * Math.sin(lngDiff /2);
11. double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
12. double distance = earthRadius * c;
13. int meterConversion = 1609;
14. float meter = (float) (distance * meterConversion);
15. if ((int) meter > distance2){
16. ada = false;
17. showToast("Anda tidak pada jangkauan lokasi acara ini!");}
18. if (ada){
19. if (first) {
    
```

```

20. Map<String, Object> absen = new HashMap<>();
21. absen.put("id_acara", id_acara);
22. absen.put("nama_acara", nama_acara);
23. absen.put("id_user", mAuth.getCurrentUser().getUid());
24. absen.put("nama_user", nama_user);
25. absen.put("des_lokasi", des_lokasi);
26. absen.put("tanggal_acara", tanggal);
27. absen.put("tanggal_absen", currentTime);
28. db = FirebaseFirestore.getInstance();
29. db.collection("Absen")
30. .add(absen)
31. .addOnSuccessListener(new
  OnSuccessListener<DocumentReference>() {
32. @Override
33. public void onSuccess(DocumentReference
  documentReference) {
34. progressDialog.dismiss();
35. showToast("Anda berhasil absen!");
36. finish();}})
    
```

Kode Program 2 merupakan kode program yang ada di *activity class AbsenTodayActivity*. Pada baris program 1 sampai 15 merupakan inialisasi variabel serta perhitungan jarak lokasi acara dengan lokasi anggota, adapun perhitungan yang dilakukan dengan mencocokkan jangkauan *longitude* dan *latitude* pada kedua lokasi tersebut. Jika lokasi sudah benar maka akan menjalankan kode program 20 sampai 36 yang merupakan kode program untuk inialisasi variabel serta fungsi untuk menyimpan data pada *database firestore*, jika lokasi tidak cocok maka akan menjalankan kode program 16 sampai 17. Anggota dan penyelenggara acara juga dapat melihat rekap data absensi. Berikut adalah tampilan rekap data absensi pada aplikasi AbsenLoc yang dapat dilihat pada Gambar 8.



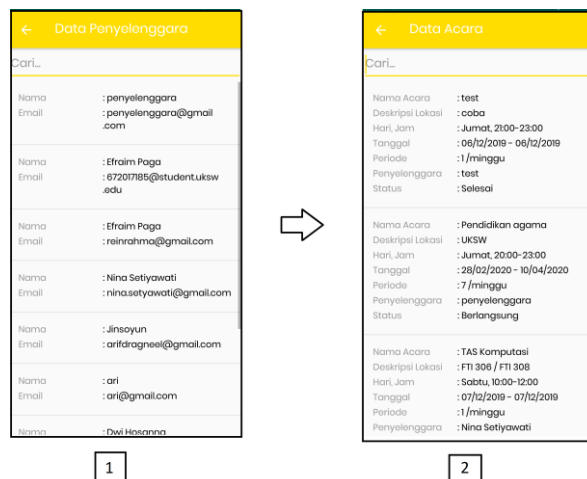
Gambar 8. Tampilan Absensi (1) Statistik Absensi (2)

Gambar 8 merupakan tampilan rekap data absensi, penyelenggara acara dan anggota dapat melihat rekap data absensi pada aplikasi AbsenLoc. Berikut adalah tampilan menu administrator yang dapat dilihat pada Gambar 9.

Gambar 9 merupakan tampilan data penyelenggara, anggota dan acara yang dapat diolah oleh administrator, administrator dapat melihat dan menghapus data-data tersebut. Pada penelitian ini aplikasi AbsenLoc diuji menggunakan *black box testing*. *Black box testing* merupakan metode pengujian perangkat lunak yang

digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program [14]. Berikut adalah tabel pengujian menggunakan *black box testing* untuk *role* penyelenggara acara yang akan dijelaskan pada Tabel I.

Berdasarkan Tabel I, terlihat bahwa hasil pengujian dilakukan pada modul penyelenggara dengan tugas menambahkan acara dan menambahkan anggota menunjukkan bahwa semua pola pengujian diterima. Hasil ini memperlihatkan bahwa proses perancangan yang dilakukan sesuai dengan yang diharapkan. Adapun hasil pengujian aplikasi AbsenLoc untuk modul anggota terlihat pada Tabel II.



Gambar 9. Tampilan Data Penyelenggara (1) dan Data Anggota (2)

TABEL I  
HASIL PENGUJIAN ABSENLOC UNTUK PENYELENGGARA

No	Pola Pengujian	Validasi Input	Data Input	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Login	Email	Email yang terdaftar pada database	Berhasil login	Sesuai harapan	Valid
			Email dikosongkan	Gagal login	Sesuai harapan	
			Email tidak valid	Gagal login	Sesuai harapan	
		Password	Password valid	Berhasil login	Sesuai harapan	
			Password dikosongkan	Gagal login	Sesuai harapan	
			Password tidak valid	Gagal login	Sesuai harapan	
2	Tambah Acara	Input Nama Acara, Tanggal, Hari, Jam mulai, Jam Selesai, Periode, Jam Keterlambatan, Lokasi, Jangkauan Lokasi dan Deskripsi Lokasi	Data valid	Data berhasil disimpan, muncul pesan untuk menambahkan anggota	Sesuai harapan	Valid
			Data dikosongkan	Muncul pesan kesalahan	Sesuai harapan	
3	Tambah Anggota	Input data anggota	Memilih data anggota	Muncul pesan data berhasil disimpan	Sesuai harapan	Valid
			Tidak memilih data anggota	Muncul pesan silahkan pilih data anggota	Sesuai harapan	
4	Logout		Memilih menu logout	Keluar dari menu utama dan kembali ke menu login	Sesuai harapan	Valid

TABEL II  
HASIL PENGUJIAN ABSENLOC UNTUK ANGGOTA

No	Pola Pengujian	Validasi Input	Data Input	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Login	Email	Email yang terdaftar pada database	Berhasil login	Sesuai harapan	Valid
			Email dikosongkan	Gagal login	Sesuai harapan	
			Email tidak valid	Gagal login	Sesuai harapan	
		Password	Password valid	Berhasil login	Sesuai harapan	
			Password dikosongkan	Gagal login	Sesuai harapan	
			Password tidak valid	Gagal login	Sesuai harapan	
2	Melakukan Absen	Validasi lokasi anggota dan lokasi swafoto	Validasi lokasi dan swafoto anggota valid	Berhasil absen	Sesuai harapan	Valid

		anggota	Validasi lokasi dan swafoto anggota tidak valid	Gagal absen	Sesuai harapan	
<b>No</b>	<b>Pola Pengujian</b>	<b>Validasi Input</b>	<b>Data Input</b>	<b>Hasil yang diharapkan</b>	<b>Hasil Pengujian</b>	<b>Kesimpulan</b>
3	<i>Logout</i>		Memilih menu <i>logout</i>	Keluar dari menu utama dan kembali ke menu <i>login</i>	Sesuai harapan	Valid

Berdasarkan Tabel II, terlihat bahwa hasil pengujian pada modul anggota untuk tugas melakukan absen adalah valid dan diterima. Hasil ini memperlihatkan bahwa proses perancangan yang dilakukan sesuai dengan yang diharapkan. Berikut adalah hasil pengujian modul administrator yang dapat dilihat pada Tabel III.

TABEL III  
HASIL PENGUJIAN ABSENLOC UNTUK ADMINISTRATOR

No	Pola Pengujian	Validasi Input	Data Input	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Login</i>	Email	Email yang terdaftar pada <i>database</i>	Berhasil <i>login</i>	Sesuai harapan	Valid
			Email dikosongkan	Gagal <i>login</i>	Sesuai harapan	
			Email tidak valid	Gagal <i>login</i>	Sesuai harapan	
		<i>Password</i>	<i>Password</i> valid	Berhasil <i>login</i>	Sesuai harapan	
			<i>Password</i> dikosongkan	Gagal <i>login</i>	Sesuai harapan	
			<i>Password</i> tidak valid	Gagal <i>login</i>	Sesuai harapan	
2	Mengolah data	Melakukan proses hapus data	Hapus data	Data berhasil dihapus	Sesuai harapan	Valid
			Batal hapus	Data tidak ditampilkan	Sesuai harapan	
3	<i>Logout</i>		Memilih menu <i>logout</i>	Keluar dari menu utama dan kembali ke menu <i>login</i>	Sesuai harapan	Valid

Berdasarkan Tabel III, terlihat bahwa hasil pengujian dilakukan terkait proses pengolahan data oleh administrator menunjukkan bahwa tidak terdapat masalah pada semua proses pengujian. Hasil ini memperlihatkan bahwa antara proses perancangan yang dilakukan sesuai dengan yang diharapkan. Adapun pengujian yang dilakukan dengan melibatkan penyelenggara serta anggota dengan mengisi kuisisioner pada Tabel IV.

Tabel IV menunjukkan hasil pernyataan *user* pada pengujian aplikasi AbsenLoc. Jawaban yang disediakan diantaranya: SS = Sangat Setuju, S = Setuju, R = Ragu-ragu, TS = Tidak Setuju, STS = Sangat Tidak Setuju. Nilai presentase yang dibutuhkan diantaranya: 0% - 19,99% = STS, 20% - 39,99% = TS, 40% - 59,99% = R, 60% - 79,99% = S, 80% - 100% = SS. Hasil kuisisioner dihitung menggunakan skala Likert yang sering digunakan sebagai skala penilaian karena memberi nilai terhadap sesuatu [15]. Berdasarkan hasil pengujian di atas, dapat disimpulkan bahwa responden sangat setuju aplikasi AbsenLoc memudahkan dalam proses absensi. Responden juga sangat setuju aplikasi dapat meningkatkan efisiensi proses absensi serta memberikan kemudahan dalam melihat data absensi. Oleh karena itu dapat disimpulkan bahwa aplikasi AbsenLoc sudah sesuai dengan tujuan pembangunannya yaitu memberikan proses validasi absensi yang efektif, mengurangi waktu antrean dan membantu perusahaan dalam mengelola pengarsipan absensi.

TABEL IV  
HASIL PENGUJIAN USER

	Pernyataan	SS	S	R	TS	STS
1	Warna aplikasi sesuai dan tidak mencolok	14	5	1	0	0
2	Aplikasi memudahkan dalam proses absensi	13	4	3	0	0
3	Proses pendaftaran akun aplikasi mudah	15	5	0	0	0
4	Proses absensi di aplikasi mudah dipahami	11	6	3	0	0
5	Proses absensi di aplikasi sudah efisien	18	2	0	0	0
6	Melihat data absensi mudah dipahami	18	1	1	0	0

TABEL V  
HASIL PERHITUNGAN MENGGUNAKAN SKALA LIKERT

	Pernyataan	Jumlah	%
1	Warna aplikasi sesuai dan tidak mencolok	20	93%
2	Aplikasi memudahkan dalam proses absensi	20	90%
3	Proses pendaftaran akun aplikasi mudah	20	95%
4	Proses absensi di aplikasi mudah dipahami	20	88%
5	Proses absensi di aplikasi sudah efisien	20	98%
6	Melihat data absensi mudah dipahami	20	97%

## IV. KESIMPULAN

Pada penelitian ini aplikasi AbsenLoc telah berhasil dibangun menggunakan Android Studio dan Firestore di mana untuk validasi kehadiran peserta acara dilakukan melalui pencocokan lokasi peserta dengan lokasi acara berdasarkan titik *longitude* dan *latitude* yang telah ditentukan penyelenggara. Selain itu validasi tahap kedua dilakukan dengan mencocokkan lokasi dari hasil swafoto yang dilakukan dari kamera pada *smartphone*. Berdasarkan pengujian terhadap pengguna, didapatkan bahwa aplikasi AbsenLoc dapat memberikan proses validasi absensi yang efektif, mengurangi waktu antrean dan membantu perusahaan dalam mengelola pengarsipan absensi.

Untuk penelitian ke depan, Aplikasi AbsenLoc dapat dikembangkan lebih lanjut dengan memberikan peningkatan validasi terhadap peserta atau anggota yang akan melakukan absensi, dengan memanfaatkan teknologi *face recognition* atau validasi wajah sehingga peserta yang akan melakukan absensi harus melalui tiga tahap validasi yaitu dengan validasi lokasi peserta, lokasi swafoto dan validasi wajah. Pengembangan aplikasi AbsenLoc berbasis iOS juga dapat dibangun.

## DAFTAR PUSTAKA

- [1] E. Devie and E. Winarno, "Aplikasi Location Based Service untuk Informasi Kuliner di Yogyakarta," *Dinamik*, vol. 23, no. 1, pp. 15–21, 2018.
- [2] E. Budiman, "Pemanfaatan Teknologi Location Based Service dalam Pengembangan Aplikasi Profil Kampus Universitas Mulawarman Berbasis Mobile," *J. Ilm. Ilk.*, vol. 8, no. 3, pp. 137–144, 2016.
- [3] Safaat Nazruddin, *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung, 2012.
- [4] StatCounter, "Mobile Operating System Market Share Worldwide," 2020. <https://gs.statcounter.com/os-market-share/mobile/worldwide> (accessed Jun. 04, 2020).
- [5] H. Huang and G. Gartner, "Current Trends and Challenges in Location-Based Services," *Int. J. Geo-Information*, vol. 7, no. 6, p. 199, 2018.
- [6] I. Al Fikri, D. Herumurti, and R. R. H., "Aplikasi Navigasi Berbasis Perangkat Bergerak dengan Menggunakan Platform Wikitude untuk Studi Kasus Lingkungan ITS," *J. Tek. ITS*, vol. 5, no. 1, 2016.
- [7] Google LLC, "Cloud Firestore," *Firebase*, 2019. <https://firebase.google.com/docs/firestore/?hl=es-419>.
- [8] M. Faya and N. Fiftin, "PEMANFAATAN GOOGLE MAPS API UNTUK PEMBANGUNAN SISTEM INFORMASI MANAJEMEN BANTUAN LOGISTIK PASCA BENCANA ALAM BERBASIS MOBILE WEB ( Studi Kasus : Badan Penanggulangan Bencana Daerah Kota Yogyakarta )," *J. Sarj. Tek. Inform.*, vol. 1, no. 1, pp. 162–171, 2013.
- [9] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, Dan R&D*. Bandung, 2013.
- [10] Suendri, "Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan)," *J. Ilmu Komput. dan Inform.*, vol. 3, no. 1, 2018.
- [11] T. A. Kurniawan, "Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 1, pp. 77–86, 2018.
- [12] A. A. Imam, S. Basri, R. Ahmad, J. Watada, T. M. Gonzalez-Aparicio, and A. M. Almomani, "Data Modeling Guidelines for NoSQL Document-Store Databases," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 10, pp. 544–554, 2018.
- [13] D. Kristanto and B. A. Arnado, "Implementasi Website Pencarian Kos dengan NOSQL," *J. Inform. dan Komput.*, vol. 2, no. 2, pp. 103–107, 2017.
- [14] C. Wahyu Nur, Yulianingsih, and S. M. Sri, "Pengujian Black

- Box Testing Pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *J. String*, vol. 3, no. 2, pp. 206–210, 2018.
- [15] H. Retnawati, "Perbandingan Akurasi Penggunaan Skala Likert dan Pilihan Ganda untuk Mengukur Self-Regulated Learning," *J. Kependidikan*, vol. 45, no. 2, pp. 156–167, 2015.