

Implementasi Geofencing Dengan Algoritma Ray Casting Pada Aplikasi E-Presensi Rumah Tahfidz Berbasis Android

¹ Moh. Ridho Abdul Hakim, ² Adi Fajaryanto Cobantoro, ³ Moh. Bhanu Setyawan

^{1,2,3} Program Studi Teknik Informatika, Universitas Muhammadiyah Ponorogo

¹ m.ridho826@gmail.com, ² adifajaryanto@umpo.ac.id, ³ m.banu@umpo.ac.id

Rumah Tahfidz ‘Aisyiyah Bagor in Nganjuk Regency does not yet have an effective attendance system to record teachers’ presence, which makes data management prone to irregularities. This condition requires a technology-based solution that can improve the accuracy and efficiency of attendance recording. This research aims to develop an Android-based e-attendance application that applies geofencing with the Ray Casting algorithm as a location validation method. The application is designed using the Flutter framework with Firebase as the backend to support real-time data storage. The research method includes literature study, system design, algorithm implementation, and white-box testing to verify program logic. Trials were conducted on user authentication, location validation using geofence, and check-in/check-out attendance features. The test results show that the Ray Casting algorithm can validate location points with 100% accuracy in all scenarios, both when users are inside and outside the designated area. In addition, all main application functions run properly without errors. In conclusion, the developed e-attendance application has proven to be stable, accurate, and feasible for implementation as a location-based attendance recording system, and it has the potential to be applied in similar educational institutions to support efficient attendance management.

Keywords — *Android, firebase, geofencing, presence, ray casting*

Rumah Tahfidz ‘Aisyiyah Bagor di Kabupaten Nganjuk belum memiliki sistem presensi yang efektif untuk mencatat kehadiran para pengajar, sehingga rawan menimbulkan ketidakteraturan dalam pengelolaan data. Kondisi tersebut memerlukan solusi berbasis teknologi yang dapat meningkatkan akurasi dan efisiensi pencatatan kehadiran. Penelitian ini bertujuan mengembangkan aplikasi e-presensi berbasis Android yang menerapkan geofencing dengan algoritma Ray Casting sebagai metode validasi lokasi. Aplikasi dirancang menggunakan framework Flutter dengan Firebase sebagai backend untuk mendukung penyimpanan data secara real-time. Metode penelitian yang digunakan meliputi studi literatur, perancangan sistem, implementasi algoritma, serta pengujian white-box untuk memverifikasi logika program. Uji coba dilakukan terhadap autentikasi pengguna, validasi lokasi berbasis geofence, dan fitur presensi masuk/keluar. Hasil pengujian menunjukkan bahwa algoritma Ray Casting mampu memvalidasi titik lokasi dengan akurasi 100% pada seluruh skenario, baik saat pengguna berada di dalam maupun di luar area yang ditentukan. Selain itu, seluruh fungsi utama aplikasi berjalan dengan baik tanpa error. Kesimpulannya, aplikasi e-presensi yang dikembangkan terbukti stabil, akurat, dan layak diimplementasikan sebagai sistem pencatatan kehadiran berbasis lokasi, serta berpotensi diterapkan di lembaga pendidikan serupa untuk mendukung efisiensi manajemen kehadiran.

Kata Kunci—*Android, firebase, geofencing, presensi, ray casting*

I. PENDAHULUAN

Kemajuan pendidikan tidak terlepas dari perkembangan teknologi digital yang mengubah sistem administrasi dan layanan informasi di lembaga pendidikan. Teknologi smartphone berbasis Android memberikan kemudahan dalam menjalankan berbagai aplikasi pendukung administrasi, termasuk sistem presensi digital yang mulai menggantikan metode konvensional [1]. Rumah Tahfidz Al-Qur’an (RTQ) ‘Aisyiyah Bagor, yang memiliki jumlah santri dan pengajar yang terus meningkat, masih menghadapi tantangan dalam sistem pencatatan kehadiran, sehingga berisiko menurunkan efektivitas pengawasan dan pengelolaan kehadiran [2].

Solusi yang relevan untuk mengatasi masalah ini adalah pengembangan aplikasi presensi berbasis Android yang mengintegrasikan teknologi geolocation dan geofencing. Dengan mekanisme ini, pencatatan kehadiran hanya dapat dilakukan saat perangkat pengguna berada dalam area yang telah ditentukan, sehingga mencegah praktik titip absen dan memudahkan pengawasan terhadap pengajar yang bertugas di luar lokasi RTQ [3][4]. Keunggulan utama pendekatan ini adalah fleksibilitasnya—dibandingkan metode fingerprint atau QR code, aplikasi Android tidak

memerlukan perangkat keras tambahan, serta data presensi dapat langsung tersimpan di cloud untuk kebutuhan integrasi dan pelaporan [1] [5].

Dari sisi teknis, aplikasi ini menerapkan algoritma *ray casting* untuk memeriksa apakah lokasi pengguna berada dalam batas *polygon geofence* yang telah ditentukan. Algoritma ini dipilih karena akurat dan ringan secara komputasi, sehingga cocok digunakan di perangkat Android [6]. Dengan demikian, sistem presensi digital ini diharapkan dapat meningkatkan akurasi pencatatan kehadiran serta efektivitas pengawasan terhadap pengajar di RTQ 'Aisyiyah Bagor maupun lembaga pendidikan sejenis.

II. METODE PENELITIAN

A. Studi Literatur

Studi literatur dilakukan dengan menelaah berbagai sumber referensi seperti buku, artikel ilmiah, jurnal, dan dokumentasi teknis yang relevan dengan topik penelitian[7], mencakup konsep sistem presensi berbasis lokasi, teknologi GPS, *geofencing*, serta penerapan algoritma *Ray Casting* dalam validasi posisi di area poligon. Kajian ini juga meliputi teknologi pendukung seperti framework Flutter untuk pengembangan lintas platform, layanan Firebase (Cloud Firestore dan Firebase Authentication) sebagai *backend* berbasis *cloud*, serta metode pengujian perangkat lunak yang sesuai. Hasil studi literatur ini menjadi landasan dalam merancang arsitektur sistem, memilih metode implementasi, dan menyusun skenario pengujian aplikasi e-presensi yang dikembangkan.

B. GPS

GPS adalah singkatan dari *Global Positioning System* merupakan sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit. Sistem yang pertama kali dikembangkan oleh Departemen Pertahanan Amerika Serikat ini digunakan untuk kepentingan militer maupun sipil. Sistem GPS mempunyai tiga segmen yaitu: satelit, pengontrol dan penerima/pengguna. Satelitnya mengorbit Bumi dengan orbit dan kedudukan yang tetap [8].

C. Firebase

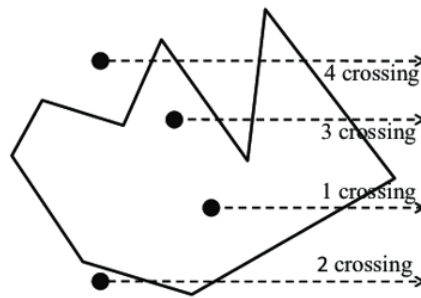
Dalam penelitian ini, Firebase, khususnya Cloud Firestore dan Firebase Authentication, digunakan sebagai layanan *Backend-as-a-Service* (BaaS) untuk mendukung pengembangan aplikasi berbasis Android secara *real-time* dan aman; Cloud Firestore dipilih karena merupakan basis data NoSQL yang fleksibel, mendukung sinkronisasi data secara *real-time*, kueri ekspresif, serta skala besar [9][10]. sementara Firebase Authentication memudahkan peneliti dalam mengimplementasikan sistem otentikasi pengguna tanpa membangun infrastruktur backend yang kompleks [11].

D. Flutter

Pengembangan aplikasi dilakukan menggunakan *framework* Flutter, sebuah *open-source SDK* dari Google yang memungkinkan pembuatan antarmuka aplikasi lintas platform (Android dan iOS) dengan satu basis kode (*single codebase*), didukung oleh mesin render sendiri untuk kinerja tinggi dan fitur *hot reload* yang mempercepat proses pengembangan melalui umpan balik visual instan tanpa kehilangan status aplikasi [12].

E. Geofencing dengan Algoritma Ray Casting

Algoritma *Ray Casting*, juga dikenal sebagai *even-odd rule* atau *crossing number*, merupakan metode klasik dalam geometri komputasi untuk menyelesaikan masalah *point-in-polygon* (PIP), yaitu menentukan apakah suatu titik berada di dalam atau di luar sebuah poligon. Dalam konteks *geofencing*, algoritma ini digunakan untuk mendeteksi apakah suatu objek, seperti perangkat GPS, berada dalam area geografis tertentu yang didefinisikan oleh poligon [13]. Prinsip kerja algoritma ini melibatkan penggambaran sinar horizontal dari titik yang diuji dan menghitung berapa kali sinar tersebut memotong sisi-sisi poligon seperti pada gambar 1. Jika jumlah potongan adalah ganjil, maka titik tersebut dianggap berada di dalam poligon; jika genap, maka di luar. Keunggulan utama dari algoritma *Ray Casting* adalah kesederhanaannya dan efisiensi waktu komputasi yang linear terhadap jumlah sisi poligon, menjadikannya cocok untuk aplikasi *real-time* seperti *geofencing* [14].



Gambar 1. Ilustrasi Geofencing dengan Ray Casting

Berikut langkah-langkah menentukan titik berada di dalam atau di luar poligon dengan *ray casting* (*Point-In-Polygon Test*):

- 1) Persiapkan Data:
 - a. Punya satu titik (x, y) yang ingin diuji.
 - b. Punya daftar titik-titik poligon (dalam urutan tertutup, misalnya $[(x_1, y_1), (x_2, y_2), \dots, (x_1, y_1)]$).
- 2) Inisialisasi:
 - a. Set jumlah crossing = 0.
- 3) Iterasi Setiap Sisi Poligon:
 - a. Untuk setiap segmen sisi poligon (p_1, p_2) :
 - Cek apakah sinar horizontal dari titik (x, y) memotong sisi tersebut.
 - Syarat memotong:
 - Y-coordinate dari titik berada di antara Y dari p_1 dan p_2 (bisa atas atau bawah).
 - Hitung titik potong X dari sinar horizontal dengan sisi tersebut.
 - Jika X dari titik potong $>$ X dari titik (x, y) , maka sinar memotong sisi tersebut.
- 4) Update Crossing:
 - a. Jika sinar memotong, tambahkan crossing ± 1 .
- 5) Tentukan Posisi Titik:
 - a. Jika total crossing ganjil, maka titik di dalam poligon.
 - b. Jika total crossing genap, maka titik di luar poligon.

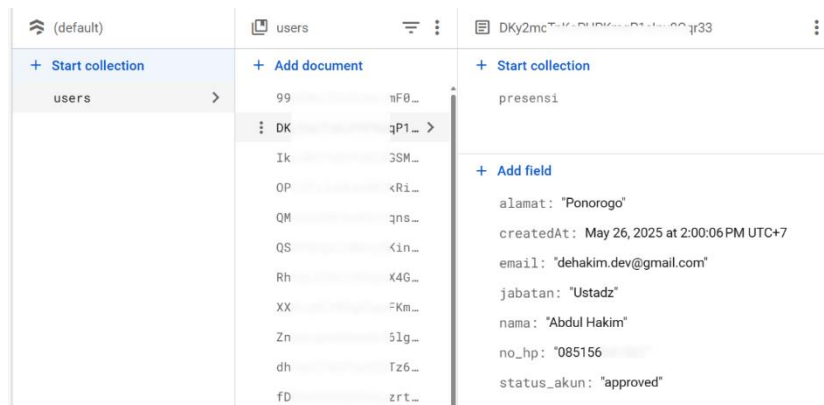
F. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi pada aplikasi berjalan sesuai dengan kebutuhan yang telah ditentukan. Proses pengujian bertujuan untuk menemukan kesalahan logika maupun kesalahan implementasi sehingga dapat dilakukan perbaikan sebelum aplikasi digunakan secara penuh. Pengujian dilakukan menggunakan pendekatan *white-box* untuk memeriksa logika internal program [15]. Pengujian khusus diterapkan pada algoritma *ray casting* untuk memvalidasi akurasi perhitungan posisi dalam sistem *geofencing*.

III. HASIL DAN PEMBAHASAN

A. Implementasi Database

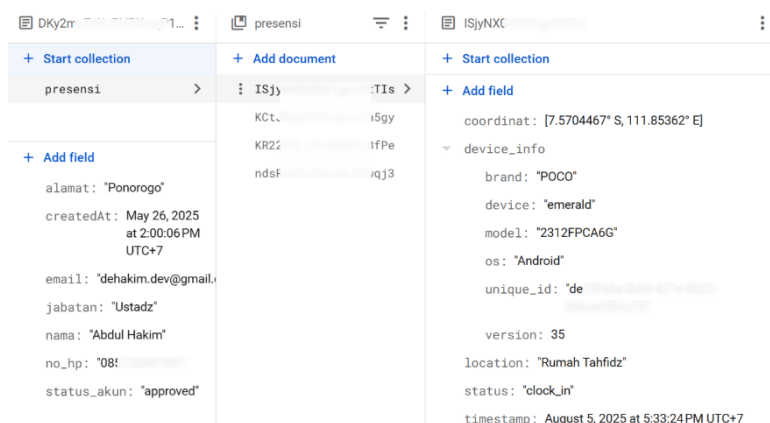
Aplikasi e-presensi ini menggunakan Firebase Firestore sebagai basis data untuk menyimpan informasi pengguna dan catatan kehadiran. Struktur data dirancang agar mendukung pengelompokan data berdasarkan pengguna dan aktivitasnya, serta mendukung pelacakan lokasi dan metadata perangkat.



Gambar 2. Struktur Koleksi Users

Pada gambar 2, setiap pengguna memiliki satu dokumen berdasarkan UID (user ID dari Firebase Auth). Di dalam dokumen ini disimpan informasi seperti:

1. nama: Nama lengkap pengajar.
2. email: Alamat email pengguna.
3. jabatan: Peran atau status di lembaga.
4. alamat: Alamat rumah atau asal.
5. no_hp: Nomor handphone.
6. status_akun: Status akun (misal: "approved" atau "pending").



Gambar 3. Struktur Subkoleksi Presensi

Pada gambar 3, setiap dokumen presensi menyimpan data seperti:

1. timestamp: Waktu presensi dilakukan.
2. status: clock_in atau clock_out.
3. location: Nama lokasi (misal: "Rumah Tahfidz").
4. coordinate: Koordinat GPS saat presensi (dalam format [lat, lng]).
5. device_info: Objek berisi informasi perangkat seperti brand, model, device, os, version, unique_id.

B. Implementasi Geofencing dengan Algoritma Ray Casting

Fitur utama dari sistem ini adalah validasi lokasi pengguna berdasarkan apakah mereka berada di dalam area yang telah ditentukan, yaitu area Rumah Tahfidz. Untuk itu, digunakan algoritma *Ray Casting* yang dapat menentukan apakah suatu titik (koordinat GPS pengguna) berada di dalam poligon (batas area *geofence*). Adapun potongan kode Dart untuk algoritma Ray Casting adalah sebagai berikut:

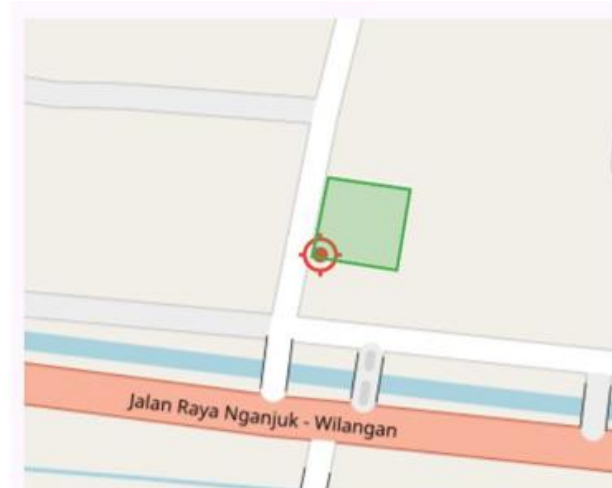
```
1 import 'package:latlong2/latlong.dart';
2
3 bool isPointInPolygon(LatLng point, List<LatLng> polygon) {
4   int intersectCount = 0;
5   for (int i = 0; i < polygon.length - 1; i++) {
6     LatLng a = polygon[i];
7     LatLng b = polygon[i + 1];
8
9     if ((a.latitude > point.latitude) != (b.latitude > point.latitude)) {
10      double atLng = (b.longitude - a.longitude) *
11        (point.latitude - a.latitude) /
12        (b.latitude - a.latitude) +
13        a.longitude;
14      if (point.longitude < atLng) {
15        intersectCount++;
16      }
17    }
18  }
19  return intersectCount % 2 == 1;
20 }
```

Gambar 4. Kode Dart Algoritma Ray Casting

Fungsi kode pada gambar 4 akan mengembalikan *true* jika titik berada di dalam *polygon*, dan *false* jika di luar. Fungsi ini dipanggil setiap kali pengguna menekan tombol presensi, untuk memvalidasi apakah mereka benar-benar berada di dalam area Rumah Tahfidz.

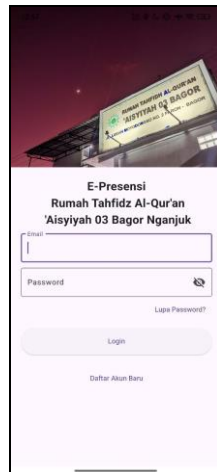
C. Rancangan Aplikasi Android

Aplikasi yang dibangun merupakan sistem E-Presensi Pengajar Rumah Tahfidz berbasis Android, yang bertujuan mempermudah proses pencatatan kehadiran para pengajar secara digital dan otomatis berdasarkan lokasi. Proses presensi hanya dapat dilakukan apabila pengguna berada di dalam area geofence yang telah ditentukan, yaitu area Rumah Tahfidz. Untuk melakukan validasi lokasi, aplikasi menggunakan algoritma *Ray Casting*, yaitu metode pengecekan apakah titik koordinat pengguna (dari GPS) berada di dalam atau di luar area poligon yang telah ditetapkan sebagai batas geofence. Data koordinat batas geofence Rumah Tahfidz disimpan secara lokal di dalam aplikasi dalam bentuk array titik-titik koordinat lintang dan bujur (latitude, longitude). Contoh visualisasi area tersebut dapat dilihat pada gam berikut:



Gambar 5. Area Batas Presensi

a) Halaman Login



Gambar 6. Halaman Login

Pada gambar 6, halaman login memungkinkan pengguna untuk masuk ke sistem menggunakan email dan password yang telah didaftarkan. Tampilan ini terdiri dari:

1. Nama Aplikasi, Field input email, Field input password (dengan fitur sembunyikan/tampilkan password).
2. Tombol “Masuk”.
3. Tombol “daftar akun baru”.
4. Notifikasi error jika email/password salah.

Setelah login berhasil, pengguna akan diarahkan ke halaman utama aplikasi.

b) Halaman Beranda

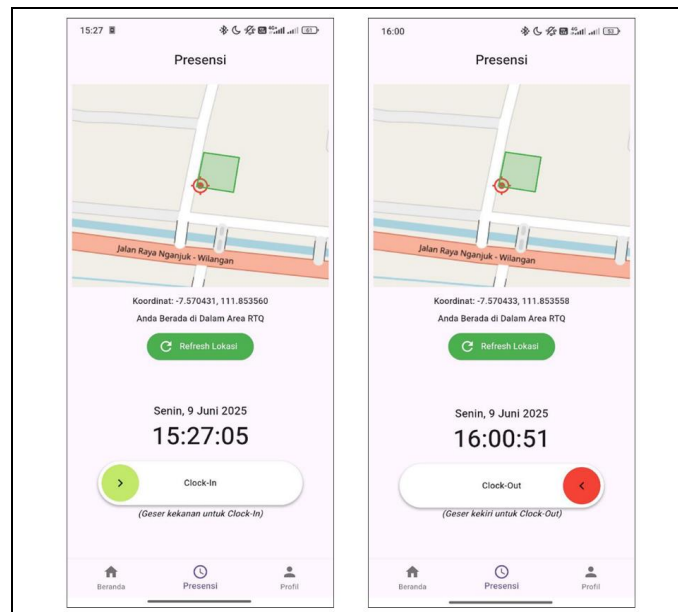


Gambar 7. Halaman Beranda

Pada gambar 7, halaman beranda merupakan halaman awal setelah pengguna login. Tampilan ini menyajikan:

1. Sambutan dengan nama pengguna.
2. Informasi lokasi saat ini.
3. Informasi hari, tanggal, dan waktu saat ini.
4. Informasi status presensi hari ini.
5. Statistik kehadiran pada bulan ini.
6. Tombol navigasi ke fitur pendukung, seperti “Izin”, “Agenda Di Luar”, “Rapat”, “Daftar Hadir Hari Ini”, “Riwayat Presensi”, “Rekap Bulanan”.
7. Tombol navigasi ke fitur utama, seperti "Beranda", "Presensi", dan "Profil".

c) Halaman Presensi

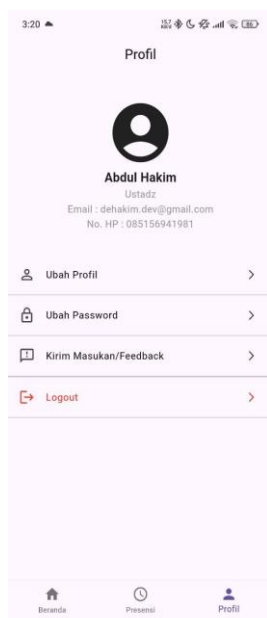


Gambar 7. Halaman Presensi

Pada gambar 8, halaman presensi merupakan inti dari aplikasi, tempat di mana pengguna melakukan check-in berdasarkan lokasi GPS. Komponen dalam halaman ini meliputi:

1. Peta interaktif menggunakan *package* “flutter_map: ^7.0.2” yang menunjukkan posisi pengguna dan batas area geofence
2. Tampilan lokasi saat ini (koordinat latitude dan longitude).
3. Tampilan lokasi pengguna apakah berada didalam atau di luar area rumah tahfidz.
4. Tombol “Refresh Lokasi” untuk mengambil ulang lokasi pengguna jika lokasi kurang akurat.
5. Tampilan hari, tanggal, waktu saat ini.
6. Tombol “Clock-in” untuk kirim data presensi ketika datang dan tombol “Clock-Out” untuk kirim data presensi ketika pulang.
7. Pesan notifikasi apakah data presensi bisa terkirim atau gagal terkirim.

d) Halaman Profil



Gambar 8. Halaman Profil

Halaman profil menampilkan informasi pengguna yang login, seperti:

1. Tampilan data pengguna yaitu: nama lengkap, Jabatan, Email, Nomor HP.
2. Tombol ubah profil, ubah *password*, kirim masukan/*feedback*, dan tombol untuk keluar (*logout*) dari akun.

D. Pengujian White Box

Autentikasi pengguna merupakan elemen penting dalam sebuah sistem yang membutuhkan kontrol akses. Dalam aplikasi e-Presensi RTQ, autentikasi dilakukan menggunakan layanan Firebase Authentication yang mengizinkan login berdasarkan email dan password. Seluruh logika autentikasi pada tabel 1 dikelola dalam class AuthService agar lebih terstruktur dan mudah diuji secara terpisah (modular).

Tabel 1. Identifikasi Method dan Tujuan Pengujian Integrasi pada Autentikasi Pengguna

Nama Class	Nama Method	Tujuan
AuthService	signIn()	Melakukan autentikasi pengguna menggunakan email dan password
AuthService	signOut()	Menghapus sesi login pengguna
AuthService	getCurrentUser()	Mengembalikan user yang sedang aktif saat ini

Saat pengguna memasukkan email dan password, method signIn() akan memanggil Firebase untuk melakukan autentikasi. Jika berhasil, informasi login disimpan menggunakan SharedPreferences dengan key isLoggedIn, yang dapat digunakan untuk menentukan apakah pengguna harus diarahkan ke halaman utama atau login ulang. Selain itu, method getCurrentUser() dapat dipanggil kapan saja untuk mengecek sesi login yang masih aktif. Fungsi signOut() digunakan untuk menghapus sesi tersebut saat pengguna ingin keluar dari aplikasi.

Berikut ini adalah hasil pengujian white box terhadap class AuthService:

Tabel 2. Hasil Pengujian Integrasi pada Autentikasi Pengguna

Input	Nama Method	Expected Result	Status
Email dan password valid	signIn()	Login berhasil, status tersimpan di SharedPreferences	Valid
Email atau password salah	signIn()	Muncul exception yang bisa ditangani	Valid

Logout ditekan	signOut()	User berhasil keluar dari akun	Valid
Setelah login	getCurrentUser()	Mengembalikan objek user yang aktif	Valid
Setelah logout	getCurrentUser()	Mengembalikan null	Valid

Pengujian dilakukan dengan memasukkan data kredensial yang benar dan salah, lalu memantau bagaimana aplikasi memberikan respons. Ketika data valid dimasukkan, pengguna dapat langsung diarahkan ke halaman utama aplikasi. Sebaliknya, ketika data tidak valid (misalnya email tidak terdaftar atau password salah), aplikasi memberikan umpan balik berupa pesan error melalui penanganan exception. Selain itu, proses logout berhasil menghapus sesi, ditandai dengan `getCurrentUser()` yang mengembalikan null.

Fitur presensi merupakan komponen utama dalam aplikasi ini. Fungsi utamanya terlihat pada tabel 3 memungkinkan pengguna untuk melakukan *clock-in* (presensi masuk) dan *clock-out* (presensi keluar) berdasarkan data lokasi GPS yang divalidasi dengan area yang telah ditentukan (geofence), serta batasan hari dan jam tertentu. Semua proses dilakukan secara otomatis dan terintegrasi dengan Firebase Firestore untuk pencatatan data presensi.

Tabel 3. Identifikasi Method dan Tujuan Pengujian Integrasi Fitur Presensi

Nama Class	Nama Method	Tujuan
LocationService	<code>getCurrentLocation()</code>	Mengambil lokasi terkini pengguna
PresensiScreen	<code>_cekLokasi()</code>	Memeriksa apakah lokasi pengguna berada di dalam geofence
PresensiScreen	<code>hasClockedInToday()</code>	Mengecek status clock-in untuk hari ini
PresensiScreen	<code>isWithinAllowedDayAndTime()</code>	Menentukan apakah waktu <i>clock-in/clock-out</i> sesuai batas
PresensiScreen	<code>ActionSlider.standard()</code>	Menyimpan data presensi ke Firestore berdasarkan validasi lokasi dan waktu

Tabel 4 berikut menunjukkan hasil pengujian integrasi pada fitur presensi.

Tabel 4. Hasil Pengujian Integrasi Fitur Presensi

Input	Nama Method	Expected Result	Status
Lokasi GPS valid	<code>getCurrentLocation()</code>	Menampilkan koordinat pengguna	Valid
Lokasi dalam polygon	<code>isPointInPolygon()</code>	Menampilka-n status “Dalam Area RTQ”	Valid
Lokasi di luar polygon	<code>isPointInPolygon()</code>	Menampilka-n status “Di Luar Area RTQ”	Valid
Belum presensi hari ini	<code>hasClockedInToday()</code>	Menampilkan tombol “Clock-In”	Valid
Sudah presensi hari ini	<code>hasClockedInToday()</code>	Menampilkan tombol “Clock-Out”	Valid
Clock-In pada waktu dan lokasi valid	<code>ActionSlider.standard()</code>	Clock-In berhasil, data tersimpan ke Firestore	Valid
Clock-In di luar lokasi	<code>ActionSlider.standard()</code>	Clock-In gagal, muncul pesan “Anda berada di luar lokasi”	Valid
Clock-In di luar jam presensi	<code>isWithinAllowedDayAndTime()</code>	Clock-In gagal, muncul pesan “Presensi hanya diizinkan pada jam tertentu”	Valid
Clock-Out setelah Clock-In	<code>ActionSlider.standard()</code>	Clock-Out berhasil, data tersimpan.	Valid
Menyegarka-n tampilan lokasi secara real-time	<code>startListeningLocation()</code>	Lokasi dan akurasi diperbarui otomatis di UI	Valid

Tabel 4 menunjukkan hasil pengujian terhadap fitur presensi pada aplikasi, yang mencakup berbagai skenario seperti pengambilan lokasi GPS, validasi titik lokasi terhadap area *geofence*, pengecekan status presensi harian, serta proses *clock-in* dan *clock-out*. Setiap metode diuji berdasarkan input yang relevan untuk memastikan fungsionalitas

berjalan sesuai harapan, seperti menampilkan koordinat, menentukan status lokasi, serta memverifikasi batasan waktu presensi. Hasil pengujian menunjukkan bahwa seluruh metode yang diuji memberikan *output* yang valid dan sesuai dengan rancangan sistem, termasuk penyimpanan data presensi ke Firebase, sehingga dapat disimpulkan bahwa fitur presensi telah berjalan dengan baik dan memenuhi kebutuhan fungsional aplikasi.

E. Pengujian Algoritma

Salah satu aspek penting dalam aplikasi presensi berbasis lokasi adalah validasi apakah pengguna benar-benar berada di dalam area yang ditentukan atau tidak. Dalam konteks aplikasi ini, area presensi didefinisikan sebagai sebuah poligon berdasarkan kumpulan koordinat (latitude dan longitude) yang membentuk batas fisik rumah tahfidz. Untuk memvalidasi apakah suatu titik koordinat berada di dalam atau di luar polygon, digunakan pendekatan algoritmik berbasis *Ray Casting Algorithm*. Implementasi algoritma ini dituangkan ke dalam *method* `isPointInPolygon()`, yang menjadi inti dari proses validasi lokasi presensi. Berikut pada tabel 5 hasil pengujian fungsi algoritma *Ray Casting*.

Tabel 5. Hasil Pengujian Fungsi Algoritma Ray Casting

No	Titik Uji (Lat, Lng)	Status Diharapkan	Hasil Fungsi	Status
1	(-7.570339238374288, 111.85358912272108)	Di dalam	Di dalam	Valid
2	(-7.570346460676234, 111.85367080043648)	Di dalam	Di dalam	Valid
3	(-7.570390174603773, 111.85358720540387)	Di dalam	Di dalam	Valid
4	(-7.570354823340466, 111.85361366438212)	Di dalam	Di dalam	Valid
5	(-7.5703821920604355, 111.85361098013823)	Di dalam	Di dalam	Valid
6	(-7.5704209644108715, 111.85361864940671)	Di dalam	Di dalam	Valid
7	(-7.570322132922996, 111.85363552179933)	Di dalam	Di dalam	Valid
8	(-7.570449853609787, 111.85368000355868)	Di dalam	Di dalam	Valid
9	(-7.570428946952646, 111.85355461101074)	Di dalam	Di dalam	Valid
10	(-7.570385993263557, 111.85369265788154)	Di dalam	Di dalam	Valid
11	(-7.570293524626379, 111.85361703461575)	Di luar	Di luar	Valid
12	(-7.570300920238807, 111.85368977584113)	Di luar	Di luar	Valid
13	(-7.570349299859814, 111.85372148355424)	Di luar	Di luar	Valid
14	(-7.570307083247741, 111.85345663088907)	Di luar	Di luar	Valid
15	(-7.570346834656533, 111.85355952552788)	Di luar	Di luar	Valid
16	(-7.5704408205283045, 111.85355237574976)	Di luar	Di luar	Valid
17	(-7.570461158452517, 111.85364003825117)	Di luar	Di luar	Valid
18	(-7.570412162543221, 111.8535349675929)	Di luar	Di luar	Valid
19	(-7.570442361280072, 111.85352160061649)	Di luar	Di luar	Valid
20	(-7.570457768798818, 111.85359589614137)	Di luar	Di luar	Valid

Hasil pengujian algoritma *Ray Casting* pada sistem e-presensi ditunjukkan melalui 20 titik uji koordinat GPS yang mewakili posisi di dalam maupun di luar area *geofence*. Dari pengujian tersebut, 10 titik yang seharusnya berada di dalam area terdeteksi dengan benar sebagai “Di dalam”, sementara 10 titik lainnya yang berada di luar area juga terdeteksi sesuai sebagai “Di luar”. Seluruh hasil uji menunjukkan status valid, sehingga dapat disimpulkan bahwa algoritma *Ray Casting* mampu memvalidasi posisi pengguna dengan tingkat akurasi 100% dan dapat diandalkan sebagai metode penentu lokasi dalam penerapan *geofencing* pada aplikasi e-presensi.

IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian, aplikasi e-presensi pengajar berbasis Android dengan algoritma *Ray Casting* terbukti berjalan sesuai dengan kebutuhan. Pengujian white-box pada fitur login, validasi lokasi, serta clock-in/clock-out menunjukkan hasil valid di seluruh skenario uji. Algoritma *Ray Casting* memberikan akurasi validasi lokasi sebesar 100%, sehingga dapat mencegah praktik presensi di luar area yang ditentukan. Dengan stabilitas dan integrasi data real-time melalui Firebase, aplikasi ini layak diterapkan untuk mendukung pengelolaan kehadiran pengajar secara lebih efisien di Rumah Tahfidz maupun lembaga pendidikan lain.

V. DAFTAR PUSTAKA

- [1] D. Abetnego and J. Susilo, "Aplikasi Presensi Kerja Berbasis Android dengan Monitoring Lokasi Karyawan Secara Real-Time Menggunakan Maps," *J. Inform. dan Bisnis*, vol. 13, no. 2, pp. 111–132, 2024, doi: 10.46806/jib.v13i2.1290.
- [2] L. Sari and G. yanti kemala Sari siregar, "Perancangan Aplikasi Pendataan Data Kepegawaian Negeri Sipil Pada Dinas Komunikasi Dan Informatika Kota Metro," *J. Mhs. Ilmu Komput.*, vol. 1, no. 2, pp. 115–135, 2021, doi: 10.24127/v2i1.1235.
- [3] Z. P. Vianto and E. R. Yulia, "Sistem Informasi Kehadiran Karyawan Berbasis Android Menggunakan Metode Geofencing Pada Pt. Gemilang Anugrah Permata," *Comput. Sci.*, vol. 2, no. 1, pp. 33–41, 2022, doi: 10.31294/coscience.v2i1.721.
- [4] L. Arizal, N. Pravitasari, and R. W. Putri, "Perancangan Sistem Informasi Absensi Berbasis Android Menggunakan Geofence Pada The Gade Coffee and Gold Kramat Raya," *J. Ilm. Multidisiplin*, vol. 2, no. 2, pp. 209–214, 2023, doi: 10.59000/jim.v2i2.153.
- [5] H. Shinde and G. Raul, "GPS based Attendance Management System with RFID Technology," vol. 5, no. 01, pp. 1–3, 2017, [Online]. Available: www.ijert.org
- [6] E. Schwinger, R. Twum, T. Katsepor, and G. Schwinger, "Point in polygon calculation using vector geometric methods with application to geospatial data," 2023, [Online]. Available: <http://arxiv.org/abs/2306.04316>
- [7] N. Fadhila Akbar, F. Putra Ramadi, R. Kusuma Dewa, and A. Wulansari, "Kajian Literatur (Systematic Literature Review) : It Performance Measurement Pada Perusahaan Jasa," *Sci. J. Ilm. Sain dan Teknol.*, vol. 2, no. 1, pp. 275–288, 2024.
- [8] Rianandra, Arsali, and A. A. Bama, "Studi Perbandingan Penentuan Posisi Geografis berdasarkan Pengukuran dengan GPS (Global Positioning System), Peta Google Earth, dan Navigasi.Net," *J. Penelit. Sains Mipa UNSRI*, vol. 17, no. 2, pp. 82–90, 2015.
- [9] J. Panjaitan and A. F. Pakpahan, "Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 20–34, 2021, doi: 10.28932/jutisi.v7i1.3098.
- [10] Sugiyatno, "Pengiriman Informasi Real Time Menggunakan Teknologi Database Firebase pada Aplikasi Mobile Android," *J. Inform. Komputer, Bisnis dan Manaj.*, vol. 21, no. 2, pp. 46–55, 2023, doi: 10.61805/fahma.v21i2.17.
- [11] W. Kurniawan, I. Prihandi, and N. Husufa, "Prototype Firebase Authentication Menggunakan Fitur Firebase Pada Aplikasi Android," *J. Satya Inform.*, vol. 4, no. 1, pp. 71–78, 2023, doi: 10.59134/jsk.v4i1.406.
- [12] Jauzaa Maylia Suhendro, Made Sudarma, and Duman Care Khrisne, "Rancang Bangun Aplikasi Seluler Penyedia Jasa Perawatan Dan Kecantikan Menggunakan Framework Flutter," *J. SPEKTRUM*, vol. 8, no. no. 2, pp. 68–82, 2021.
- [13] D. Teng, F. Baig, Q. Sun, J. Kong, and F. Wang, "IDEAL: A Vector-Raster Hybrid Model for Efficient Spatial Queries over Complex Polygons," *Proc. - IEEE Int. Conf. Mob. Data Manag.*, vol. 2021-June, pp. 99–108, 2021, doi: 10.1109/MDM52706.2021.00024.
- [14] P. M. N. S. A. Basid and F. Nugroho, "Smart-geofencing for system of reporting inadequate regional infrastructure using crossing and winding number," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 26, no. 3, pp. 1662–1671, 2022, doi: 10.11591/ijeecs.v26.i3.pp1662-1671.
- [15] A. Dimas Saputro and A. Azzahra Narwastika, "Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Peminjaman Sistem Aplikasi Perpustakaan," *Pros. Semin. Nas. Teknol. Inf. dan Bisnis*, pp. 407–412, 2023.