

Pengembangan Aplikasi Mobile untuk Early Warning System (EWS) Kebakaran Hutan

1st Shasyabilla Ardita Hadi

School of Electrical Engineering

Telkom University

Bandung, Indonesia

shasyaad@student.telkomuniversity.ac.id

2nd Uke Kurniawan Usman

School of Electrical Engineering

Telkom University

Bandung, Indonesia

ukeusman@telkomuniversity.ac.id

3rd Rizky Satria

School of Electrical Engineering

Telkom University

Bandung, Indonesia

rizkysat@telkomuniversity.ac.id

Abstrak — Kebakaran hutan di Indonesia merupakan bencana tahunan yang berdampak besar terhadap lingkungan, kesehatan, dan perekonomian. Penelitian ini mengembangkan FireGuard, aplikasi Early Warning System (EWS) untuk kebakaran hutan. FireGuard menyediakan fitur pemantauan data sensor suhu, kelembapan, dan gas asap secara real-time, penentuan status potensi kebakaran, integrasi pantauan CCTV langsung (*live streaming*), serta notifikasi peringatan dini kepada pengguna. Pengujian dilakukan menggunakan *black box testing* untuk memverifikasi fungsionalitas, dan *System Usability Scale* (SUS) untuk mengukur kegunaan.

Hasil menunjukkan seluruh fitur berjalan sesuai spesifikasi dan skor SUS sebesar 79,16 menempatkan FireGuard pada kategori baik. Aplikasi ini dinilai mudah digunakan, informatif, dan efektif dalam mendukung deteksi dini kebakaran hutan. Dengan kemampuannya menyajikan informasi cepat dan akurat, FireGuard diharapkan dapat mempercepat koordinasi antara masyarakat dan pihak berwenang serta meminimalkan dampak kebakaran.

Kata kunci— Aplikasi Mobile, Flutter, Firebase, Early Warning System

I. PENDAHULUAN

Kebakaran hutan merupakan permasalahan tahunan di Indonesia dengan dampak signifikan terhadap ekosistem, kesehatan masyarakat, dan perekonomian. Pada tahun 2023, tercatat lebih dari 1,6 juta hektar lahan dan hutan mengalami kerusakan akibat kebakaran [1]. Upaya deteksi dini yang telah ada sering kali menghadapi berbagai kendala, di antaranya keterlambatan penyampaian informasi, jangkauan wilayah yang terbatas, dan tingginya biaya operasional sistem konvensional. Hal ini mengakibatkan keterlambatan respon, sehingga api terlanjur meluas sebelum dilakukan penanganan. Dengan luasnya wilayah hutan di Indonesia serta keterbatasan sumber daya manusia dan infrastruktur, dibutuhkan solusi yang lebih cepat, terjangkau, dan mudah diakses oleh berbagai pihak.

Kemajuan teknologi informasi, khususnya pada platform mobile, membuka peluang pengembangan Early Warning System (EWS) yang lebih efektif. Penelitian ini mengusulkan pengembangan aplikasi untuk EWS yang bernama “Fireguard”. Aplikasi ini memiliki beberapa fitur utama, antara lain menampilkan data dari sensor lapangan,

memantau status hutan, menampilkan pantauan CCTV di lokasi rawan kebakaran, serta mengirimkan notifikasi peringatan kebakaran secara langsung kepada pengguna.. Fireguard diharapkan mampu mempercepat koordinasi antara masyarakat dan pihak berwenang, serta meminimalkan risiko kebakaran berkembang menjadi bencana besar.

II. KAJIAN TEORI

A. Aplikasi Mobile

Aplikasi mobile adalah perangkat lunak yang dirancang untuk berjalan di perangkat mobile seperti smartphone, tablet, dan lainnya. Aplikasi memungkinkan pengguna mengakses berbagai layanan digital secara fleksibel dan praktis melalui berbagai perangkat. Aplikasi mobile memiliki peran penting dalam sistem pemantauan kebakaran hutan karena dapat mengirim notifikasi secara *real-time* dan menampilkan kondisi hutan.

B. Flutter

Flutter adalah sebuah *software development kit* (SDK) *open-source* yang dikembangkan oleh Google dan digunakan untuk membangun aplikasi mobile secara *cross-platform*. Dengan menggunakan Flutter, pengembang dapat membuat aplikasi untuk Android, iOS, web, desktop hanya dengan satu *codebase*.

Flutter menggunakan bahasa pemrograman Dart sebagai *framework*. Dart merupakan bahasa pemrograman serbaguna (*general-purpose*) yang dikembangkan oleh Google. Bahasa ini mendukung berbagai kebutuhan pengembangan, mulai dari aplikasi, antarmuka web (*front-end*), IoT, aplikasi baris perintah (*CLI*), dan lainnya [2].

Salah satu fitur unggulan dari Flutter adalah sistem *hot reload* yang memungkinkan pengembang untuk melihat perubahan kode secara langsung tanpa harus melakukan build ulang aplikasi. Fitur ini sangat membantu dalam proses pengembangan user interface (UI) dan debugging karena perubahan kecil dapat segera diuji coba tanpa mengganggu alur kerja.

C. Firebase

Firebase adalah platform *Backend as a Service* (BaaS) yang dikembangkan oleh Google dan dirancang untuk membantu pengembang membangun aplikasi web dan mobile. Firebase menyediakan berbagai layanan backend, termasuk

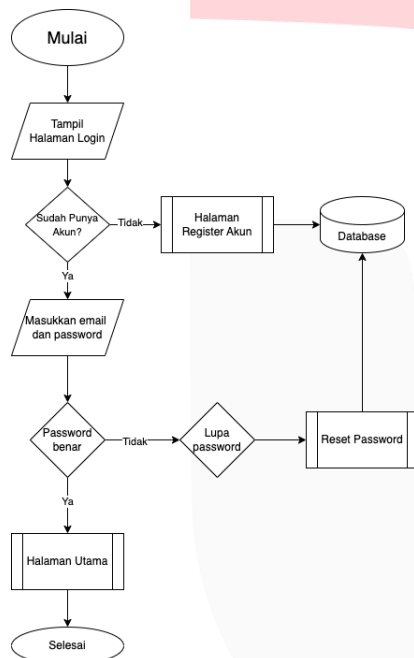
otentikasi pengguna (*Firestore Authentication*), penyimpanan data real-time (*Realtime Database* dan *Cloud Firestore*), pengelolaan file (*Cloud Storage*), analitik, serta notifikasi push (*Firestore Cloud Messaging/FCM*) [3]. *Firestore Auth* mendukung login dengan email dan kata sandi, *Google Sign-In*, nomor telepon, dan penyedia identitas pihak ketiga lainnya.

Realtime Database memungkinkan sinkronisasi data secara otomatis ke semua perangkat klien yang terhubung, sehingga setiap perubahan data akan langsung diperbarui tanpa memuat ulang aplikasi. Selain itu, *Firestore* dilengkapi fitur tambahan seperti *Crashlytics*, *Analytics*, dan *Remote Config* untuk mendukung pengembangan aplikasi yang stabil dan mudah disesuaikan bahkan setelah dirilis.

III. METODE

Penelitian ini dilakukan untuk membuat aplikasi yang mudah digunakan oleh masyarakat umum.

A. Diagram Alur Aplikasi Mobile

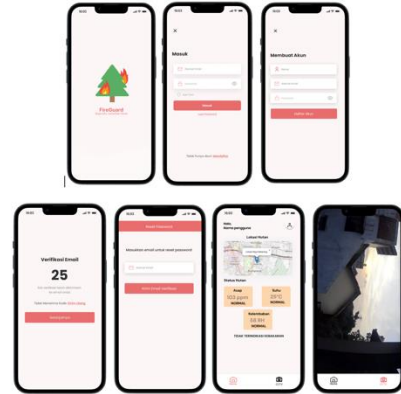


Gambar 1 Diagram Alur Aplikasi Mobile

Diagram alur pada Gambar 1 akan menjelaskan proses yang terjadi pada aplikasi mobile. Proses dimulai dari menampilkan halaman *login*. Pada tahap ini, terdapat opsi sudah memiliki akun atau belum. Jika belum, pengguna akan diarahkan ke halaman registrasi untuk membuat akun baru dan jika pengguna sudah memiliki akun, akan diminta untuk memasukkan email dan password yang telah dibuat.

Setelah data login dimasukkan, sistem akan melakukan pengecekan terhadap kecocokan email dan password. Jika informasi yang dimasukkan benar, pengguna akan diarahkan ke halaman utama aplikasi. Namun, jika terjadi kesalahan seperti password yang tidak sesuai, pengguna diberikan opsi untuk memilih fitur "Lupa Password". Fitur ini akan mengarahkan pengguna ke halaman reset password, di mana mereka dapat mengganti kata sandi baru. Setelah proses ini selesai, pengguna kembali ke halaman login.

B. Desain Antarmuka Aplikasi Mobile



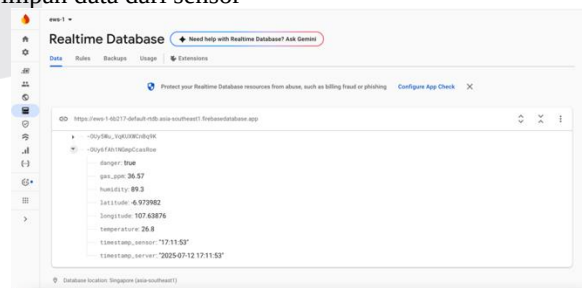
Gambar 2 Desain Antarmuka Aplikasi Mobile

Gambar 2 menunjukkan tampilan antarmuka aplikasi *FireGuard* yang dirancang untuk memantau kondisi hutan secara real-time. Tampilan awal (*splash screen*) menampilkan logo aplikasi dengan slogan "Siaga Dini, Lindungi Alam". Selanjutnya, pengguna diarahkan ke halaman masuk (*login*) yang terdiri dari kolom *email* dan *password* serta tombol untuk masuk, mendaftar akun baru, atau melakukan reset kata sandi. Pada halaman pembuatan akun (*sign up*), pengguna diminta mengisi nama, alamat *email*, dan kata sandi untuk mendaftar.

Jika pengguna lupa kata sandi, aplikasi menyediakan halaman verifikasi email yang menampilkan penghitung waktu (*countdown timer*) dan tombol untuk mengirim ulang kode verifikasi. Setelah itu, tersedia halaman reset kata sandi dengan kolom input email untuk menerima tautan verifikasi. Halaman utama aplikasi menampilkan peta lokasi sensor, serta informasi status hutan yang mencakup parameter asap (*ppm*), suhu (*°C*), dan kelembapan (*%*) yang dilengkapi indikator status. Apabila terdeteksi kondisi kebakaran, sistem akan memberikan peringatan pada halaman ini. Selain itu, terdapat fitur pemantauan langsung (*live streaming*) dari CCTV yang terhubung dengan lokasi sensor untuk memberikan visualisasi kondisi lapangan secara real-time.

C. Implementasi Firebase

Dalam implementasi menggunakan fitur platform *Firestore* yaitu *Firestore Realtime Database* sebagai penyimpanan data dari sensor



Gambar 3 Firebase Realtime Database

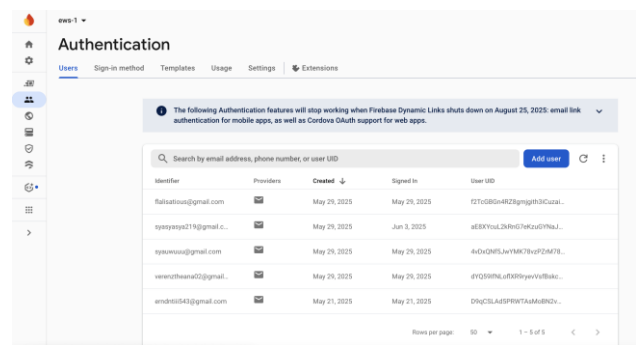
Firestore Realtime Database merupakan layanan basis data berbasis cloud yang menyimpan data dalam format *JSON* dan memperbarui data secara real-time ke semua klien yang terhubung. Dalam konteks aplikasi *FireGuard*, *Realtime Database* digunakan untuk menyimpan data hasil pembacaan

sensor lingkungan seperti suhu, kelembapan, dan gas asap (MQ2), serta status indikator potensi kebakaran.

Setiap entri memiliki beberapa parameter utama yaitu:

- temperature: menunjukkan suhu udara yang terukur (dalam satuan Celcius),
- humidity: menunjukkan kelembapan relatif udara (dalam persen),
- gas_ppm: nilai konsentrasi gas/asap yang terdeteksi oleh sensor MQ2 (dalam satuan ppm),
- danger: status logika (true/false) yang menunjukkan apakah kondisi tersebut berpotensi menyebabkan kebakaran.
- Latitude dan longitude: menunjukan lokasi geografis dari sensor.
- timestamp_sensor: waktu pengambilan data oleh sensor
- timestamp_server: waktu saat data diterima oleh server Firebase.

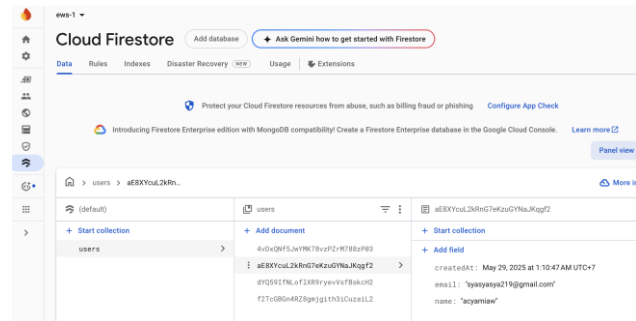
Penyimpanan data sensor secara real-time ini memungkinkan aplikasi untuk selalu menampilkan informasi terkini kepada pengguna dan mendeteksi secara cepat jika terjadi perubahan lingkungan yang signifikan, seperti peningkatan suhu, penurunan kelembapan, atau asap.



Gambar 4 Firebase Authentication

Firebase Authentication adalah layanan yang disediakan oleh Firebase untuk mengelola sistem login dan registrasi pengguna dengan aman dan efisien. Melalui fitur ini, pengembang dapat dengan mudah menangani autentikasi berbasis email dan password, serta memantau aktivitas pengguna yang telah mendaftar dan masuk ke dalam sistem.

Setiap pengguna memiliki informasi seperti alamat email, metode autentikasi yang digunakan, tanggal akun dibuat, tanggal terakhir pengguna login ke sistem, dan User UID yang merupakan ID pengguna yang dibuat secara otomatis oleh Firebase.



Gambar 5 Firebase Cloud Firestore

Cloud Firestore merupakan penyimpanan data dalam bentuk dokumen dan koleksi secara terstruktur serta mendukung sinkronisasi data real-time ke berbagai perangkat. Firestore digunakan untuk menyimpan data pengguna yang telah melakukan proses registrasi. Koleksi utama diberi nama users, dan setiap dokumen di dalamnya merepresentasikan satu akun pengguna. Penyimpanan data pengguna di Firestore ini memungkinkan aplikasi untuk menampilkan informasi profil pengguna, seperti nama atau email, saat pengguna sudah login.

IV. HASIL DAN PEMBAHASAN

Pengujian aplikasi "FireGuard" menggunakan black box testing dan System Usability Scale (SUS), dengan hasil sebagai berikut:

A. Hasil Pengujian Blackbox Testing

Blackbox testing adalah metode pengujian yang berfokus pada pemeriksaan fungsionalitas aplikasi tanpa perlu melihat dan menguji source code pada program[4]. Pada pengujian ini berfungsi untuk memastikan fitur yang ada di dalam aplikasi FireGuard berjalan dengan baik. Pengujian ini dilakukan dengan menjalankan setiap fitur yang ada pada aplikasi FireGuard. Dari hasil black box testing menunjukan bahwa semua fitur dalam aplikasi FireGuard berjalan dan berfungsi dengan yang diharapkan. Tidak ada kesalahan yang ditemukan selama pengujian berlangsung.

Test Case	Langkah Pengujian	Expected Result	Actual Result	Status
Login Berhasil	Pengguna memasukkan email dan password yang benar	Pengguna berhasil masuk ke Home Screen	Pengguna berhasil masuk ke Home Screen	Sukses
Reset Password	Pengguna memasukkan email terdaftar	Pengguna mendapatkan link email untuk verifikasi dan mengisi password baru	Pengguna mendapatkan link email untuk verifikasi dan mengisi password baru	Sukses
Login dengan Password Baru	Pengguna memasukkan email dan password yang baru	Pengguna berhasil masuk ke Home Screen	Pengguna berhasil masuk ke Home Screen	Sukses
Sign Up Berhasil	Pengguna mengisi email dan password	Pengguna mendapatkan link verifikasi dan berhasil ke halaman login	Pengguna mendapatkan link verifikasi dan berhasil ke halaman login	Sukses

Test Case	Langkah Pengujian	Expected Result	Actual Result	Status
Home Screen	Pengguna telah memiliki akun dan berhasil login	Pengguna dapat mengakses <i>Home Screen</i>	Pengguna dapat mengakses <i>Home Screen</i>	Sukses
Home Screen	Pengguna dapat melihat peta letak sensor nilai dari sensor suhu, asap, dan kelembapan secara <i>real-time</i>	Pengguna dapat melihat peta letak sensor nilai dari sensor suhu, asap, dan kelembapan secara <i>real-time</i>	Pengguna dapat melihat peta letak sensor nilai dari sensor suhu, asap, dan kelembapan secara <i>real-time</i>	Sukses
Status Hutan	Pengguna dapat melihat status hutan terindikasi kebakaran atau tidak	Pengguna dapat melihat status hutan terindikasi kebakaran atau tidak	Pengguna dapat melihat status hutan terindikasi kebakaran atau tidak	Sukses
CCTV Aktif	Pengguna dapat melihat pantauan CCTV secara <i>real-time</i>	Pengguna dapat melihat pantauan CCTV secara <i>real-time</i>	Pengguna dapat melihat pantauan CCTV secara <i>real-time</i>	Sukses

B. Hasil Pengujian System Usability Scale (SUS)

System Usability Scale (SUS) adalah metode evaluasi *usability* (efektivitas, efisiensi, dan kepuasan) yang dirancang untuk mengukur seberapa mudah suatu sistem digunakan oleh pengguna [5]. Pengguna akan menjawab 10 pertanyaan dengan skala 1 hingga 5, mulai dari sangat tidak setuju hingga sangat setuju. Tujuan dari pengujian ini untuk mengetahui seberapa mudah aplikasi digunakan oleh pengguna. Berikut daftar pertanyaan yang diberikan kepada responden:

No	Pertanyaan
1	Apakah desain tampilan antarmuka (UI) aplikasi ini menarik?
2	Apakah navigasi antar halaman di aplikasi terasa jelas dan mudah dipahami?
3	Apakah aplikasi berjalan dengan lancar di perangkat anda?
4	Apakah aplikasi merespons dengan cepat saat digunakan?
5	Apakah data sensor ditampilkan secara <i>real-time</i> sesuai kondisi?
6	Apakah notifikasi bahaya muncul saat kondisi darurat terdeteksi?
7	Apakah tampilan fitur CCTV dapat terlihat dengan jelas?

No	Pertanyaan
8	Apakah Anda merasa aplikasi ini memenuhi kebutuhan Anda dalam hal informasi kebakaran hutan?
9	Apakah anda merasa puas dengan performa aplikasi secara keseluruhan?
10	Apakah anda akan merekomendasikan aplikasi ini ke orang lain?

Terdapat 24 responden yang sudah memberikan penilaian terhadap performa dari aplikasi FireGuard melalui survei yang telah disediakan. Setiap pertanyaan dijawab berdasarkan skala 1 (Sangat Tidak Setuju), 2 (Tidak Setuju), 3 (Netral), 4 (Setuju), 5 (Sangat Setuju). Berikut adalah hasil survei dari para responden:

R	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q10
1	5	5	5	5	5	5	5	5	5	5
2	5	4	5	5	5	5	5	5	5	5
3	5	5	4	4	4	4	4	3	4	3
4	5	5	4	3	3	4	4	4	3	4
5	5	5	5	5	5	5	1	5	5	5
...	...	...	...	...	...	...	...	...	...	...
2										
4	5	5	5	5	5	5	5	5	5	5

Pada Tabel 2 memiliki keterangan R = Responden dan Q = Question. Untuk pertanyaan bernomor ganjil (1, 3, 5, 7, dan 9), skor kontribusi diperoleh dengan mengurangi nilai jawaban responden dengan angka 1. Untuk pertanyaan bernomor genap (2, 4, 6, 8, dan 10), skor kontribusi diperoleh dengan mengurangi angka 5 dengan nilai jawaban responden. Seluruh skor tersebut dijumlahkan dan dikalikan dengan 2,5 untuk mendapatkan skor SUS akhir dengan rentang 0–100.

$$\text{SUS Score} = \{(Q1-1)+(5-Q2)+(Q3-1)+(5-Q4)+(Q5-1) \\ + (5-Q6) + (Q7-1)+(5-Q8)+(Q9-1)+(5-Q10)\} \times 2.5$$

Dari hasil perhitungan SUS tersebut mendapatkan nilai akhir 79,16 yang menandakan bahwa aplikasi sudah berjalan dengan baik dan desain antarmuka yang mudah untuk digunakan.

V. KESIMPULAN

Aplikasi FireGuard dikembangkan menggunakan *Flutter* sebagai kerangka kerja utama untuk membangun antarmuka yang responsif serta *Firestore* sebagai layanan backend yang mendukung pengelolaan data dan autentikasi secara *real-time*. *Firestore Realtime Database* digunakan untuk menyimpan dan memperbarui data sensor secara langsung.

FireGuard dirancang dengan fitur utama meliputi pemantauan data sensor suhu, kelembapan, dan gas asap; penentuan status potensi kebakaran hutan; integrasi pantauan CCTV secara langsung (*live streaming*); serta pengiriman notifikasi peringatan dini kepada pengguna. Seluruh fitur ini saling terintegrasi dalam antarmuka yang intuitif untuk

memudahkan interaksi pengguna dan mempercepat proses pengambilan keputusan dalam situasi darurat.

Berdasarkan pengujian *System Usability Scale* (SUS), FireGuard memperoleh skor rata-rata 79,16 yang menempatkannya dalam kategori baik. Nilai ini menunjukkan bahwa aplikasi memiliki tingkat kegunaan yang tinggi, mudah dipahami, dan memenuhi kebutuhan pengguna dalam mendukung deteksi dini kebakaran hutan.

REFERENSI

- [1] Direktorat Pengendalian Kebakaran Hutan dan Lahan - Kementerian Lingkungan Hidup dan Kehutanan, “Jangan Biarkan Hutan Kita Terbakar, Lindungi Dengan Tindakan Nyata.”
- [2] Muslim *et al.*, “IMPLEMENTASI FRAMEWORK FLUTTER PADA SISTEM INFORMASI PERPUSTAKAAN MASJID,” 2022.
- [3] J. M. Suhendro, M. Sudarma, and D. Care Khrisne, “RANCANG BANGUN APLIKASI SELULER PENYEDIA JASA PERAWATAN DAN KECANTIKAN MENGGUNAKAN FRAMEWORK FLUTTER,” 2021.
- [4] S. Dika Pratama and M. Noviansyah Dadaprawira, “Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence Dan Boundary Value,” *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, vol. 6, no. 2, pp. 560–569, 2023, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsk/index>
- [5] E. Kurniawan, A. Nata, and S. Royal, “PENERAPAN SYSTEM USABILITY SCALE (SUS) DALAM PENGUKURAN KEBERGUNAAN WEBSITE PROGRAM STUDI DI STMIK ROYAL,” 2022. [Online]. Available: <http://jurnal.goretanpena.com/index.php/JSSR>