

Implementasi Aplikasi Android Pada *Smartdoor* Untuk Monitoring Dengan Menggunakan Firebase Sebagai Basis Data

1st Muhammad Azriel
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
mhmdazriel05@gmail.com

2nd Rendi Munadi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
rendimunadi@telkomuniversity.ac.id

3rd Iman Hedi Santoso
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
imanhedis@telkomuniversity.ac.id

Abstrak — Kejahatan merupakan fenomena sosial yang telah berlangsung sepanjang sejarah. Salah satunya adalah kejahatan dalam rumah, yang mengancam kenyamanan dan keamanan penghuni. Untuk mencegahnya, pendekatan preventif tradisional seperti penguatan sistem pengamanan fisik dan penguatan komunitas lokal telah diterapkan. Namun, perkembangan teknologi membuka peluang baru dalam mencegah kejahatan.

Dalam era teknologi yang berkembang pesat, solusi preventif semakin tergantung pada inovasi teknologi. Salah satu contohnya adalah penggunaan pintu pintar berbasis ESP-32 dengan fitur pengenalan wajah. Teknologi ini memungkinkan pintu hanya membuka diri saat wajah yang dikenali mendekat, mengurangi risiko akses ilegal. Dengan aplikasi terkait, pemilik rumah dapat memantau aktivitas di pintu depan secara jarak jauh melalui ponsel pintar, memberikan kontrol dan pemantauan yang lebih efektif.

Penelitian ini merangkum evolusi usaha pencegahan kejahatan dari pendekatan tradisional hingga pemanfaatan teknologi pintar. Implementasi aplikasi Android menggunakan teknologi Firebase dan ESP-32 telah diujikan untuk memastikan fungsionalitas dan performa yang baik. Hasil pengujian menunjukkan aplikasi ini dapat berjalan dengan lancar dan memenuhi harapan pengguna. Dengan pengembangan yang tepat, solusi teknologi seperti pintu pintar dapat menjadi sarana penting dalam menciptakan lingkungan yang aman dan nyaman bagi masyarakat.

Kata kunci— Aplikasi Android, *Smartdoor*, Kotlin, Firebase

I. PENDAHULUAN

Kejahatan merupakan suatu fenomena sosial yang telah ada sejak zaman dahulu. Kejahatan dapat merujuk pada berbagai tindakan yang melanggar hukum dan norma yang berlaku dalam suatu masyarakat. Kejahatan seringkali mencakup tindakan seperti pencurian, perampokan, penipuan, dan kekerasan fisik. Salah satu bentuk kejahatan yang sering terjadi adalah kejahatan dalam rumah, yang melibatkan intrusi ilegal ke dalam rumah seseorang untuk mencuri barang berharga atau melukai penghuninya. Kejahatan dalam rumah ini dapat berdampak serius terhadap kenyamanan dan rasa aman penghuni.

Dalam upaya mencegah kejahatan, terdapat berbagai usaha preventif yang dapat dilakukan tanpa melibatkan teknologi canggih. Salah satu usaha preventif yang umum dilakukan adalah memperkuat sistem pengamanan fisik

rumah, seperti memasang kunci ganda, pagar, dan pagar berduri. Selain itu, menjalin hubungan baik dengan tetangga dan mengadakan kegiatan komunitas lokal juga dapat meningkatkan pengawasan dan respons terhadap situasi yang mencurigakan.

Namun, dengan perkembangan teknologi yang semakin pesat, usaha preventif dalam mencegah kejahatan juga telah mengalami evolusi. Teknologi canggih seperti sistem pengamanan rumah pintar telah menjadi pilihan populer bagi banyak orang. Salah satu contoh penerapan teknologi ini adalah penggunaan *smartdoor* dengan memanfaatkan perangkat ESP-32 dan *deep learning* untuk pengenalan wajah. Teknologi ini memungkinkan pintu untuk terbuka hanya ketika wajah yang dikenali oleh sistem didekati, meminimalkan risiko akses tidak sah. Selain itu, dengan adanya aplikasi terkoneksi, pemilik rumah dapat memantau aktivitas di pintu depan secara jarak jauh melalui ponsel pintar mereka, memberikan tingkat kendali dan pemantauan yang lebih tinggi.

Secara keseluruhan, pencegahan kejahatan adalah upaya yang terus berkembang seiring dengan perubahan teknologi dan dinamika sosial. Mulai dari usaha preventif tradisional hingga teknologi modern seperti pintu pintar dengan pengenalan wajah, tujuan utamanya tetap sama: menciptakan lingkungan yang aman dan nyaman bagi individu dan masyarakat.

II. KAJIAN TEORI

Pada pengimplementasi aplikasi Android melibatkan *software* dan bahasa pemrograman untuk membantu pengembangan aplikasi. Selain itu dalam pengembangan aplikasi ini juga diperlukan *Cloud Computing* sebagai *back-end as a service*, *Cloud computing* sendiri adalah suatu konsep yang memungkinkan akses jaringan yang mudah dan fleksibel ke sumberdaya komputasi yang terdiri dari jaringan, *server*, penyimpanan, aplikasi dan layanan. Dalam pengimplementasian aplikasi ini penulis menggunakan beberapa *software* dan layanan *cloud* agar aplikasi dapat berjalan dengan baik, berikut *software* dan layanan *cloud* yang penulis gunakan:

A. Android Studio

Android Studio adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi

Android. Android Studio menyediakan berbagai fitur dan alat yang memudahkan pengembangan aplikasi, termasuk editor kode yang kuat, alat pemecah masalah, pengelola proyek, emulator Android, dan debugger. Android Studio dibangun di atas platform IntelliJ IDEA dari JetBrains dan dikembangkan oleh Google [1]. IDE ini menawarkan lingkungan pengembangan yang lengkap untuk membangun aplikasi Android, mulai dari merancang antarmuka pengguna dengan XML, menulis kode dengan Java atau Kotlin.

B. Kotlin

Kotlin adalah bahasa pemrograman modern yang dirancang dan berjalan pada platform Java Virtual Machine (JVM). Bahasa ini dikembangkan oleh JetBrains, perusahaan yang juga mengembangkan xliIJ IDEA, IDE yang digunakan oleh Android Studio[2].

C. Firebase Cloud Function

Cloud Function adalah kerangka tanpa server yang dapat menjalankan kode backend secara otomatis sebagai tanggapan terhadap peristiwa yang telah dipicu oleh fitur Firebase dan permintaan HTTPS. Cloud Functions juga merupakan salah satu layanan Google Cloud Platform[3].

D. Firebase Cloud Storage

Firebase Cloud Storage adalah layanan penyimpanan objek yang sederhana dan hemat biaya yang dibuat untuk skala Google. Cloud Storage dapat menyimpan gambar, file suara, video, atau konten lainnya dengan menggunakan API atau dengan menghubungkan langsung dengan aplikasi[4].

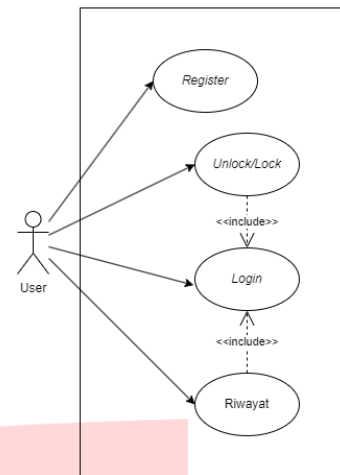
E. Firebase Cloud Messaging

Cloud Messaging adalah salah satu layanan kirim pesan notifikasi lintas platform yang dimiliki oleh Google Cloud Platform. Layanan Cloud Messaging ini dapat mengirimkan notifikasi atau pesan data ke semua user yang terdaftar dalam waktu yang bersamaan [5].

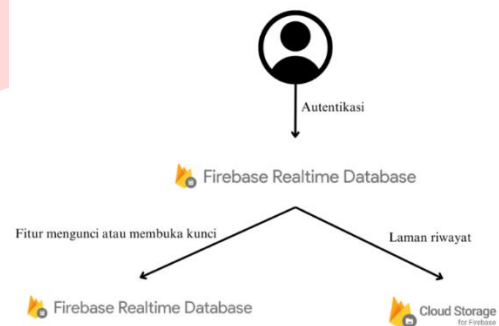
F. Firebase Realtime database

Firebase Realtime Database adalah layanan basisdata yang dihosting di cloud. Data disimpan dan disinkronasikan secara realtime dengan setiap user yang terhubung. Ketika menghubungkan aplikasi lintas platform dengan SDK platform Apple, Android, dan JavaScript Firebase dapat menggunakan satu instance realtime database yang sama dan menerima perubahan terbaru secara otomatis [4].

III. METODE



GAMBAR 1
(Diagram User Aplikasi Android)



GAMBAR 2
(Diagram Blok Feedback User)

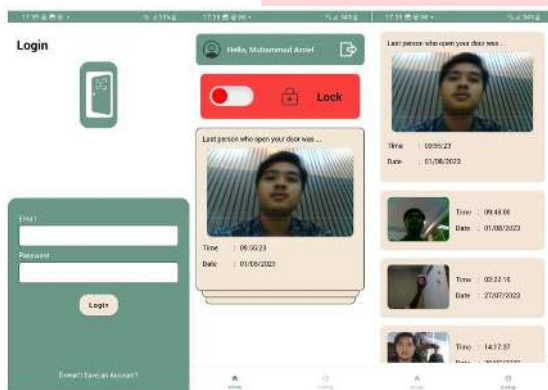
Aplikasi yang diimplementasikan akan memiliki *usecase diagram* seperti pada Gambar 1 yang menggambarkan interaksi antara user dan sistem. Agar user dapat menggunakan fitur membuka pintu, mengunci pintu dan melihat riwayat, user diharuskan untuk masuk (*Login*) ke akun pribadi mereka dengan menginputkan *email* dan kata sandi yang sudah didaftarkan pada laman *Register* aplikasi. Dapat dilihat pada Gambar 2 yang merupakan diagram blok, aplikasi harus dapat terhubung dengan Firebase, aplikasi memanfaatkan Firebase Realtime Database untuk mengautentikasi akun user dengan menyamakan inputan yang diinput pada laman *Login* dengan *database*. Aplikasi juga menggunakan Firebase Realtime Database untuk mengirimkan perintah membuka atau mengunci pintu pada aplikasi dengan mengubah nilai pada *database*. Cloud Storage Firebase digunakan untuk menyimpan citra-citra yang sudah dikirim ke *database*.

Aplikasi yang dirancang merupakan aplikasi berbasis sistem operasi android, dengan menggunakan bahasa pemrograman kotlin, aplikasi ini memiliki beberapa tampilan antarmuka yang diimplementasikan. Berikut rincian implementasi tampilan antarmuka pada aplikasi ini:

- Tampilan antarmuka *splashscreen*
- Tampilan antarmuka halaman pembuka
- Tampilan antarmuka halaman login
- Tampilan antarmuka halaman registrasi
- Tampilan antarmuka halaman utama
- Tampilan antarmuka halaman Riwayat



Gambar 3 (Antarmuka aplikasi)



GAMBAR 4 (Antarmuka aplikasi)

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini, penulis akan melakukan pengujian aplikasi android yang terbagi menjadi dua tahapan. Kedua tahapan pengujian akan menggunakan metode pengujian *black box*, dimana pengujian hanya melihat input kedalam sistem dan output yang dikeluarkan sistem.

Tahapan pertama adalah pengujian yang dilakukan oleh penulis untuk memastikan bahwa semua fitur dan fungsi dalam aplikasi dapat berjalan dengan baik. Selain itu, penguji juga melakukan pengujian integrasi untuk memverifikasi apakah aplikasi dapat berintegrasi dengan basis data. Sistem aplikasi android akan diuji dengan menggunakan gawai yang bersistem operasi android dan dalam proses pengujiannya gawai harus terhubung dengan internet agar aplikasi dapat berjalan dengan baik, pengujian akan dilaksanakan dengan menggunakan spesifikasi gawai sebagai berikut:

- a. CPU: Snapdragon 439
- b. GPU: Adreno 505
- c. Sistem Operasi: Android 10
- d. RAM: 2GB
- e. Memori: 16GB

Pada pengujian tahap kedua yaitu pengujian kepada para responden, responden akan terdiri dari berbagai lingkup seperti mahasiswa, pemilik kost, dan pemilik rumah yang sering meninggalkan rumah. Para responden diberikan akses untuk mencoba aplikasi dan memberikan umpan balik terkait fungsionalitas, penggunaan, dan kepuasan pengguna. Pengujian juga mencakup kompatibilitas dengan berbagai

perangkat Android. Dengan melibatkan para responden dalam pengujian, diharapkan dapat mendapatkan perspektif pengguna yang beragam untuk meningkatkan kualitas aplikasi android ini.

A. Pengujian Penulis Terhadap Aplikasi

Pada pengujian tahap pertama ini penulis melakukan pengujian dengan menggunakan gawai bersistem operasi android dan menggunakan metode *black box*. Berikut merupakan skenario tahap pertama:

TABEL 1. Pengujian Metode *Blackbox*

Skenarion Pengujian	Hasil yang didapatkan dan Sesuai yang Diharapkan
Pengguna melakukan pendaftaran akun baru dengan menekan tombol "Register"	Aplikasi dapat memuat laman Register
Pengguna masuk ke akun yang sudah di buat dengan menekan tombol "Login"	Aplikasi dapat memuat laman login
Pengguna melakukan pendaftaran dengan memasukan email, kata sandi dan nama. Kemudian menekan tombol "Register"	Aplikasi dapat memberi balasan "Register Berhasil"
Pengguna melakukan pendaftaran dengan email yang sudah terdaftar. Kemudian menekan tombol "Register"	Aplikasi dapat memberi balasan "Email sudah terdaftar"
Pengguna masuk dengan alamat email dan kata sandi yang valid dengan menekan tombol "Login"	Aplikasi dapat memuat laman utama
Pengguna masuk dengan alamat email atau kata sandi yang tidak valid dengan menekan tombol "Login"	Aplikasi dapat memberi balasan "email atau kata sandi salah"
Pengguna mencoba sebanyak tiga kali dengan menggunakan alamat email atau kata sandi yang tidak valid dengan menekan tombol "Login"	Aplikasi dapat melakukan sistem blokir sementara selama lima menit

<p>Pengguna melakukan percobaan mengunci atau membuka kunci dengan menekan tombol “Unlock/Lock”</p>	<p>Aplikasi dapat menampilkan berdasarkan input membuka kunci atau mengunci dan dapat menampilkan status terakhir pada basis data realtime</p>
<p>Pengguna menekan tombol “Riwayat”</p>	<p>Aplikasi dapat menampilkan data yang terdapat pada basis data komputasi awan</p>
<p>Ketika wajah terverifikasi oleh <i>machine learning</i>.</p>	<p>Aplikasi dapat memberikan pemberitahuan “Door opened successfully! Face detected and authorized entry via facial recognition”</p>
<p>Ketika wajah tidak terverifikasi oleh <i>machine learning</i></p>	<p>Aplikasi dapat memberikan pemberitahuan “Alert! Someone is at your door, tap here to view who enter your house and remotely unlock the door if needed”</p>

Dalam penelitian ini, akan dilakukan beberapa pengujian untuk mengevaluasi perfoma dan kehandalan dalam segi aplikasi android. Pengujian akan melibatkan simulasi sebagai pengguna dalam melakukan registrasi, login, fungsi tombol membuka atau mengunci, dan laman riwayat.

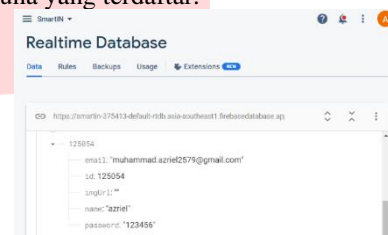
1. Pengujian Registrasi

Pengujian registrasi dilakukan dengan memasukan data seperti nama, alamat *email*, dan kata sandi pada aplikasi android.



GAMBAR 5 (Pengujian Registrasi)

Kemudian setelah di inputkan dalam aplikasi android, android akan langsung mengirimkan data ke *realtime database firebase* sebagai basis data utama untuk data pengguna yang terdaftar.

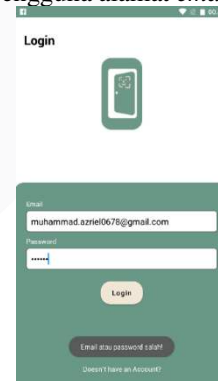


GAMBAR 6 (Tampilan RTDB Firebase)

2. Pengujian Login

Dalam pengujian fitur login ini penting untuk memastikan bahwa mekanisme login yang diimplementasikan pada aplikasi dapat berfungsi dengan baik dan dapat mengakses data pengguna yang sudah terdaftar dengan aman. Pengujian akan dilakukan dengan dua skenario dalam berbeda kondisi, seperti login yang berhasil dengan kombinasi data pengguna yang valid dan login yang gagal dengan kombinasi data pengguna yang tidak valid.

a. Skenario data pengguna alamat *email* tidak valid



GAMBAR 7 (Pengujian Alamat Email tidak Valid)

Maka aplikasi akan menerima respon pesan “Email atau Password salah!”.

b. Skenario data pengguna kata sandi tidak valid



GAMBAR 8
(Pengujian Data Pengguna Kata Sandi Tidak Valid)

Maka aplikasi akan menerima respon pesan “Email atau Password salah!”.

c. Skenario data pengguna valid

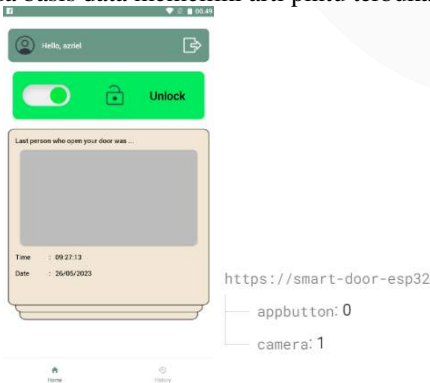


GAMBAR 9
(Pengujian Data Pengguna Valid)

Jika data pengguna yang dimasukan adalah data pengguna yang valid baik dari alamat *email* dan kata sandi, maka pengguna akan diberi akses masuk ke laman utama aplikasi.

a) Pengujian tombol membuka atau mengunci pintu

Pengujian terhadap tombol membuka atau mengunci pintu pada aplikasi android yang terhubung ke Firebase Realtime database ini memanfaatkan perubahan nilai pada basis data *realtime*. Dalam penelitian ini nilai 0 pada basis data memiliki arti pintu dalam kondisi terkunci sedangkan nilai 1 pada basis data memiliki arti pintu terbuka.



GAMBAR 10
(Tampilan Tombol Membuka dan mengunci Pintu)

b) Pengujian Laman Riwayat

Pada penelitian ini, hasil foto dari ESP-32 Cam ini akan disimpan pada *cloud storage* firebase sebagai basis data utama dalam hal ini. Sehingga akan terkumpul riwayat foto yang pernah tertangkap oleh ESP-32 Cam.



GAMBAR 11
(Pengujian Laman Riwayat)

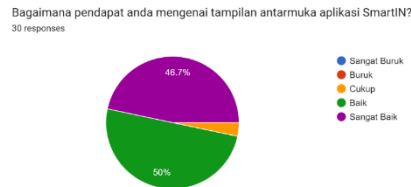
B. Pengujian Responden Terhadap Aplikasi

Pada skema pengujian yang kedua ini, penulis melakukan pengujian pengalaman pengguna dengan melibatkan sejumlah responden yang memiliki pengalaman menggunakan aplikasi sejenis. Para responden akan diberikan tugas untuk menggunakan aplikasi dan memberikan masukan terkait antarmuka pengguna dan kualitas keseluruhan aplikasi. Para responden terdiri dari berbagai lingkup seperti mahasiswa, pemilik kost, dan pemilik rumah yang sering meninggalkan rumah. Berikut merupakan hasil dari pengujian pengalaman pengguna terhadap 30 responden.



GAMBAR 12
(Pengujian Terhadap Responden)

Berdasarkan grafik diatas, dapat diketahui bahwa sebagian besar responden merupakan pengguna aplikasi *Smart home*, dengan total 93,3,% dari total responden.



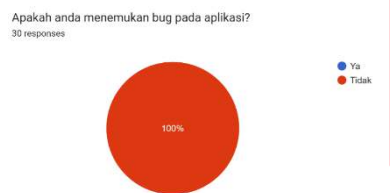
GAMBAR 13
(Pengujian Terhadap Responden)

Selanjutnya, dalam hal tampilan antarmuka aplikasi, grafik menunjukkan bahwa Sebagian besar responden memberikan penilaian “Baik” dengan total persentase sebanyak 50% dari total responden. Ini menunjukkan bahwa tombol navigasi aplikasi dianggap intuitif dan memudahkan pengguna dalam menemukan fitur-fitur yang mereka butuhkan.



GAMBAR 14
(Penggujian Terhadap Responden)

Grafik diatas mencerminkan tingkat kemudahan dalam penggunaan aplikasi SmartIN. Dalam hal ini, Sebagian besar responden memeberikan nilai “Sangat Mudah” sebanyak 60% dari total responden. Hal ini menunjukkan bahwa antarmuka aplikasi mudah dipahami dan digunakan oleh para pengguna.



GAMBAR 15
(Penggujian Terhadap Responden)

Dalam hal *bug* dalam aplikasi, sebanyak 100% dari para responden tidak menemukan *bug* baik dari segi fitur ataupun tampilan dari aplikasi



GAMBAR 16
(Penggujian Terhadap Responden)

Grafik diatas menjelaskan mengenai pendapat pengguna setelah mencoba aplikasi SmartIN, didapatkan sebanyak 56,7% dari total responden hal ini dapat disimpulkan bahwa aplikasi dapat berjalan dengan baik dari segi fungsionalitas maupun tampilan aplikasi.

Hasil dari pengujian dan penelitian ini diharapkan dapat memberikan rekomendasi dan saran untuk meningkatkan pengalaman pengguna pada aplikasi android, sehingga dapat meningkatkan kepuasan pengguna.

V. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan yaitu pengujian Black Box dan pengujian *User Experience* dapat disimpulkan bahwa aplikasi yang telah diimplementasikan berjalan dengan baik. Pengujian Black Box dilakukan untuk menguji fungsi-fungsi aplikasi tanpa memperhatikan detail implementasi atau kode sumbernya. Hasil pengujian menunjukkan bahwa aplikasi tersebut tidak mengalami kegagalan fungsi. Selain itu, pengujian *User Experience* melibatkan evaluasi pengalaman pengguna saat menggunakan aplikasi. Dari hasil pengujian ini, ditemukan bahwa antarmuka pengguna aplikasi intuitif, pengguna merasa terlibat, dan tingkat kepuasan pengguna tinggi. Pengembangan aplikasi ini berhasil menghasilkan produk yang memenuhi kebutuhan dan preferensi pengguna, serta memberikan pengalaman pengguna yang memuaskan.

REFERENSI

- [1] S. Periyanyagi, A. Manikandan, M. Muthukrishnan, and M. Ramakrishnan, “BDoor App-Blood Donation Application using Android Studio,” *J. Phys. Conf. Ser.*, vol. 1917, no. 1, 2021, doi: 10.1088/1742-6596/1917/1/012018.
- [2] S. Bose, “a Comparative Study: Java Vs Kotlin Programming in Android Application Development,” *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 3, pp. 41–45, 2018, doi: 10.26483/ijarcs.v9i3.5978.
- [3] M. Perron, R. Castro Fernandez, D. Dewitt, and S. Madden, “Starling: A Scalable Query Engine on Cloud Functions,” *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 131–141, 2020, doi: 10.1145/3318464.3380609.
- [4] C. Khawas and P. Shah, “Application of Firebase in Android App Development-A Study,” *Int. J. Comput. Appl.*, vol. 179, no. 46, pp. 49–53, 2018, doi: 10.5120/ijca2018917200.
- [5] A. Heryandi, “Developing Application ProgramMing Interface (API) for Student Academic Activity Monitoring using Firebase Cloud Messaging (FCM),” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 407, no. 1, 2018, doi: 10.1088/1757-899X/407/1/012149.