

Sistem Informasi Monitoring dan Kendali Perangkat Listrik Gedung Berbasis *Interactive Floorplan* Menggunakan ESP32

Building Electrical Device Monitoring and Control Information System Based on Interactive Floorplan Using ESP32

Bagas Fitriandra¹, Rudi Santoso², Hartono³

Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Duta Bangsa Surakarta

E-mail : bagasf06@gmail.com¹, rudi_susanto@udb.ac.id², hartono@udb.ac.id³

Received 14 June 2026; Revised 29 July 2026; Accepted 30 July 2026

Abstrak - Perkembangan teknologi *Internet of Things (IoT)* mendorong penerapan sistem monitoring dan kendali perangkat listrik pada lingkungan gedung secara *real-time*. Namun, sebagian besar sistem *smart building* masih menggunakan antarmuka berbasis daftar teks yang kurang efektif untuk memantau perangkat pada gedung dengan jumlah ruangan dan perangkat yang banyak. Penelitian ini bertujuan mengembangkan sistem informasi monitoring dan kendali perangkat listrik berbasis *Interactive Floorplan* menggunakan ESP32 dan *Firestore Real-time Database*. Sistem dikembangkan menggunakan aplikasi Android berbasis *Kotlin* dan *Jetpack Compose*, sedangkan *Firestore Real-time Database* digunakan sebagai media sinkronisasi data *real-time* antara aplikasi dan ESP32. Metode pengembangan yang digunakan adalah *Waterfall*. Pengujian fungsional menggunakan metode *Black Box Testing* menunjukkan seluruh fitur sistem berjalan sesuai kebutuhan. Hasil pengujian waktu respon menunjukkan rata-rata waktu respon sebesar 724 ms. Implementasi *Interactive Floorplan* memungkinkan visualisasi lokasi perangkat pada denah digital sehingga memudahkan proses monitoring dan pengendalian perangkat listrik.

Kata Kunci - *IoT*, ESP32, *Interactive Floorplan*, *Firestore*, Android

Abstract - The development of *Internet of Things (IoT)* technology has encouraged the implementation of *real-time* monitoring and control systems for electrical devices in building environments. However, most *smart building* systems still rely on text-based interfaces, which are less effective for managing numerous devices distributed across multiple rooms. This study aims to develop a building electrical device monitoring and control information system based on an *Interactive Floorplan* using ESP32 and *Firestore Realtime Database*. The system was developed as an Android application using *Kotlin* and *Jetpack Compose*, while *Firestore Realtime Database* was utilized for *real-time* data synchronization between the application and ESP32. The *Waterfall* model was employed as the system development methodology. Functional testing using the *Black Box Testing* method showed that all system features operated as expected. Response time testing produced an average latency of 724 ms. The implementation of the *Interactive Floorplan* enables device location visualization on a digital floor plan, thereby facilitating monitoring and control processes.

Keywords: *IoT*, ESP32, *Interactive Floorplan*, *Firestore*, Android

1. PENDAHULUAN

Perkembangan teknologi *Internet of Things (IoT)* telah mendorong transformasi sistem monitoring dan pengendalian perangkat listrik pada lingkungan gedung perkantoran maupun industri. Konsep *smart building* memungkinkan perangkat elektronik seperti lampu, kipas, dan peralatan listrik lainnya dikendalikan serta dipantau secara jarak jauh melalui jaringan internet sehingga meningkatkan efisiensi operasional dan pengelolaan energi. Implementasi sistem monitoring berbasis *IoT* juga dinilai mampu mengurangi konsumsi energi serta meminimalkan kesalahan manusia dalam pengoperasian perangkat listrik [1].

PT Solo Abadi Indonesia sebagai perusahaan manufaktur memiliki area operasional yang luas dengan berbagai perangkat listrik yang masih dikendalikan secara manual menggunakan saklar konvensional. Kondisi tersebut menyebabkan proses monitoring dan pengendalian perangkat menjadi kurang efisien karena petugas harus mendatangi lokasi perangkat secara langsung. Selain itu, penggunaan sistem manual meningkatkan risiko pemborosan energi akibat perangkat yang lupa dimatikan setelah jam operasional. Permasalahan ini menunjukkan perlunya sistem monitoring dan kendali perangkat listrik yang mampu diakses secara *real-time* dan terpusat.

Berbagai penelitian sebelumnya telah mengembangkan sistem monitoring dan kendali perangkat listrik berbasis *IoT* menggunakan mikrokontroler ESP32. ESP32 banyak digunakan karena memiliki konektivitas *Wi-Fi* terintegrasi, biaya implementasi rendah, dan kemampuan pemrosesan data secara *real-time* [2]. Penelitian lain menunjukkan bahwa integrasi *Firebase Real-time Database* pada sistem *IoT* mampu mendukung sinkronisasi data dengan latensi rendah sehingga cocok diterapkan pada aplikasi monitoring perangkat secara langsung [3].

Di sisi lain, sebagian besar penelitian sebelumnya masih menggunakan antarmuka berbasis daftar teks atau dashboard konvensional. Pendekatan tersebut memiliki keterbatasan dalam membantu pengguna mengidentifikasi posisi perangkat secara cepat, terutama pada lingkungan gedung dengan jumlah ruangan dan perangkat yang banyak. Visualisasi berbasis tabel atau daftar dinilai kurang intuitif karena pengguna harus mencari perangkat berdasarkan nama atau kategori tertentu [4].

Beberapa penelitian terkait visualisasi posisi perangkat dalam gedung umumnya memanfaatkan teknologi Indoor Positioning System (IPS) seperti *WiFi Positioning System* (WPS)[5]. Meskipun metode tersebut mampu mendukung pemetaan posisi perangkat secara otomatis, implementasinya membutuhkan perangkat tambahan dan biaya instalasi yang lebih tinggi. Kondisi tersebut menjadi kendala bagi perusahaan yang membutuhkan solusi monitoring dengan biaya implementasi yang lebih efisien.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan sistem informasi monitoring dan kendali perangkat listrik gedung berbasis *Interactive Floorplan* menggunakan ESP32 dan *Firebase Real-time Database*. Sistem yang dikembangkan memvisualisasikan posisi perangkat secara langsung pada denah digital gedung sehingga pengguna dapat melakukan monitoring dan pengendalian perangkat dengan lebih intuitif dan *real-time* melalui aplikasi Android native.

Kontribusi utama penelitian ini adalah pengembangan antarmuka monitoring berbasis denah digital interaktif untuk sistem kendali perangkat listrik berbasis *IoT* tanpa memerlukan sensor tambahan seperti *Bluetooth Beacon* maupun teknologi pelacakan posisi lainnya. Selain itu, penelitian ini menawarkan pendekatan visualisasi spasial perangkat yang lebih mudah dipahami pengguna dibandingkan dashboard konvensional berbasis daftar atau tabel.

2. METODE PENELITIAN

2.1. Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan sistem *Waterfall* karena proses pengembangan dilakukan secara bertahap, sistematis, dan terstruktur mulai dari analisis kebutuhan hingga evaluasi sistem. Metode *Waterfall* dinilai sesuai untuk pengembangan sistem monitoring berbasis *IoT* karena setiap tahapan dapat terdokumentasi dengan baik dan memudahkan proses pengujian sistem [6].

Tahapan penelitian terdiri dari:

1. Analisis kebutuhan sistem.

Tahap awal untuk mengidentifikasi kebutuhan fungsional dan non-fungsional di PT Solo Abadi Indonesia. Hal ini mencakup pemetaan perangkat listrik yang akan dikendalikan dan penentuan spesifikasi hardware ESP32 serta fitur aplikasi Android.

2. Perancangan sistem..
Melakukan perancangan arsitektur sistem, desain skematik rangkaian elektronika antara ESP32 dan relay, serta perancangan antarmuka Visual Positioning menggunakan *Jetpack Compose*. Desain basis data juga disusun menggunakan model NoSQL pada *Firebase Real-time Database*.
3. Implementasi.
Tahap penerjemahan desain ke dalam kode program. Implementasi dilakukan pada sisi perangkat keras menggunakan bahasa C++ (Arduino IDE) dan pada sisi perangkat lunak menggunakan bahasa *Kotlin* untuk membangun fitur *Interactive Floorplan*.
4. Pengujian.
Melakukan verifikasi terhadap sistem yang telah dibangun menggunakan metode *Black Box Testing* untuk memastikan seluruh fitur berjalan sesuai rencana. Pengujian juga mencakup pengukuran waktu respon kendali perangkat.
5. Evaluasi hasil penelitian.
Tahap akhir untuk menganalisis hasil pengujian dan memastikan bahwa sistem berbasis *Interactive Floorplan* telah berhasil mempermudah identifikasi lokasi perangkat dibandingkan metode konvensional.

2.2. Analisis Kebutuhan Sistem

Berdasarkan hasil observasi lapangan dan wawancara yang dilakukan dengan Direktur serta Kepala Departemen IT PT Solo Abadi Indonesia, diperoleh beberapa kebutuhan utama yang harus dipenuhi oleh sistem yang akan dikembangkan. Sistem diharapkan mampu melakukan monitoring status perangkat listrik secara *real-time* sehingga pengguna dapat mengetahui kondisi perangkat secara langsung tanpa harus melakukan pengecekan ke lokasi. Selain itu, sistem harus menyediakan fitur kendali perangkat listrik melalui aplikasi Android untuk memudahkan proses pengoperasian dari jarak jauh. Sistem juga perlu menampilkan posisi perangkat pada denah digital gedung agar pengguna dapat dengan mudah mengidentifikasi lokasi setiap perangkat secara visual. Untuk mendukung fleksibilitas penggunaan, sistem harus dapat diakses melalui jaringan internet sehingga proses monitoring dan pengendalian dapat dilakukan dari berbagai lokasi. Selain aspek fungsional, sistem juga diharapkan memiliki antarmuka yang mudah dipahami dan digunakan oleh pengguna guna meningkatkan efektivitas operasional serta meminimalkan kesalahan dalam pengelolaan perangkat listrik.

2.3. Arsitektur Sistem

Perancangan sistem dilakukan untuk menggambarkan hubungan antar komponen yang terlibat dalam proses monitoring dan kendali perangkat listrik berbasis *Internet of Things*. Sistem yang dikembangkan terdiri dari tiga komponen utama yaitu aplikasi Android, *Firebase Real-time Database*, dan perangkat ESP32. Perancangan sistem meliputi diagram blok sistem, flowchart sistem, diagram pengkabelan perangkat keras, serta diagram interface.

2.3.1. Diagram Blok Sistem



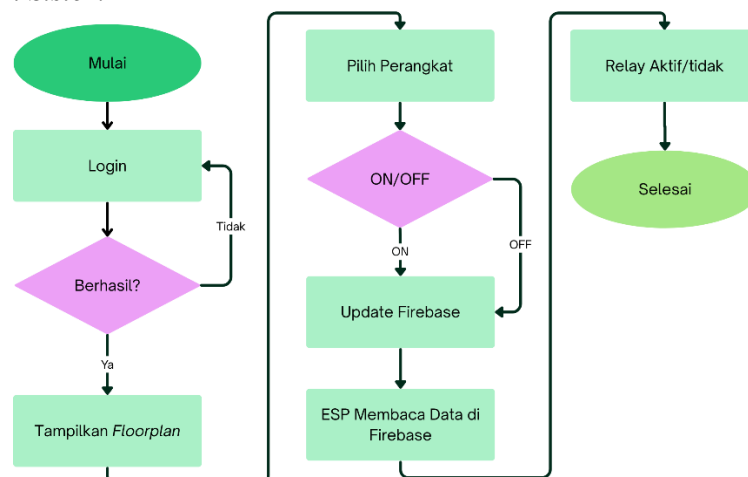
Gambar 1. Diagram Blok Sistem.

Gambar 1 merupakan diagram blok sistem yang terdiri dari aplikasi Android, *Firebase Realtime Database*, ESP32, modul relay, dan perangkat listrik sebagai beban. Komponen-komponen tersebut saling terhubung melalui jaringan internet untuk mendukung proses monitoring dan kendali perangkat listrik secara *real-time*. Aplikasi Android digunakan sebagai antarmuka bagi pengguna untuk melakukan monitoring dan pengendalian perangkat listrik melalui fitur *Interactive Floorplan*, sedangkan *Firebase Realtime Database* berfungsi sebagai

media pertukaran dan sinkronisasi data antara aplikasi Android dan perangkat ESP32 secara *real-time*[7].

ESP32 berperan sebagai mikrokontroler utama yang terhubung ke jaringan internet melalui *Wi-Fi*. Perangkat ini membaca perubahan data yang tersimpan pada *Firebase Realtime Database* dan menerjemahkannya menjadi perintah kendali untuk mengaktifkan atau menonaktifkan modul relay sesuai tindakan pengguna pada aplikasi Android [8]. Selain menjalankan fungsi kendali, ESP32 juga mengirimkan status perangkat kembali ke *Firebase* sehingga kondisi perangkat dapat ditampilkan secara *real-time* pada aplikasi Android. Dengan mekanisme tersebut, proses monitoring dan pengendalian perangkat listrik dapat dilakukan dari jarak jauh melalui jaringan internet.

2.3.2. Flowchart Sistem

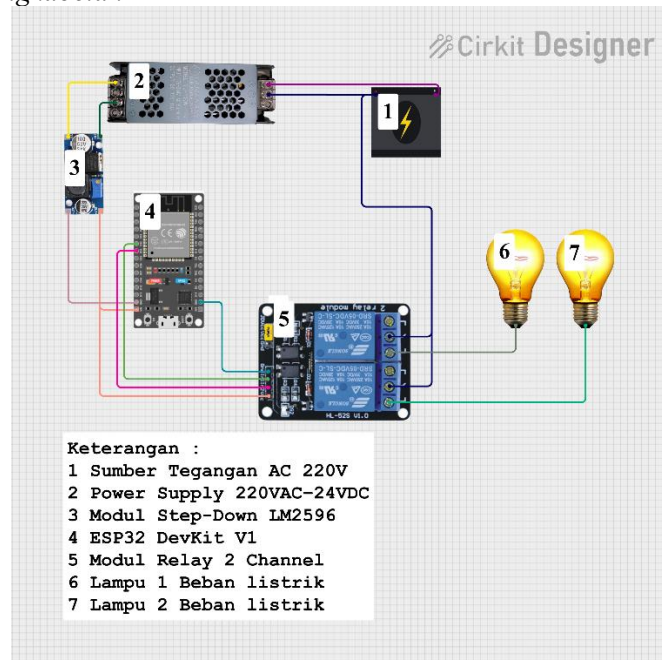


Gambar 2. Flowchart sistem

Gambar 2 menggambarkan alur kerja sistem monitoring dan kendali perangkat listrik mulai dari proses autentikasi pengguna hingga proses pengendalian perangkat. Proses diawali dengan pengguna melakukan login ke aplikasi Android menggunakan akun yang telah terdaftar. Setelah proses autentikasi berhasil, aplikasi mengambil data perangkat dari *Firebase Real-time Database* dan menampilkan informasi perangkat beserta posisinya pada denah digital gedung.

Pengguna selanjutnya dapat melakukan monitoring status perangkat atau memilih node perangkat pada *Interactive Floorplan* untuk mengirimkan perintah kendali ON atau OFF. Perintah tersebut dikirimkan ke *Firebase Real-time Database* dan diteruskan ke ESP32 melalui jaringan internet. ESP32 kemudian memproses data yang diterima dan mengendalikan relay sesuai perintah pengguna. Setelah proses kendali dilakukan, ESP32 mengirimkan status perangkat terbaru ke *Firebase* sehingga aplikasi Android dapat menampilkan kondisi perangkat secara *real-time*.

2.3.3. Diagram Pengkabelan



Gambar 3. Diagram Pengkabelan ESP32 dan Relay untuk Kendali Perangkat Listrik

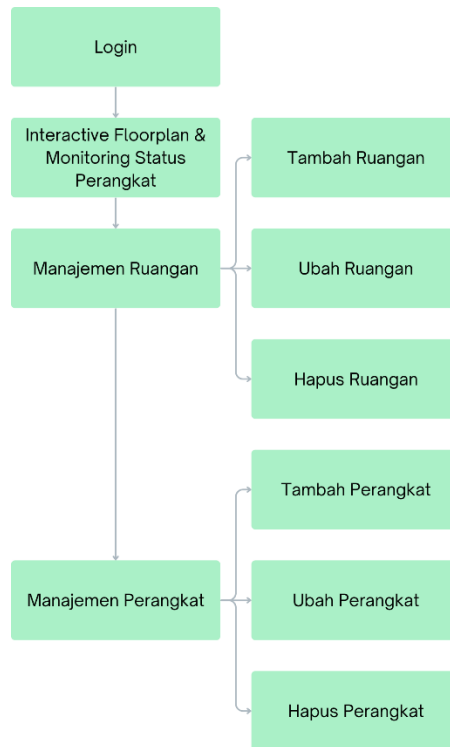
Gambar 3 menunjukkan konfigurasi hubungan antara sumber tegangan, modul catu daya, ESP32, modul relay, dan beban listrik yang digunakan pada sistem. Sumber tegangan AC 220V dikonversi menjadi tegangan DC 24V menggunakan power supply switching. Tegangan tersebut kemudian diturunkan menjadi 5V menggunakan modul step-down LM2596 untuk mensuplai ESP32 dan modul relay.

ESP32 berfungsi sebagai mikrokontroler utama yang terhubung ke modul relay melalui pin GPIO sebagai sinyal kendali digital. Ketika ESP32 menerima perubahan data dari *Firestore Real-time Database*, mikrokontroler akan memproses data tersebut dan mengaktifkan atau menonaktifkan relay sesuai perintah yang diberikan pengguna melalui aplikasi Android. Relay kemudian berfungsi sebagai saklar elektronik yang menghubungkan atau memutus aliran listrik menuju beban berupa lampu. Dengan konfigurasi ini, proses monitoring dan kendali perangkat listrik dapat dilakukan secara *real-time* melalui jaringan internet.

2.3.4. Diagram Interface

Gambar 4 menggambarkan alur navigasi antar halaman pada aplikasi Android yang digunakan untuk proses monitoring dan kendali perangkat listrik. Aplikasi diawali dengan halaman login yang berfungsi untuk autentikasi pengguna. Setelah proses login berhasil, pengguna akan diarahkan menuju halaman utama yang menampilkan *Interactive Floorplan* beserta status perangkat listrik secara *real-time*.

Dari halaman utama tersebut, pengguna dapat memilih node perangkat untuk melihat informasi perangkat maupun melakukan proses kendali ON/OFF. Selain itu, aplikasi juga menyediakan halaman manajemen ruangan dan manajemen perangkat untuk mendukung pengelolaan data perangkat listrik berdasarkan lokasi ruangan. Diagram interface digunakan untuk memberikan gambaran mengenai struktur navigasi aplikasi dan hubungan antar halaman yang tersedia pada sistem.



Gambar 4. Diagram Interface

2.4. Implementasi Interactive Floorplan

Interactive Floorplan dikembangkan menggunakan *Jetpack Compose* pada platform Android untuk menyediakan visualisasi perangkat listrik secara interaktif pada denah digital gedung[9]. Setiap perangkat direpresentasikan dalam bentuk node yang memiliki koordinat dua dimensi (x,y) yang disimpan pada *Firebase Realtime Database*. Koordinat tersebut digunakan untuk menentukan posisi node pada denah digital melalui mekanisme pengaturan offset yang disediakan oleh *Jetpack Compose*. Pengguna dapat memindahkan posisi node melalui mekanisme *drag-and-drop*. Setelah proses pemindahan selesai, koordinat terbaru disimpan kembali ke *Firebase Realtime Database* sehingga tata letak perangkat dapat diperbarui secara dinamis sesuai kebutuhan pengguna. Melalui antarmuka ini, pengguna dapat melakukan proses monitoring dan kendali perangkat secara langsung dengan memilih node yang ditampilkan pada denah digital.

Algoritma 1. Penempatan dan Pembaruan Posisi Node pada Interactive Floorplan

```

Input : Data node perangkat (id, x, y)
Output : Posisi node pada denah digital

Ambil data node dari Firebase Realtime Database.

FOR setiap node perangkat DO
  Tampilkan node pada koordinat (x, y).

  IF mode edit aktif THEN
    IF pengguna melakukan drag THEN
       $x \leftarrow x + \Delta x$ 
       $y \leftarrow y + \Delta y$ 
    END IF

    IF proses drag selesai THEN

```

```

        Simpan koordinat terbaru (x, y)
        ke Firebase Realtime Database.
    END IF
END IF
END FOR

Tampilkan kembali posisi node
berdasarkan koordinat terbaru.

Keterangan :
•  $x, y$  = koordinat awal node
•  $\Delta x, \Delta y$  = perpindahan posisi akibat proses drag
    
```

Berdasarkan Algoritma 1, sistem terlebih dahulu mengambil data koordinat setiap node dari Firebase Realtime Database untuk ditampilkan pada denah digital melalui mekanisme *offset* yang disediakan oleh *Jetpack Compose*. Ketika mode edit diaktifkan, pengguna dapat memindahkan posisi node melalui mekanisme *drag-and-drop*. Selama proses tersebut, koordinat node diperbarui berdasarkan perpindahan posisi pengguna, kemudian disimpan kembali ke Firebase Realtime Database setelah proses *drag* selesai sehingga posisi node tetap tersimpan dan digunakan kembali pada saat aplikasi dijalankan berikutnya.

2.5. Pengujian Sistem

Pengujian sistem dilakukan melalui beberapa tahap berikut:

1. Pengujian Fungsional.
Pengujian dilakukan menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan pengguna. Pengujian mencakup fungsi login, monitoring perangkat, kendali perangkat, sinkronisasi data *real-time*, dan tampilan *Interactive Floorplan*.
2. Pengujian Waktu Respon.
Pengujian waktu respon dilakukan dengan menghitung selisih waktu antara pengiriman perintah dari aplikasi Android hingga relay pada ESP32 aktif. Pengujian dilakukan sebanyak beberapa kali percobaan untuk memperoleh nilai rata-rata latensi sistem.
3. Pengujian Sinkronisasi *Real-time*.
Pengujian dilakukan untuk memastikan perubahan data pada *Firebase Real-time Database* dapat diterima secara *real-time* oleh aplikasi Android dan ESP32 tanpa terjadi keterlambatan sinkronisasi yang signifikan.
4. Pengujian Tampilan *Interactive Floorplan*.
Pengujian dilakukan untuk memastikan posisi node perangkat ditampilkan sesuai koordinat yang tersimpan pada Firebase Realtime Database serta dapat diperbarui melalui mekanisme *drag-and-drop*.

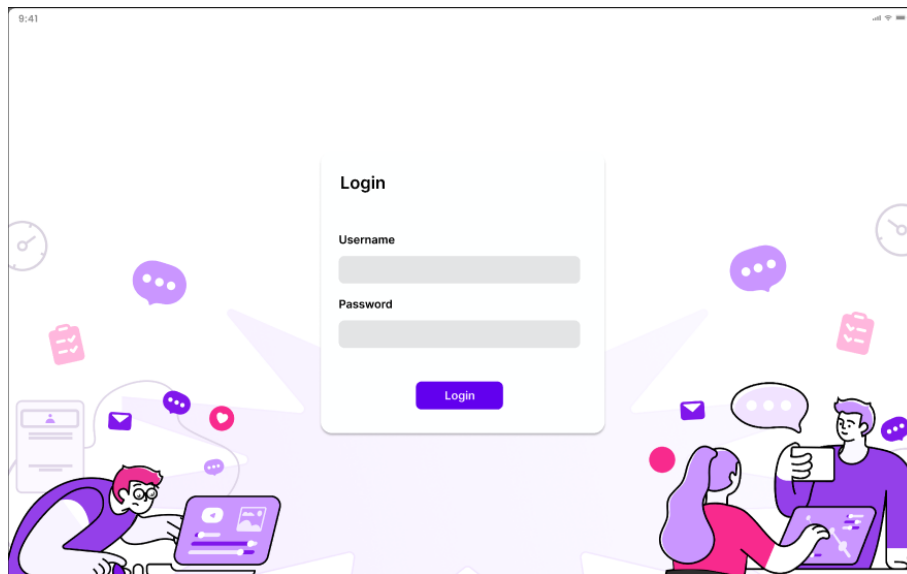
3. HASIL DAN PEMBAHASAN

3.1. Hasil Implementasi Sistem

Hasil implementasi menunjukkan aplikasi *Android* berhasil menampilkan denah digital gedung beserta *node* perangkat listrik yang dapat dikendalikan secara *real-time*.

Sistem memiliki beberapa fitur utama sebagai berikut:

1. *Login* pengguna



Gambar 5. Halaman Login

Gambar 5 menunjukkan halaman login yang digunakan sebagai mekanisme autentikasi pengguna sebelum mengakses sistem. Fitur ini bertujuan untuk memastikan bahwa hanya pengguna yang memiliki hak akses yang dapat melakukan proses monitoring dan pengendalian perangkat listrik. Pada halaman ini, pengguna diwajibkan memasukkan alamat email dan kata sandi yang telah terdaftar pada sistem. Setelah proses autentikasi berhasil dilakukan, pengguna akan diarahkan ke halaman utama aplikasi yang menampilkan *Interactive Floorplan* beserta informasi status perangkat listrik secara *real-time*.

2. Monitoring status perangkat dan *Interactive Floorplan*

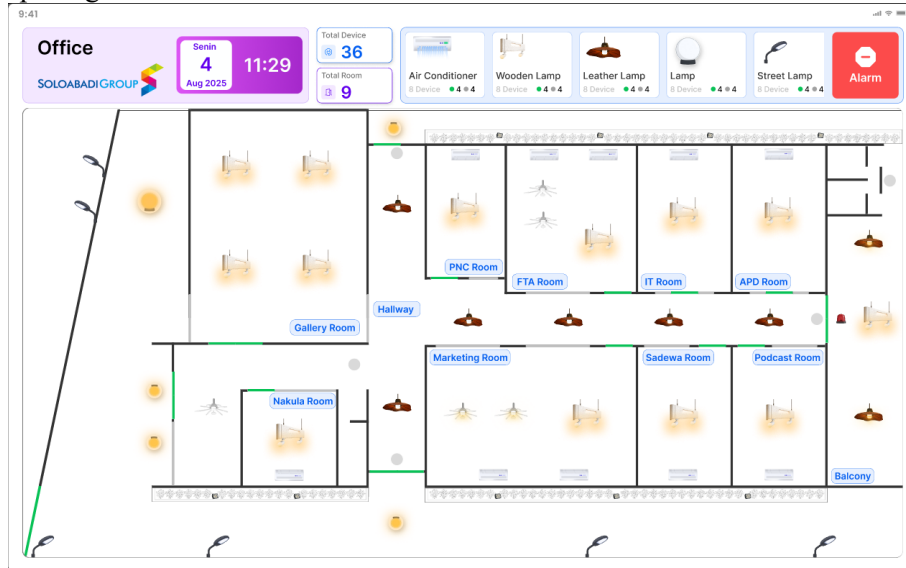


Gambar 6. Halaman *Interactive Floorplan*

Gambar 6 menunjukkan halaman utama aplikasi yang menampilkan kondisi perangkat listrik melalui *Interactive Floorplan*. Setiap perangkat direpresentasikan dalam bentuk node yang ditempatkan sesuai dengan posisi perangkat pada denah gedung. Node perangkat dilengkapi indikator visual untuk menunjukkan status aktif atau nonaktif sehingga pengguna dapat dengan mudah memantau kondisi perangkat secara langsung.

Informasi status perangkat diperbarui secara *real-time* melalui sinkronisasi data menggunakan *Firestore Real-time Database*. Dengan mekanisme tersebut, setiap perubahan status yang terjadi pada perangkat akan langsung ditampilkan pada aplikasi Android. Selain berfungsi sebagai media monitoring, halaman ini juga menjadi pusat interaksi pengguna untuk memilih perangkat yang akan dikendalikan melalui denah digital.

3. Kendali perangkat on/off



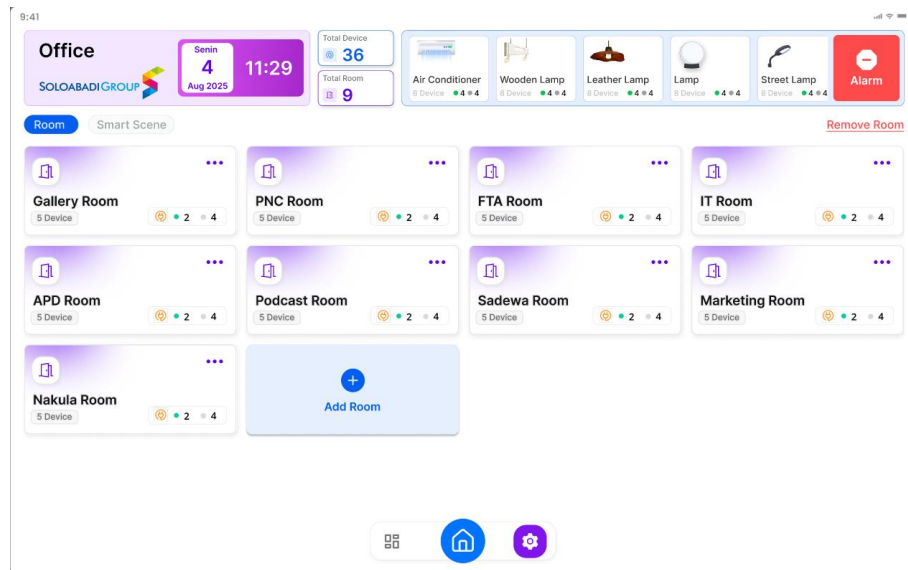
Gambar 7. Kendali Perangkat pada *Interactive Floorplan*

Gambar 7 menunjukkan fitur kendali perangkat yang memungkinkan pengguna mengaktifkan atau menonaktifkan perangkat listrik secara langsung melalui aplikasi Android. Proses kendali dilakukan dengan memilih atau menekan node perangkat pada denah digital yang ditampilkan pada *Interactive Floorplan*. Setelah node dipilih, sistem akan mengirimkan perintah kendali ON atau OFF sesuai tindakan pengguna.

Perintah yang dikirimkan dari aplikasi Android diteruskan melalui *Firestore Real-time Database* menuju ESP32 melalui jaringan internet. Selanjutnya, ESP32 memproses data yang diterima dan mengendalikan modul relay untuk menghubungkan atau memutus aliran listrik pada perangkat yang dikendalikan. Setelah proses kendali berhasil dilakukan, status perangkat diperbarui kembali ke *Firestore Real-time Database* sehingga perubahan kondisi perangkat dapat ditampilkan secara *real-time* pada aplikasi Android.

4. Manajemen Ruangan

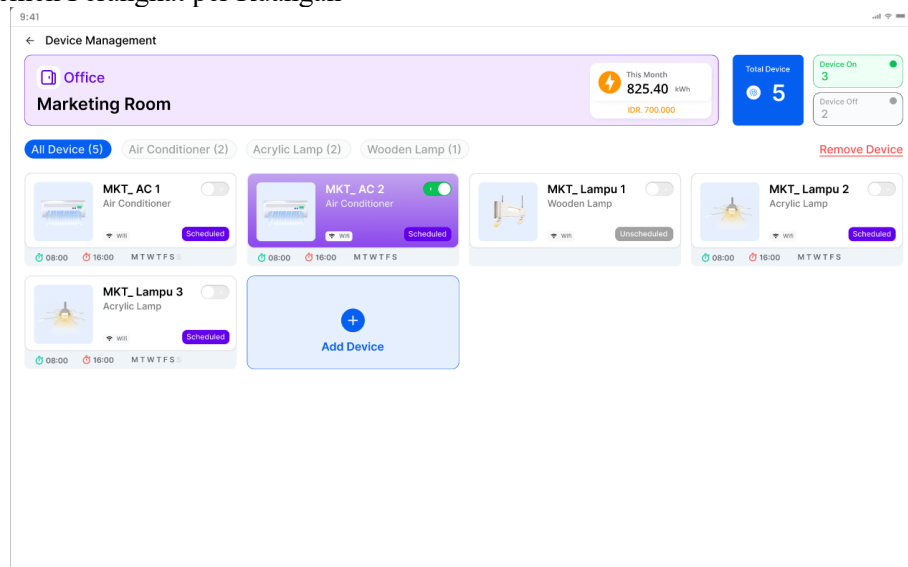
Gambar 8 menunjukkan fitur manajemen ruangan yang digunakan untuk mengelola data ruangan pada denah digital gedung. Melalui fitur ini, pengguna dapat menambahkan, mengubah, dan menghapus data ruangan sesuai dengan kebutuhan pengelolaan fasilitas gedung. Informasi ruangan yang tersimpan digunakan sebagai dasar dalam proses pemetaan perangkat pada *Interactive Floorplan*.



Gambar 8. Halaman Manajemen Ruangan

Fitur manajemen ruangan memudahkan pengguna dalam mengatur struktur gedung secara terorganisir sehingga setiap perangkat dapat dikelompokkan berdasarkan lokasi ruangan yang sesuai. Dengan adanya pengelompokan tersebut, proses monitoring dan pengendalian perangkat listrik menjadi lebih mudah dilakukan karena pengguna dapat mengidentifikasi lokasi perangkat berdasarkan ruangan tempat perangkat berada.

5. Manajemen Perangkat per Ruangan



Gambar 9. Halaman Manajemen Perangkat

Gambar 9 menunjukkan fitur manajemen perangkat yang digunakan untuk mengelola data perangkat listrik pada setiap ruangan. Melalui fitur ini, pengguna dapat melakukan penambahan, pengubahan, dan penghapusan data perangkat sesuai dengan kebutuhan sistem. Setiap perangkat yang terdaftar memiliki informasi seperti nama perangkat, status perangkat, serta lokasi penempatan pada ruangan tertentu.

Fitur manajemen perangkat memungkinkan setiap perangkat dikaitkan dengan ruangan yang telah terdaftar pada sistem sehingga proses monitoring dan pengendalian dapat dilakukan berdasarkan lokasi perangkat. Selain itu, data perangkat yang tersimpan akan digunakan sebagai acuan dalam menampilkan node perangkat pada *Interactive Floorplan*. Dengan demikian, perubahan data perangkat dapat langsung tercermin pada tampilan denah digital dan mendukung proses pengelolaan perangkat listrik secara lebih terstruktur.

Implementasi antarmuka *Interactive Floorplan* menggunakan *Jetpack Compose* menghasilkan tampilan interaktif yang mampu memvisualisasikan posisi setiap perangkat pada denah digital berdasarkan koordinat yang tersimpan pada *Firebase Realtime Database* [10]. Setiap node perangkat ditampilkan sesuai koordinat yang telah ditentukan, sehingga representasi lokasi perangkat pada denah digital dapat ditampilkan secara konsisten. Selain itu, mekanisme *drag-and-drop* memungkinkan pengguna memindahkan posisi node sesuai kebutuhan, kemudian menyimpan koordinat terbaru ke *Firebase Realtime Database* sehingga perubahan tata letak dapat dipertahankan pada penggunaan berikutnya. Melalui pendekatan tersebut, pengguna dapat melakukan monitoring dan pengendalian perangkat secara lebih intuitif melalui representasi visual pada denah digital.

3.2. Hasil Pengujian Waktu Respon

Pengujian waktu respons dilakukan untuk mengetahui waktu yang diperlukan sistem dalam meneruskan perintah perubahan status dari aplikasi Android hingga perangkat ESP32 merespons perintah tersebut. Saat pengguna menekan tombol kendali, aplikasi mencatat timestamp pengiriman perintah menggunakan *System.currentTimeMillis()* dan menyimpannya pada *Firebase Realtime Database* bersama perubahan status perangkat. ESP32 memantau perubahan data tersebut melalui listener Firebase, kemudian menjalankan perubahan kondisi relay dan mencatat *timestamp* respons perangkat pada *Firebase*. Waktu respons dihitung berdasarkan selisih antara *timestamp* respons ESP32 dan timestamp pengiriman perintah dari aplikasi, dengan persamaan $T_{respons} = T_{device} - T_{app}$. Pengujian dilakukan sebanyak lima kali dengan perubahan kondisi ON dan OFF untuk memperoleh nilai rata-rata waktu respons sistem. Hasil rincian waktu respon sistem dapat dilihat pada Tabel 1.

Tabel 1. Hasil pengujian waktu respon

Pengujian	Kondisi	Waktu Respon (ms)
1	OFF	899
2	ON	668
3	OFF	730
4	ON	636
5	OFF	687
Rata-rata		724 ms

Berdasarkan hasil pengujian, diperoleh rata-rata waktu respons sistem sebesar **724 ms**. Nilai tersebut menunjukkan bahwa sistem mampu meneruskan perintah perubahan status dari aplikasi Android hingga ESP32 mengubah kondisi relay dalam waktu kurang dari satu detik. Waktu respons tersebut dipengaruhi oleh beberapa faktor, antara lain stabilitas koneksi internet, proses sinkronisasi data pada *Firebase Realtime Database*, serta waktu pemrosesan data pada mikrokontroler ESP32[11].

3.3. Hasil Pengujian Fungsional

Pengujian fungsional pada sistem ini dilakukan dengan menggunakan metode *Black Box Testing* untuk memastikan seluruh fitur berjalan sesuai dengan kebutuhan pengguna tanpa melihat proses internal atau struktur kode program. Fokus pengujian ini diarahkan pada fungsionalitas utama aplikasi Android, seperti proses otentikasi login, akurasi pemetaan node pada denah digital, serta keberhasilan pengiriman perintah kendali On/Off ke perangkat keras. Melalui pendekatan

ini, validasi sistem didasarkan pada kesesuaian antara input yang diberikan oleh pengguna dan output fisik yang dihasilkan oleh relai pada mikrokontroler [12]. Hasil pengujian fungsional sistem ditunjukkan pada Tabel 2.

Tabel 2. Hasil pengujian fungsional sistem.

No	Fitur Sistem	Hasil Pengujian	Status
1	Login pengguna	Berhasil menampilkan halaman utama	Valid
2	Monitoring perangkat	Status perangkat tampil <i>real-time</i>	Valid
3	Kendali ON/OFF perangkat	Relay berhasil aktif/nonaktif	Valid
4	Sinkronisasi Firebase	Data tersinkron secara <i>real-time</i>	Valid
5	<i>Interactive Floorplan</i>	Posisi node sesuai denah	Valid

Berdasarkan hasil pengujian fungsional yang telah dilakukan menggunakan metode *Black Box Testing*, seluruh fitur sistem dinyatakan dapat berjalan dengan baik sesuai dengan kebutuhan fungsional yang dirancang pada tahap analisis sistem. Validasi ini dibuktikan melalui keberhasilan eksekusi pada seluruh skenario uji, termasuk fungsionalitas autentikasi pengguna, pemetaan koordinat node perangkat, hingga sinkronisasi data kendali antara aplikasi Android dan mikrokontroler ESP32. Dengan demikian, sistem ini telah memenuhi spesifikasi teknis yang diharapkan dan siap digunakan untuk mendukung digitalisasi pengelolaan fasilitas gedung di PT Solo Abadi Indonesia.

3.4. Pembahasan

Berdasarkan hasil pengujian, sistem monitoring dan kendali perangkat listrik berbasis *Interactive Floorplan* berhasil diimplementasikan dengan mengintegrasikan mikrokontroler ESP32, *Firebase Realtime Database*, dan aplikasi *Android native*. Melalui pemanfaatan jaringan internet, sistem ini mampu melakukan pemantauan status serta proses kendali perangkat listrik secara *real-time* dengan rata-rata waktu respons sebesar 724 ms. Pendekatan visualisasi spasial dalam bentuk *node* di atas denah digital ini memberikan kemudahan bagi petugas di PT Solo Abadi Indonesia untuk mengidentifikasi lokasi fisik perangkat secara lebih intuitif dibandingkan menggunakan antarmuka daftar teks konvensional.

Hasil pengujian waktu respons menunjukkan rata-rata waktu respons sebesar 724 ms. Nilai tersebut menunjukkan bahwa komunikasi data antara aplikasi Android dan perangkat ESP32 berlangsung dalam waktu kurang dari satu detik untuk kebutuhan monitoring dan kendali perangkat listrik pada lingkungan gedung. Hasil penelitian ini sejalan dengan penelitian yang dilakukan oleh Harlan Kurnia[13] yang menyatakan bahwa *Firebase Real-time Database* memiliki kemampuan sinkronisasi data dengan latensi rendah pada implementasi sistem *IoT* berbasis Android.

Implementasi *Interactive Floorplan* memberikan peningkatan dari sisi visualisasi monitoring perangkat dibandingkan pendekatan *dashboard* konvensional berbasis tabel atau daftar teks. Pengguna dapat langsung mengetahui posisi perangkat melalui representasi visual pada denah digital gedung tanpa harus mencari perangkat secara manual berdasarkan nama atau kategori tertentu. Pendekatan ini dinilai lebih efektif pada lingkungan gedung dengan jumlah perangkat dan ruangan yang banyak[14].

Penggunaan *Jetpack Compose* memberikan fleksibilitas dalam pengembangan antarmuka interaktif. Posisi setiap node perangkat ditentukan berdasarkan koordinat yang tersimpan pada *Firebase Realtime Database* dan dapat diperbarui melalui mekanisme *drag-and-drop*, sehingga tata letak perangkat pada denah digital dapat disesuaikan berdasarkan kebutuhan pengguna.[10].

Dibandingkan penelitian sebelumnya yang menggunakan aplikasi pihak ketiga seperti Blynk[15] atau *dashboard* monitoring sederhana, sistem yang dikembangkan memiliki fleksibilitas antarmuka yang lebih tinggi karena dikembangkan menggunakan *Android native*. Pendekatan ini memungkinkan pengembangan visualisasi spasial perangkat secara lebih dinamis dan interaktif.

Namun demikian, sistem masih memiliki beberapa keterbatasan. Performa sistem sangat bergantung pada kestabilan koneksi internet karena seluruh proses sinkronisasi data dilakukan melalui layanan *cloud Firebase*. Ketika koneksi internet mengalami gangguan, waktu respon sistem dapat meningkat sehingga mempengaruhi proses monitoring dan kendali perangkat secara *real-time*.

Di samping berbagai keunggulan yang ditawarkan, sistem ini masih memiliki beberapa keterbatasan teknis dalam proses implementasinya di lapangan. Proses penempatan posisi perangkat pada denah digital saat ini masih dilakukan secara manual menggunakan penentuan titik koordinat statis, sehingga jika terjadi perubahan tata letak ruangan di PT Solo Abadi Indonesia, sistem memerlukan proses penyesuaian ulang koordinat *node* perangkat secara manual. Selain itu, fungsionalitas sistem juga belum mendukung fitur otomatisasi berbasis sensor lingkungan maupun modul analisis konsumsi energi listrik secara otomatis untuk mengoptimalkan efisiensi manajemen fasilitas gedung secara menyeluruh.

Berdasarkan hasil penelitian yang diperoleh, pendekatan *Interactive Floorplan* memiliki potensi besar untuk diterapkan pada sistem monitoring fasilitas gedung dengan skala yang lebih luas maupun multi-gedung. Keunggulan utamanya terletak pada kemampuan sistem dalam meningkatkan representasi visual perangkat secara spasial, sehingga dapat mempermudah interaksi pengguna dalam proses monitoring dan kendali perangkat listrik berbasis IoT. Jika diimplementasikan pada kompleks bangunan yang lebih masif, metode visualisasi ini diproyeksikan dapat mendukung proses monitoring dan identifikasi lokasi perangkat secara lebih efisien.

5. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem monitoring dan kendali perangkat listrik gedung berbasis *Interactive Floorplan* menggunakan mikrokontroler ESP32 dan *Firebase Realtime Database* pada PT Solo Abadi Indonesia. Sistem mampu menyinkronkan data secara *real-time* antara aplikasi Android dan perangkat IoT dengan rata-rata waktu respons sebesar 724 ms sehingga mendukung proses monitoring dan pengendalian perangkat listrik secara terintegrasi. Implementasi *Interactive Floorplan* memberikan kemudahan dalam mengidentifikasi lokasi perangkat berdasarkan posisi pada denah digital dibandingkan penyajian dalam bentuk daftar teks. Meskipun kinerja sistem dipengaruhi oleh stabilitas koneksi internet, penelitian ini menunjukkan bahwa pendekatan tersebut dapat diterapkan sebagai solusi monitoring dan pengendalian perangkat listrik berbasis Internet of Things. Kontribusi utama penelitian ini tidak terletak pada inovasi perangkat keras, karena masih menggunakan kombinasi ESP32 dan modul relay yang telah umum digunakan, melainkan pada pengembangan arsitektur perangkat lunak yang mengintegrasikan aplikasi Android berbasis *Jetpack Compose*, *Firebase Realtime Database*, dan ESP32, serta implementasi *Interactive Floorplan* sebagai antarmuka visual untuk mendukung monitoring dan pengendalian perangkat listrik secara lebih intuitif.

DAFTAR PUSTAKA

- [1] P. A. Auliasari, E. Astuti, S. Jamilatun, F. R. Rhomadoni, B. S. Wardhana, and A. V. Febriani, "Transformasi Manajemen Energi di Gedung Perkantoran Modern melalui Teknologi Cerdas," Jl. Ahmad Yani, Banguntapan, Bantul, D.I. Yogyakarta, 55191, Indonesia., 2025.
- [2] E. Masyudi, P. Informatika, F. Teknologi Informasi, and U. K. Abdul Wahab Hasbullah, "Rancang Bangun Otomatisasi Lampu Ruangan Berbasis *Internet of Things* Dengan Kendali Web Server Lokal Sujono, Erdi Masyudi RANCANG BANGUN OTOMATISASI LAMPU RUANGAN BERBASIS *INTERNET OF THINGS* DENGAN KENDALI WEB SERVER LOKAL," *J. ELEKTRO Inform. SWADHARMA*, p. 8, 2026.
- [3] I. N. W. R. Samkayana and Bagus Gede Krishna Yudistira, "IMPLEMENTASI *FIREBASE REAL-TIME DATABASE* PADA SMART CAT FEEDER BERBASIS

- MIKROKONTROLER ESP32,” *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3, Jul. 2025, doi: 10.23960/jitet.v13i3.7198.
- [4] A. F. Isnanto, A. Surriani, S. Lestari, and U. Y. Oktiawati, “Prototype of Smart Home and Monitoring Application Based On *Internet of Things (IoT)* Using Android,” *J. List. Instrumentasi dan Elektron. Terap.*, vol. 1, no. 1, pp. 6–9, 2021, doi: 10.22146/juliet.v1i1.53939.
- [5] R. Siswanto, “Pemanfaatan Received Signal Strength Indicator (RSSI) Access Point untuk Menentukan Posisi Objek Menggunakan Metode Triangulation,” *Dewantara J. Technol.*, vol. 1, no. 1, pp. 30–33, 2020.
- [6] Y. S. Rahayu, Y. Saputra, and D. Irawan, “IMPLEMENTASI METODE *WATERFALL* PADA PENGEMBANGAN SISTEM INFORMASI MOBILE E-DISARPUS,” *J. Sist. Inf.*, vol. 6, no. 2, pp. 523–534, 2024.
- [7] M. Sovian, S. Ibrahim, A. Faisol, and R. Promaswara, “RANCANG BANGUN APLIKASI CONTROLING DAN MONITORING RUANGAN ANDROID,” *J. Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 979–983, 2023.
- [8] A. Satria Restu Ageng, A. Mei Galuh Ageng, R. Kurniawan, and P. Pramono, “Alat Pengontrol AC Otomatis Menggunakan Relay Dengan Koneksi Blynk Berbasis Esp32,” *Pros. Semin. Nas. Teknol. Inf. dan Bisnis*, pp. 609–614, 2025, doi: 10.47701/kppexy33.
- [9] C. Diantoni, O. Komarudin, and A. Rizal, “ARSITEKTUR MVVM DAN FRAMEWORK *JETPACK COMPOSE* PADA PENGEMBANGAN APLIKASI ANDROID (STUDI KASUS : APLIKASI SUKACOLAB),” *J. Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3216–3224, 2024.
- [10] M. Arrizal and A. Rizal, “APLIKASI BERBAGI MAKANAN BERBASIS ANDROID CLEAN ARCHITECTURE DAN *JETPACK COMPOSE*,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024.
- [11] A. A. Syahrifah, E. Suwarno, A. Rahayu, A. A. A. Muzammil, A. V Saktiya, and Hartini, “IMPLEMENTASI *SMART BUILDING* SYSTEM PADA PENGENDALIAN UTILITAS PENCAHAYAAN DAN SUHU RUANG BERBASIS *IOT* Aqilah,” *J. Ilm. Pendidik. Dasar*, vol. 11, 2026.
- [12] M. Maulida, F. Zahro, R. Hakim, and M. S. Akbar, “PENGUJIAN *BLACK BOX TESTING* PADA SISTEM WEBSITE,” *J. MEDIA Akad.*, vol. 3, no. 5, 2025.
- [13] H. Kurnia, “IMPLEMENTASI *IOT* PADA SISTEM MONITORING SUHU DAN KELEMBABAN MENGGUNAKAN ESP32, FIREBASE DAN KODULAR,” 2025.
- [14] I. Kurniawan, S. Andriyanto, and I. Dwisaputra, “Aplikasi Smarthome Berbasis *IoT* Menggunakan Android Studio,” *J. Inov. Teknol. Terap.*, vol. 03, no. 2, 2025.
- [15] I. Menggunakan and E. S. P. Dan, “RANCANG BANGUN ALAT UKUR KADAR GULA DALAM DARAH NON INVASIVE MENGGUNAKAN ESP32 DAN BLYNK,” *J. Teknol. dan Sist. Inf. Univrab*, vol. 9, no. 2, pp. 361–371, 2024.