Rancang Bangun Smart Home Berbasis ESP32 dan Firebase Realtime Database dengan Kendali Ganda Smartphone-Pushbutton

Sufianti Munirman[1], Muhira Dzar Faraby[2], Anugerah Yoga Finandi[3]

[1]Program Studi D4 Teknik Listrik, Teknik Elektro, Politeknik Negeri Ujung Pandang, Makassar, Indonesia

[2,3]Program Studi D4 Teknik Listrik, Teknik Elektro, Politeknik Negeri Ujung Pandang, Makassar, Indonesia

Correspondent Author: sufiantimunirman@poliupg.ac.id

***Abstract* — The Internet of Things (IoT) enables smart-home systems in which electrical devices can be controlled through networked interfaces. This study designs and evaluates a four-load control prototype based on an ESP32, Firebase Realtime Database, an MIT App Inventor application, a four-channel relay module, and four local pushbuttons. The main problem addressed is maintaining state consistency when the same loads are operated through two different interfaces, namely a smartphone and physical buttons. The research method includes requirement analysis, IoT architecture design, hardware assembly, interface integration, and black-box functional testing. The ESP32 reads L1-L4 states from Firebase to drive an active-low relay module, while changes initiated by the pushbuttons are written back to Firebase so that local actuation and cloud states remain synchronized. Eight ON/OFF scenarios for four loads produced the expected database, relay, and lamp conditions, resulting in 100% coverage of the predefined functional scenarios. These results demonstrate the basic functionality of the dual-control mechanism; however, they do not represent operational reliability because repeated trials, end-to-end latency, reconnection behavior, energy consumption, and security were not evaluated.**

***Keyword* — ESP32, Firebase Realtime Database, Internet of Things, load control, smart home**

Abstrak — Perkembangan Internet of Things (IoT) mendorong penerapan sistem rumah pintar yang memungkinkan perangkat listrik dikendalikan melalui jaringan. Penelitian ini merancang dan mengevaluasi purwarupa kendali empat beban listrik berbasis ESP32 dengan Firebase Realtime Database, aplikasi MIT App Inventor, modul relay empat kanal, dan empat pushbutton lokal. Permasalahan yang dikaji adalah bagaimana mempertahankan konsistensi status ketika beban dikendalikan melalui dua antarmuka yang berbeda, yaitu smartphone dan tombol fisik. Metode penelitian mencakup analisis kebutuhan, perancangan arsitektur IoT, perakitan perangkat keras, integrasi antarmuka, serta pengujian fungsional dengan pendekatan black-box. ESP32 membaca status L1-L4 dari Firebase untuk mengendalikan relay active-low, sedangkan perubahan yang berasal dari pushbutton dituliskan kembali ke Firebase agar status lokal dan cloud tetap sinkron. Pengujian delapan skenario ON/OFF pada empat beban menghasilkan kondisi basis data, relay, dan lampu yang sesuai pada seluruh skenario, sehingga cakupan skenario fungsional mencapai 100%. Hasil tersebut membuktikan fungsi dasar kendali ganda, tetapi belum merepresentasikan reliabilitas operasional karena pengujian berulang, latensi end-to-end, pemulihan koneksi, konsumsi energi, dan keamanan belum dievaluasi.

Kata kunci — ESP32, Firebase Realtime Database, Internet of Things, kendali beban, rumah pintar

I. PENDAHULUAN

Rumah pintar (smart home) merupakan salah satu penerapan Internet of Things (IoT) yang menggabungkan perangkat kendali, aktuator, jaringan komunikasi, dan antarmuka pengguna agar perangkat rumah tangga dapat dipantau atau dikendalikan secara lebih fleksibel. Smartphone banyak digunakan sebagai antarmuka karena tersedia luas dan dapat terhubung melalui jaringan lokal maupun Internet. Khan et al. mengembangkan otomasi rumah berbasis Android yang memadukan kendali perangkat dan pemantauan energi [1], sedangkan Elkholy et al. menunjukkan bahwa kendali lokal dan jarak jauh dapat diintegrasikan dengan fungsi manajemen energi pada sistem rumah pintar [2]. Kedua penelitian tersebut memperlihatkan bahwa keberhasilan sistem tidak hanya ditentukan oleh aktuator, tetapi juga oleh arsitektur komunikasi dan rancangan antarmuka pengguna.

Pada sisi perangkat tertanam, mikrokontroler dengan konektivitas Wi-Fi banyak digunakan karena mampu menggabungkan fungsi komunikasi dan masukan-keluaran digital dalam satu pengendali. Niranjana et al. menggunakan ESP32 dan relay untuk pengendalian peralatan melalui platform IoT [3], sedangkan Sridhar et al. menggunakan ESP8266 NodeMCU, Arduino IoT Cloud, relay, serta tombol manual pada sistem otomasi rumah [4]. Mustofa et al. mengembangkan SECHA yang menggunakan ESP32, Firebase, MIT App Inventor, sensor, serta komunikasi Wi-Fi/GSM untuk membangun sistem otomasi rumah berbiaya rendah [5]. Orfanos et al. menunjukkan bahwa pemilihan teknologi jaringan untuk smart home perlu mempertimbangkan jangkauan, laju data, kualitas dan keamanan komunikasi, konsumsi energi, biaya, serta kompatibilitas antarteknologi [6].

Efisiensi energi sering disebut sebagai manfaat smart home, tetapi kemampuan menyalakan dan mematikan beban dari jarak jauh tidak dengan sendirinya membuktikan terjadinya penghematan energi. Fakhar et al. menunjukkan bahwa evaluasi konservasi energi memerlukan atribut, indikator, dan metrik yang terukur [7]. Umair et al. mengevaluasi manajemen energi pada lingkungan smart

home melalui implementasi dan pengukuran konsumsi energi [8], sedangkan Raza et al. menempatkan peramalan beban dan penjadwalan penggunaan energi sebagai komponen penting dalam Home Energy Management System [9]. Atas dasar itu, penelitian ini tidak menggunakan klaim penghematan energi sebagai hasil utama karena purwarupa belum dilengkapi sensor daya atau kWh meter.

Konektivitas cloud juga memperluas permukaan risiko keamanan. Vardakis et al. mengidentifikasi autentikasi, kontrol akses, kerahasiaan data, dan perlindungan terhadap akses tidak sah sebagai isu penting pada ekosistem smart home berbasis IoT [10]. Dengan demikian, keberhasilan fungsi ON/OFF perlu dibedakan dari reliabilitas, keamanan, dan kelayakan produk untuk penggunaan jangka panjang.

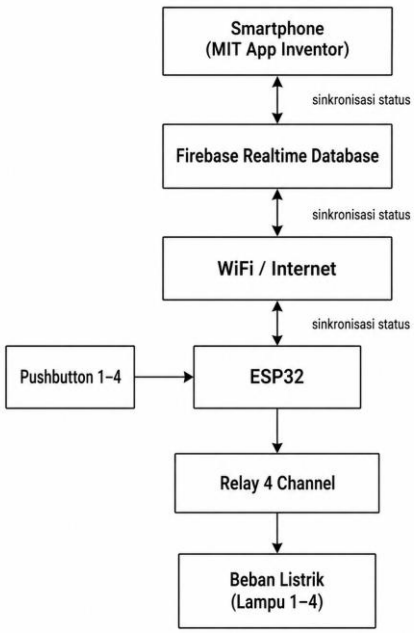
Perkembangan penelitian mutakhir memperlihatkan variasi pendekatan kendali dan komunikasi. Anuradha et al. memanfaatkan kendali berbasis aplikasi, suara, dan gestur untuk meningkatkan aksesibilitas [11]. Litayem mengusulkan kendali perangkat rumah berbasis ESP32-S3 yang menekankan biaya rendah, konektivitas, dan skalabilitas [12]. Arigela et al. menggunakan beberapa node ESP32, MQTT, Node-RED, dan Raspberry Pi untuk kendali serta pemantauan jarak jauh [13]. Pada tingkat ekosistem, Mota et al. menunjukkan bahwa protokol Matter dapat digunakan untuk meningkatkan interoperabilitas perangkat smart home dengan dukungan solusi Espressif [14]. Di sisi keamanan perangkat, Wijesundara et al. menekankan verifikasi autentisitas dan integritas firmware serta mekanisme pembaruan firmware yang aman pada perangkat IoT rumah [15].

Berdasarkan penelitian terdahulu, kombinasi kendali jarak jauh dan kendali lokal telah diterapkan dengan berbagai arsitektur [2], [4], [5]. Oleh karena itu, penelitian ini tidak mengklaim mekanisme kendali ganda sebagai konsep baru. Celah yang dikaji diarahkan pada implementasi dan evaluasi sinkronisasi status pada arsitektur yang lebih sederhana, yaitu empat kanal beban berbasis ESP32 dengan Firebase Realtime Database sebagai keadaan bersama (shared state) antara aplikasi smartphone dan pushbutton lokal. Perubahan dari aplikasi disimpan pada Firebase dan dibaca ESP32 untuk mengubah kondisi relay, sedangkan perubahan dari pushbutton ditulis kembali ke Firebase sehingga kondisi aktuator dan status pada antarmuka pengguna dapat dipertahankan konsisten. Penelitian ini bertujuan merancang arsitektur tersebut, mengimplementasikan integrasi ESP32-Firebase-relay-pushbutton, menguji fungsi ON/OFF pada empat beban, serta menganalisis posisi dan keterbatasannya dibandingkan penelitian sejenis.

II. METODE PENELITIAN

A. Arsitektur Sistem

Arsitektur pada Gambar 1 terdiri atas dua jalur kendali yang bertemu pada ESP32. Jalur kendali jarak jauh dimulai dari aplikasi MIT App Inventor pada smartphone, diteruskan melalui Firebase Realtime Database dan koneksi Wi-Fi/Internet menuju ESP32. Jalur kendali lokal berasal dari empat pushbutton yang terhubung langsung ke pin masukan ESP32. ESP32 mengubah status keluaran digital untuk mengendalikan modul relay empat kanal, sedangkan setiap kanal relay memutus atau menghubungkan suplai menuju satu beban lampu. Arsitektur ini menempatkan Firebase sebagai keadaan bersama yang dapat dibaca oleh aplikasi dan ESP32, sehingga perubahan status dapat disinkronkan pada kedua sisi.



Gambar 1. Arsitektur sistem Smart Home IoT.

B. Tahapan Penelitian

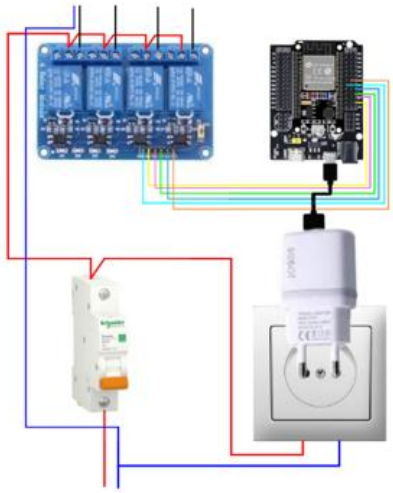
Tahapan penelitian ditunjukkan pada Gambar 2. Proses dimulai dari identifikasi masalah dan studi literatur untuk merumuskan kebutuhan kendali empat beban. Tahap berikutnya adalah analisis kebutuhan sistem, perancangan perangkat keras dan arsitektur IoT, serta perancangan antarmuka kendali. Setelah rancangan tersedia, komponen dirakit dan diintegrasikan, kemudian dilakukan pengujian fungsional. Hasil pengujian dievaluasi; jika ditemukan ketidaksesuaian, proses kembali ke tahap perancangan untuk melakukan revisi. Tahap terakhir adalah dokumentasi hasil. Alur iteratif ini digunakan agar perubahan pada perangkat keras, logika relay, atau komunikasi data dapat dikoreksi sebelum sistem dinyatakan memenuhi fungsi dasar yang ditetapkan.



Gambar 2. Tahapan penelitian.

C. Perangkat Keras dan Logika Digital

ESP32 berfungsi sebagai pengendali utama. Empat pin keluaran digunakan untuk memberi sinyal ke modul relay, sedangkan empat pin masukan menerima kondisi pushbutton. Relay yang digunakan memiliki karakteristik active-low, sehingga level LOW pada masukan relay mengaktifkan kanal, sedangkan level HIGH menonaktifkannya. Oleh karena itu, status logis ON/OFF tidak dapat disamakan langsung dengan level HIGH/LOW. Pemetaan status logis, level GPIO, kondisi relay, dan kondisi beban ditunjukkan pada Tabel I. Pushbutton menggunakan konfigurasi pull-up internal; masukan terbaca HIGH ketika tombol tidak ditekan dan berubah menjadi LOW ketika ditekan. Pada sisi daya, kontak relay berfungsi sebagai pemisah fungsional antara rangkaian kendali bertegangan rendah dan jalur beban AC. Diagram pengawatan satu kanal ditunjukkan pada Gambar 3 dan prinsip yang sama diterapkan pada kanal lainnya. MCB dipasang pada jalur daya sebagai proteksi arus lebih, tetapi penggunaan purwarupa tetap harus memperhatikan rating relay, isolasi, penutup rangkaian, dan praktik instalasi listrik yang aman.



Gambar 3. Rangkaian perangkat keras sistem Smart Home IoT.

TABEL I

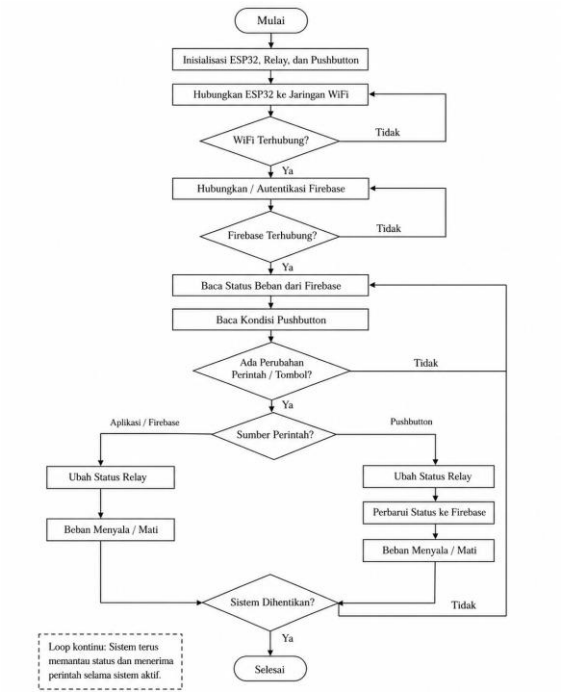
PEMETAAN STATUS LOGIS, GPIO, RELAY, DAN BEBAN

Status logis	GPIO relay	Relay	Beban
ON (1)	LOW	Aktif	Menyala
OFF (0)	HIGH	Nonaktif	Mati

D. Mekanisme Kendali dan Sinkronisasi Data

Mekanisme kendali menggunakan dua sumber masukan. Pada jalur aplikasi, pengguna memilih perintah ON atau OFF untuk beban L1-L4; status tersebut disimpan pada Firebase dan dibaca ESP32 untuk menentukan kondisi relay. Pada jalur lokal, penekanan pushbutton mengubah status beban yang bersesuaian. ESP32 kemudian menulis status terbaru ke Firebase sehingga perubahan lokal dapat tercermin kembali pada antarmuka smartphone. Mekanisme ini membedakan purwarupa dari sistem yang hanya menyediakan tombol fisik sebagai bypass tanpa pembaruan status jaringan.

Flowchart pada Gambar 4 memperlihatkan urutan operasi. Setelah inisialisasi ESP32, relay, dan pushbutton, sistem mencoba terhubung ke Wi-Fi lalu melakukan autentikasi ke Firebase. Jika koneksi belum tersedia, sistem mengulangi tahap koneksi. Setelah terhubung, ESP32 membaca status beban dari Firebase dan kondisi pushbutton secara berulang. Jika tidak terjadi perubahan, proses kembali ke pembacaan status. Jika terdapat perubahan, sumber perintah diidentifikasi. Perintah aplikasi mengubah relay sesuai status Firebase, sedangkan perintah pushbutton mengubah relay sekaligus memperbarui Firebase. Sistem terus beroperasi dalam loop hingga perangkat dihentikan.



Gambar 4. Flowchart sistem kendali Smart Home IoT.

Dengan struktur tersebut, kendali ganda membentuk dua arah pembaruan status. Perintah smartphone menghasilkan perubahan nilai pada Firebase yang kemudian dibaca ESP32, sedangkan perintah lokal menghasilkan perubahan keluaran dan penulisan balik ke Firebase. Dengan demikian, Firebase berfungsi sebagai keadaan bersama antara kendali jaringan dan kondisi lokal selama komunikasi berlangsung normal.

E. Prosedur Pengujian Fungsional

Pengujian dilakukan dengan pendekatan black-box terhadap fungsi ON/OFF empat beban. Masing-masing beban diberi dua kondisi uji sehingga terdapat delapan skenario, yaitu L1 ON, L1 OFF, L2 ON, L2 OFF, L3 ON, L3 OFF, L4 ON, dan L4 OFF. Pada setiap skenario diamati status basis data, kondisi relay, dan kondisi lampu. Skenario dinyatakan sesuai apabila status 1 menghasilkan relay aktif dan lampu

menyala, sedangkan status 0 menghasilkan relay nonaktif dan lampu mati. Data proyek tidak mencatat jumlah pengulangan untuk setiap skenario, distribusi waktu respons, kekuatan sinyal, atau gangguan jaringan. Karena itu, persentase keberhasilan yang dihitung pada penelitian ini hanya menyatakan cakupan skenario fungsional yang terdokumentasi dan tidak diperlakukan sebagai ukuran reliabilitas statistik.

III. HASIL DAN PEMBAHASAN

A. Implementasi Purwarupa dan Antarmuka

Hasil implementasi menunjukkan bahwa ESP32, modul relay empat kanal, instalasi beban, aplikasi smartphone, dan tombol lokal dapat diintegrasikan dalam satu purwarupa. Bentuk fisik purwarupa ditunjukkan pada Gambar 5. Empat jalur beban disusun sebagai empat kanal yang dapat dikendalikan secara individual. Pada sisi antarmuka, MIT App Inventor menyediakan pasangan tombol ON dan OFF untuk setiap grup beban seperti pada Gambar 6. Pemisahan tombol per kanal memudahkan verifikasi karena setiap aksi pengguna dapat dikaitkan langsung dengan satu status L1-L4 pada Firebase.



Gambar 5. Purwarupa sistem kendali empat beban.



Gambar 6. Antarmuka MIT App Inventor untuk empat beban.

Antarmuka pada Gambar 6 merealisasikan pola kendali berbasis aplikasi yang umum digunakan pada home automation [1], [3], [5]. Perbedaannya, status beban pada penelitian ini digunakan sebagai keadaan bersama antara aplikasi dan ESP32, sehingga tombol lokal tetap dapat memperbarui status jaringan. Pendekatan ini relevan dengan gagasan kendali multi-antarmuka pada penelitian aksesibilitas [11] dan kendali jarak jauh berbasis ESP32-S3 [12], meskipun kompleksitas fungsi purwarupa ini lebih rendah.

B. Hasil Pengujian Fungsional

TABEL II

HASIL PENGUJIAN FUNGSIONAL EMPAT BEBAN

Skenario	Status basis data	Relay	Lampu	Hasil
L1 ON	1	ON	Menyala	Sesuai
L1 OFF	0	OFF	Mati	Sesuai
L2 ON	1	ON	Menyala	Sesuai
L2 OFF	0	OFF	Mati	Sesuai
L3 ON	1	ON	Menyala	Sesuai
L3 OFF	0	OFF	Mati	Sesuai
L4 ON	1	ON	Menyala	Sesuai
L4 OFF	0	OFF	Mati	Sesuai

Berdasarkan delapan skenario yang didefinisikan, delapan skenario dinyatakan sesuai sehingga cakupan pemenuhan skenario fungsional adalah 8/8 atau 100%. Nilai ini menunjukkan konsistensi fungsi dasar untuk kombinasi ON/OFF yang diuji. Namun, nilai 100% tersebut tidak dapat disebut sebagai success rate operasional karena setiap skenario tidak dilaporkan diuji berulang dalam jumlah tertentu. Untuk memperoleh ukuran reliabilitas, diperlukan pengulangan yang terdefinisi pada berbagai kondisi jaringan disertai pencatatan perintah gagal, koneksi terputus, dan waktu respons. Penelitian terdahulu juga mengevaluasi sistem menggunakan indikator yang lebih luas, misalnya konsumsi energi dan biaya operasi [2], serta pengujian konektivitas dan kemampuan monitoring pada arsitektur multi-node [13].

C. Analisis Konsistensi Kendali Ganda

Karakteristik active-low menjadi bagian penting dalam konsistensi sistem. Nilai logis ON diterjemahkan menjadi level LOW pada GPIO relay, sedangkan nilai OFF diterjemahkan menjadi HIGH. Apabila pemetaan tersebut dibalik, aplikasi dapat menampilkan status ON ketika beban justru mati. Pemisahan antara status logis dan level listrik seperti pada Tabel I mencegah ambiguitas tersebut. Pada pushbutton, pembacaan HIGH ketika dilepas dan LOW ketika ditekan juga harus diproses sebagai kejadian perubahan, bukan sebagai status relay secara langsung. Dari sisi aliran data, kendali ganda membentuk dua arah pembaruan. Perintah smartphone menghasilkan perubahan nilai pada Firebase yang kemudian dibaca ESP32, sedangkan perintah lokal menghasilkan perubahan keluaran dan penulisan balik ke Firebase. Konsep ini mengurangi risiko

perbedaan antara kondisi fisik dan status aplikasi selama komunikasi berjalan normal. Penggunaan basis data cloud sebagai mediator berbeda dengan pendekatan MQTT dan Node-RED yang dilaporkan Arigela et al. [13]; kedua pendekatan sama-sama memerlukan mekanisme sinkronisasi, tetapi arsitektur broker MQTT dan basis data realtime memiliki karakteristik berbeda dalam pengelolaan pesan dan status.

D. Perbandingan Penelitian Sejenis dan Celah Penelitian

Perbandingan penelitian sejenis digunakan untuk menempatkan kontribusi purwarupa secara proporsional, bukan untuk menyatakan sistem ini lebih unggul. Beberapa penelitian yang memiliki kedekatan dari sisi perangkat, antarmuka, atau mekanisme kendali dibandingkan secara ringkas pada uraian berikut.

Khan et al. [1] menggunakan Android dan IoT untuk otomasi rumah sekaligus pemantauan energi, sedangkan Elkholy et al. [2] membangun sistem yang lebih luas dengan Raspberry Pi, sensor, Cayenne, Alexa, kendali lokal/jarak jauh, dan optimasi energi. Dibandingkan kedua penelitian tersebut, purwarupa ini memiliki ruang lingkup yang lebih sempit karena tidak melakukan pengukuran maupun optimasi energi; fokusnya adalah konsistensi kendali empat kanal melalui smartphone dan pushbutton.

Sridhar et al. [4] telah menyediakan kendali cloud/voice dan tombol manual menggunakan ESP8266 NodeMCU serta Arduino IoT Cloud. Mustofa et al. [5] juga menggunakan kombinasi teknologi yang sangat dekat dengan penelitian ini, yaitu ESP32, Firebase, MIT App Inventor, Wi-Fi/GSM, dan dukungan kendali manual dalam sistem multi-sensor. Karena itu, penelitian ini tidak mengklaim kendali lokal-online sebagai konsep baru. Perbedaannya diarahkan pada evaluasi sinkronisasi status empat kanal dengan Firebase sebagai keadaan bersama antara aplikasi dan pushbutton.

Pada pendekatan lain, Anuradha et al. [11] mengembangkan kendali multimodal melalui aplikasi, suara, dan gestur untuk aksesibilitas, sedangkan Arigela et al. [13] menggunakan beberapa node ESP32, MQTT, Node-RED, dan Raspberry Pi untuk kendali serta pemantauan jarak jauh. Perbandingan ini menunjukkan bahwa kontribusi penelitian bersifat implementatif dan sengaja dibatasi agar hubungan antara input fisik, keluaran active-low, dan status cloud dapat dianalisis secara eksplisit.

Dibandingkan penelitian tersebut, kontribusi penelitian ini bersifat implementatif: empat kanal beban dikendalikan oleh dua antarmuka yang berbagi status yang sama pada Firebase. Fokus ini lebih sempit daripada sistem multi-sensor, optimasi energi, atau kendali multimodal, tetapi memungkinkan hubungan antara input fisik, keluaran active-low, dan status cloud dianalisis secara eksplisit. Litayem [12] menunjukkan relevansi ESP32-S3 untuk kendali rumah berbiaya rendah dan skalabel, sedangkan Mota et al. [14] menunjukkan penggunaan Matter bersama solusi Espressif untuk meningkatkan interoperabilitas perangkat smart home. Dengan demikian, arsitektur pada penelitian ini diposisikan sebagai purwarupa dasar yang masih perlu diperluas sebelum mencapai tingkat interoperabilitas dan skalabilitas yang lebih tinggi.

E. Keterbatasan dan Arah Pengembangan

Keterbatasan pertama adalah evaluasi kinerja. Dokumentasi yang tersedia hanya mendukung pengujian delapan skenario ON/OFF dan belum menyediakan data latensi end-to-end, jitter, tingkat keberhasilan dari pengulangan perintah, atau waktu pemulihan setelah Wi-Fi terputus. Pengujian lanjutan perlu dilakukan dengan sejumlah pengulangan yang terdefinisi, pencatatan waktu sejak perintah dikirim sampai perubahan beban teramati, serta variasi kualitas jaringan. Dengan data tersebut, istilah real-time atau responsif dapat dinilai menggunakan metrik, bukan berdasarkan pengamatan subjektif.

Keterbatasan kedua adalah tidak adanya pengukuran energi. Evaluasi efisiensi energi pada smart home sebaiknya didasarkan pada indikator dan metrik yang terukur [7]. Umair et al. menunjukkan contoh evaluasi melalui implementasi nyata dan pengukuran konsumsi energi [8], sedangkan Raza et al. menegaskan pentingnya peramalan beban dan penjadwalan dalam pengelolaan energi rumah [9]. Pengembangan sistem dapat menambahkan sensor tegangan-arus atau power meter untuk mengukur daya aktif dan energi sebelum serta sesudah strategi kendali diterapkan. Dengan demikian, penelitian lanjutan dapat menguji apakah kemudahan kendali benar-benar menghasilkan perubahan konsumsi energi.

Keterbatasan ketiga berkaitan dengan keamanan dan interoperabilitas. Vardakis et al. menekankan autentikasi, kontrol akses, dan perlindungan data pada smart home [10], sedangkan Wijesundara et al. menekankan autentisitas, integritas, serta pembaruan firmware yang aman pada perangkat IoT rumah [15]. Pada purwarupa, kredensial Firebase, aturan akses basis data, dan mekanisme pembaruan firmware belum dievaluasi secara khusus. Pengembangan berikutnya perlu menerapkan aturan akses basis data, proteksi kredensial, enkripsi komunikasi yang tepat, serta prosedur pembaruan firmware yang aman. Di sisi interoperabilitas, dukungan standar seperti Matter dapat dipertimbangkan agar perangkat dapat berintegrasi dengan ekosistem yang lebih luas [14].

Aspek keselamatan listrik juga menjadi batas penting. Keberhasilan mengendalikan relay tidak sama dengan sertifikasi keselamatan instalasi. Untuk implementasi pada jaringan PLN, rating kontak relay, ukuran penghantar, proteksi arus lebih, isolasi sisi tegangan rendah dan tegangan jala-jala, grounding, serta enclosure harus memenuhi praktik instalasi yang berlaku. Oleh karena itu, purwarupa pada penelitian ini diposisikan sebagai media pengujian akademik dan bukan produk akhir siap pakai.

IV. KESIMPULAN

Penelitian ini menghasilkan purwarupa smart home berbasis ESP32 yang mengintegrasikan Firebase Realtime Database, MIT App Inventor, modul relay empat kanal, dan empat pushbutton untuk mengendalikan empat beban listrik melalui dua antarmuka. Mekanisme kendali menggunakan status L1-L4 sebagai keadaan bersama: perintah aplikasi dibaca ESP32 untuk mengubah relay, sedangkan perubahan dari pushbutton dituliskan kembali ke Firebase agar kondisi lokal dan status cloud tetap konsisten. Delapan skenario ON/OFF yang terdokumentasi menghasilkan kondisi basis data, relay, dan lampu yang sesuai, sehingga seluruh skenario fungsional yang didefinisikan terpenuhi. Kontribusi penelitian terletak pada implementasi sinkronisasi kendali smartphone-pushbutton pada empat kanal, bukan pada klaim keunggulan terhadap sistem smart home yang lebih kompleks. Hasil ini belum membuktikan reliabilitas jangka panjang, respons real-time, penghematan energi, keamanan, maupun kelayakan keselamatan produk. Penelitian lanjutan perlu menambahkan pengujian berulang, pengukuran latensi dan pemulihan koneksi, pengukuran energi, pengujian konflik perintah, pengamanan akses cloud, serta evaluasi interoperabilitas untuk meningkatkan validitas dan kesiapan sistem.

DAFTAR ACUAN

[1] M. A. Khan, I. Ahmad, A. N. Nordin, A. E.-S. Ahmed, H. Mewada, Y. I. Daradkeh, S. Rasheed, E. T. Eldin, and M. Shafiq, "Smart Android Based Home Automation System Using Internet of Things (IoT)," Sustainability, vol. 14, no. 17, Art. no. 10717, Aug. 2022, doi: 10.3390/su141710717.

[2] M. H. Elkholy, T. Senjyu, M. E. Lotfy, A. Elgarhy, N. S. Ali, and T. S. Gaafar, "Design and Implementation of a Real-Time Smart Home Management System Considering Energy Saving," Sustainability, vol. 14, no. 21, Art. no. 13840, Oct. 2022, doi: 10.3390/su142113840.

[3] R. Niranjana, Arvind S, Vignesh M, and Vishaal S, "Effectual Home Automation using ESP32 NodeMCU," in 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), 2022, pp. 1-5, doi: 10.1109/ICACRS55517.2022.10028992.

[4] H. S. Sridhar, S. Somvanshi, M. Ramsinghani, and N. H. C., "Designing of Smart Home Automation Using IoT," in 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), 2022, pp. 1-6, doi: 10.1109/ICERECT56837.2022.10059802.

[5] A. A. Mustofa, Y. A. Dagnew, P. Gantela, and M. J. Idrisi, "SECHA: A Smart Energy-Efficient and Cost-Effective Home Automation System for Developing Countries," Journal of Computer Networks and Communications, vol. 2023, Art. no. 8571506, pp. 1-12, Mar. 2023, doi: 10.1155/2023/8571506.

[6] V. A. Orfanos, S. D. Kaminaris, P. Papageorgas, D. Piromalis, and D. Kandris, "A Comprehensive Review of IoT Networking Technologies for Smart Home Automation Applications," Journal of Sensor and Actuator Networks, vol. 12, no. 2, Art. no. 30, Apr. 2023, doi: 10.3390/jsan12020030.

[7] M. Z. Fakhar, E. Yalcin, and A. Bilge, "A survey of smart home energy conservation techniques," Expert Systems with Applications, vol. 213, Art. no. 118974, Mar. 2023, doi: 10.1016/j.eswa.2022.118974.

[8] M. Umair, M. A. Cheema, B. Afzal, and G. Shah, "Energy management of smart homes over fog-based IoT architecture," Sustainable Computing: Informatics and Systems, vol. 39, Art. no. 100898, Sep. 2023, doi: 10.1016/j.suscom.2023.100898.

[9] A. Raza, L. Jingzhao, Y. Ghadi, M. Adnan, and M. Ali, "Smart home energy management systems: Research challenges and survey," Alexandria Engineering Journal, vol. 92, pp. 117-170, Apr. 2024, doi: 10.1016/j.aej.2024.02.033.

[10] G. Vardakis, G. Hatzivasilis, E. Koutsaki, and N. Papadakis, "Review of Smart-Home Security Using the Internet of Things," Electronics, vol. 13, no. 16, Art. no. 3343, Aug. 2024, doi: 10.3390/electronics13163343.

[11] P. Anuradha, K. Vasanth, G. Renuka, and A. Rajeshwar Rao, "IoT based enabling home automation system for individuals with diverse disabilities," e-Prime - Advances in Electrical Engineering, Electronics and Energy, vol. 6, Art. no. 100366, Dec. 2023, doi: 10.1016/j.prime.2023.100366.

[12] N. Litayem, "Scalable Smart Home Management with ESP32-S3: A Low-Cost Solution for Accessible Home Automation," in 2024 International Conference on Computer and Applications (ICCA), Cairo, Egypt, 2024, pp. 1-7, doi: 10.1109/ICCA62237.2024.10927887.

[13] A. K. Arigela, C. Banapuram, and N. Venu, "Remote based Home Automation with MQTT: ESP32 Nodes and Node-RED on Raspberry Pi," in 2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 2024, pp. 313-318, doi: 10.1109/I-SMAC61858.2024.10714655.

[14] A. Mota, C. Serôdio, and A. Valente, "Matter Protocol Integration Using Espressif's Solutions to Achieve Smart Home Interoperability," Electronics, vol. 13, no. 11, Art. no. 2217, Jun. 2024, doi: 10.3390/electronics13112217.

[15] W. M. A. B. Wijesundara, J.-S. Lee, D. Tith, E. Aloupogianni, H. Suzuki, and T. Obi, "Security-enhanced firmware management scheme for smart home IoT devices using distributed ledger technologies," International Journal of Information Security, vol. 23, no. 3, pp. 1927-1937, Jun. 2024, doi: 10.1007/s10207-024-00827-x.