

Sistem Kendali Peralatan Elektronik Menggunakan Voice Recognition

Hisyam Arsyad Zakiyuddin^{1*}, Izzul Maali², Ravinolddy Firza Pratama³, Pramono⁴

¹Teknik Informatika/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

^{1*}230103226@mhs.udb.ac.id

²Teknik Informatika/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

²230103025@mhs.udb.ac.id

³Teknik Informatika/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

³230103234@mhs.udb.ac.id

⁴Teknik Informatika/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

⁴Pramono@udb.ac.id

Abstrak—Perkembangan teknologi otomasi rumah pintar (*smart home*) saat ini menjadi solusi penting untuk meningkatkan efisiensi aktivitas manusia dan membantu aksesibilitas, khususnya dalam meminimalkan kelalaian pengelolaan interaksi fisik perangkat domestik. Penelitian ini bertujuan untuk merancang dan mengimplementasikan prototipe sistem kendali lampu pada dua ruangan, yaitu ruang tamu dan kamar tidur, yang dapat dioperasikan melalui perintah suara, teks, maupun kontrol manual berbasis aplikasi web. Metodologi yang digunakan meliputi integrasi perangkat keras berupa mikrokontroler ESP32 dan modul relai sebagai aktuator lampu, serta perangkat lunak berupa aplikasi web yang memanfaatkan *Web Speech API* untuk konversi suara menjadi teks, layanan *Natural Language Processing* (NLP) Dialogflow ES untuk mengenali maksud (*intent*) dan ruangan target pengguna, protokol *Message Queuing Telemetry Transport* (MQTT) melalui *broker* HiveMQ Cloud sebagai jalur kendali perangkat, serta *Firebase Realtime Database* sebagai media sinkronisasi status secara waktu nyata. Validasi sistem dilakukan melalui pengujian eksperimental yang mencakup kesesuaian kontrol manual pada dasbor aplikasi serta analisis waktu tanggap (*response time*) kontrol berbasis suara dan teks. Hasil pengujian menunjukkan bahwa seluruh percobaan kontrol manual berjalan tanpa selisih status antara perintah dan kondisi fisik lampu, sedangkan kontrol berbasis suara dan teks memiliki rata-rata waktu tanggap total sebesar 2.201,33 milidetik, dengan kontribusi layanan NLP mencapai sekitar 98,1% dari total waktu tersebut. Sistem juga terbukti memiliki ketahanan terhadap gangguan koneksi jaringan, ditandai dengan kemampuan penyambungan ulang (*auto-reconnect*) otomatis pada protokol MQTT tanpa kehilangan perintah pengguna. Hasil penelitian menunjukkan bahwa integrasi ESP32, NLP *cloud*, protokol MQTT, dan basis data waktu nyata mampu menghasilkan sistem kendali rumah pintar yang fungsional, responsif, dan andal untuk diterapkan pada skala rumah tangga.

Kata kunci—ESP32, MQTT, Firebase, Dialogflow, *Smart Home*, *Voice Recognition*.

Abstract—The rapid development of smart home automation technology has become a crucial solution for improving human activity efficiency and enhancing accessibility, particularly in minimizing negligence in the physical management of domestic devices. This research aims to design and implement a prototype lighting control system for two rooms, namely the living room and the bedroom, which can be operated through voice commands, text commands, or manual control via a web application. The methodology involves the integration of hardware consisting of an ESP32 microcontroller and a relay module as the lamp actuator, along with software comprising a web application that utilizes the *Web Speech API* for speech-to-text conversion, the Dialogflow ES *Natural Language Processing* (NLP) service to recognize user *intent* and target room, the *Message Queuing Telemetry Transport* (MQTT) protocol via the HiveMQ Cloud *broker* as the device control channel, and *Firebase Realtime Database* as the real-time status synchronization medium. System validation was conducted through experimental testing covering the consistency of manual control on the application dashboard and the *response time* analysis of voice- and text-based control. The test results show that all manual control trials ran without any discrepancy between the issued command and the physical lamp condition, while voice- and text-based control recorded an average total response time of 2,201.33 milliseconds, with the NLP service contributing approximately 98.1% of that total time. The system also demonstrated resilience against network connectivity disruptions, evidenced by the automatic *reconnect* capability of the MQTT protocol without losing any user command. The results indicate that the integration of ESP32, *cloud*-based NLP, the MQTT protocol, and a real-time database can produce a functional, responsive, and reliable smart home control system suitable for household-scale implementation.

Keywords—ESP32, MQTT, Firebase, Dialogflow, *Smart Home*, *Voice Recognition*.

I. PENDAHULUAN

Perkembangan teknologi yang sangat pesat saat ini membawa kita menuju era modernisasi di mana hampir semua aspek kehidupan manusia sangat bergantung pada teknologi untuk mempermudah berbagai aktivitas sehari-hari. Pengoperasian peralatan elektronik domestik seperti menyalakan dan mematikan lampu, kipas angin, maupun setrika merupakan kegiatan umum yang rutin dilakukan, terutama oleh masyarakat yang tinggal di daerah perkotaan. Namun, pola interaksi fisik konvensional ini sering kali memicu kelalaian dalam pengelolaan energi listrik akibat kelupaan mematikan perangkat saat meninggalkan ruangan. Interaksi manusia dengan ekosistem rumah tangga kini mulai dipermudah melalui pemanfaatan bahasa alami atau *Natural Language Processing* (NLP) yang menjembatani komunikasi komunikasi digital secara lebih intuitif. Penggunaan perintah suara sebagai antarmuka kendali utama memberikan keuntungan multifold, salah satunya menghilangkan kebutuhan pelatihan khusus bagi pengguna awam. Oleh karena itu, integrasi teknologi kontrol berbasis suara menjadi solusi krusial dalam mentransformasikan rumah tinggal biasa menjadi ekosistem *smart home* yang efisien [1][2][3][4][5].

Urgensi pelaksanaan penelitian ini didasari oleh meningkatnya kebutuhan terhadap sistem aksesibilitas yang inklusif di lingkungan tempat tinggal modern. Bagi kelompok masyarakat lanjut usia (lansia) dan individu dengan disabilitas fisik, mobilitas untuk menjangkau sakelar manual di dinding menjadi hambatan nyata dalam beraktivitas mandiri. Selain faktor aksesibilitas, efisiensi pengelolaan konsumsi daya listrik di skala rumah tangga juga menjadi prioritas penting guna menekan pemborosan energi yang tidak perlu. Melalui penerapan kendali otomatis, risiko kerusakan alat elektronik akibat durasi operasional yang berlebihan dapat diminimalkan secara signifikan. Komputer atau mikrokontroler dituntut untuk mampu menerjemahkan bahasa alami manusia yang bersifat tidak terstruktur menjadi instruksi digital yang presisi. Dengan demikian, perancangan sistem ini mendesak untuk dilakukan demi menghadirkan lingkungan domestik yang lebih

aman, hemat energi, dan ramah terhadap seluruh kategori pengguna [3][4][5][6].

Sejumlah penelitian terdahulu telah berupaya mengembangkan sistem serupa, namun masih menyisakan beberapa celah dan kelemahan operasional. Penelitian terdahulu berhasil menerapkan *speech recognition* berbasis aplikasi seluler, tetapi interaksinya masih terbatas pada jarak jangkauan jaringan lokal (LAN) konvensional. Studi lain menggunakan sensor suara fisik untuk kendali lampu, namun perangkat tersebut sangat rentan terhadap *noise* lingkungan sehingga sering memicu salah eksekusi. Penggunaan modul perangkat keras khusus seperti *Voice Recognition Module V3* juga ditemukan memiliki keterbatasan kapasitas memori dalam menyimpan variasi kata kunci perintah baru. Selain itu, sistem kendali yang mengandalkan satu jalur komunikasi database sering kali mengalami *bottleneck* ketika memproses konversi teks dan perintah perangkat keras secara bersamaan. Kelemahan-kelemahan tersebut membatasi fleksibilitas dan keandalan sistem jika diterapkan pada struktur bangunan rumah yang kompleks atau wilayah dengan internet tidak stabil [1][4][5][6][7].

Untuk mengatasi keterbatasan tersebut, penelitian ini mengusulkan sebuah arsitektur *multi-tier* yang memisahkan jalur komunikasi data dari jalur kendali perangkat keras. Pendekatan eksperimental terapan ini mengintegrasikan mikrokontroler ESP32 dengan layanan *cloud* NLP Dialogflow ES untuk pengenalan niat (*intent*) pengguna yang lebih akurat. Aplikasi berbasis web dikembangkan dengan memanfaatkan *Web Speech API* agar konversi suara ke teks dapat diakses secara fleksibel dari perangkat apa pun tanpa bergantung pada aplikasi Android tertentu. Komunikasi nirkabelnya didukung oleh kombinasi paralel antara Firebase *Realtime Database* untuk sinkronisasi status dan protokol MQTT melalui broker HiveMQ Cloud. Melalui skema ini, keterbatasan jarak fisik dan kendala penempatan perangkat akibat sekat bangunan dapat diatasi dengan baik dibandingkan jaringan lokal biasa. Sistem ini juga dilengkapi dengan fungsi *auto-reconnect* otomatis pada firmware

untuk menjamin ketahanan terhadap gangguan jaringan internet [2][3][7][8].

Tujuan utama dari penelitian ini adalah merancang, mengimplementasikan, dan memvalidasi prototipe sistem kendali peralatan elektronik dua ruangan yang responsif serta andal. Kontribusi ilmiah dari penelitian ini terletak pada pemisahan arsitektur data secara paralel yang terbukti efisien dan tidak membebani prosesor lokal. Selain itu, pengujian empiris yang disajikan memberikan analisis mendalam mengenai karakteristik *cold start* pada layanan NLP cloud serta kontribusi latensinya terhadap total waktu tanggap sistem. Mekanisme mitigasi gangguan koneksi menggunakan parameter *Quality of Service* (QoS) 1 dan status *retain* pada pesan MQTT juga menjadi nilai tambah keandalan dalam penelitian ini. Hasil evaluasi ini diharapkan dapat menjadi panduan teoretis sekaligus referensi praktis dalam pengembangan sistem otomasi rumah pintar berskala rumah tangga. Melalui implementasi nyata ini, diharapkan tercipta standarisasi sistem kendali bahasa alami yang tangguh terhadap dinamika konektivitas internet rumahan [2][8][9].

II. METODOLOGI PENELITIAN

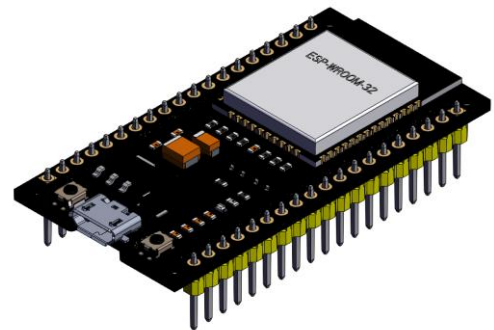
Penelitian ini menggunakan pendekatan eksperimental terapan yang mengintegrasikan perancangan perangkat keras (hardware) dan pengembangan perangkat lunak (software) berbasis Internet of Things (IoT). Proses penelitian dibagi menjadi empat tahapan utama: analisis kebutuhan sistem, perancangan arsitektur, implementasi sistem, serta evaluasi dan validasi model. Secara metodologis, pendekatan perancangan, pengembangan, dan pengujian secara bertahap terhadap prototipe sistem kontrol ini mengadopsi prinsip dasar metode *Research and Development* (R&D) yang memadukan kajian teoritis dengan pembuktian praktis secara empiris [2].

A. Analisis Kebutuhan Sistem

Tahap awal dilakukan untuk mengidentifikasi komponen yang diperlukan agar sistem dapat berjalan secara optimal

dengan tingkat respons yang tinggi. Kebutuhan tersebut dibagi menjadi dua kategori:

1. Perangkat Keras (Hardware) disini menggunakan mikrokontroler ESP32 (seperti pada gambar 2) sebagai pusat pemrosesan data yang telah dilengkapi dengan modul Wi-Fi internal, modul relay (seperti pada gambar 3) sebagai sakelar elektronik, serta peralatan elektronik rumah tangga (seperti lampu) sebagai objek kendali.



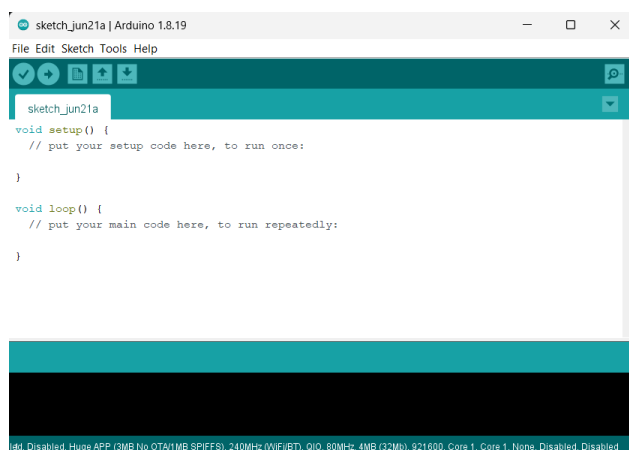
Gambar 1. ESP32



Gambar 2. Relay IoT

2. Perangkat Lunak (Software) Arduino IDE (Integrated Development Environment) adalah sebuah perangkat lunak Open Source yang digunakan untuk menulis kode. Dibuat dengan menggunakan bahasa pemrograman Java, Arduino IDE dapat dijalankan pada

berbagai platform seperti Windows, Mac, dan Linux. Arduino IDE digunakan untuk pemrograman firmware ESP32 sedangkan untuk pemrograman aplikasi berbasis web memanfaatkan fitur bawaan Speech-to-Text berbasis Natural Language Processing (NLP), serta platform Cloud/Database Real-time (seperti Firebase atau MQTT Broker) sebagai perantara data nirkabel [7].



Gambar 3. Arduino IDE

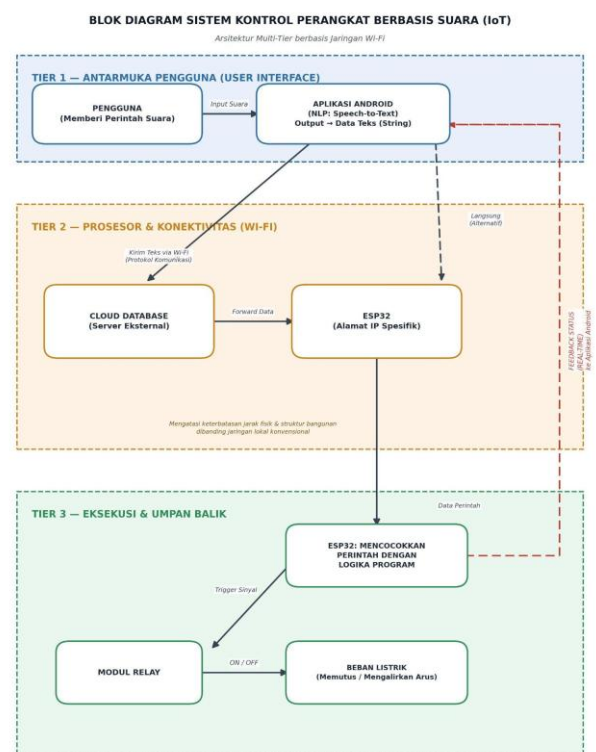
B. Blok Diagram Sistem

Sistem ini dirancang menggunakan arsitektur *multi-tier* berbasis jaringan Wi-Fi untuk memastikan transmisi data yang cepat dan stabil. Diagram alur interaksi data pada sistem ditunjukkan pada mekanika kerja berikut:

- Tier 1 (Antarmuka Pengguna)**
Pengguna memberikan perintah suara melalui aplikasi berbasis web. Aplikasi web memanfaatkan teknologi NLP untuk mengubah input suara menjadi data teks (*string*).
- Tier 2 (Prosesor & Konektivitas)**
Data teks dikirimkan secara nirkabel melalui jaringan Wi-Fi menuju *database cloud* atau langsung ke alamat IP ESP32 melalui protokol komunikasi yang ditentukan. Penerapan arsitektur nirkabel berbasis *cloud database* eksternal ini dinilai sangat efektif untuk mengatasi batasan jarak fisik serta kendala penempatan perangkat akibat struktur

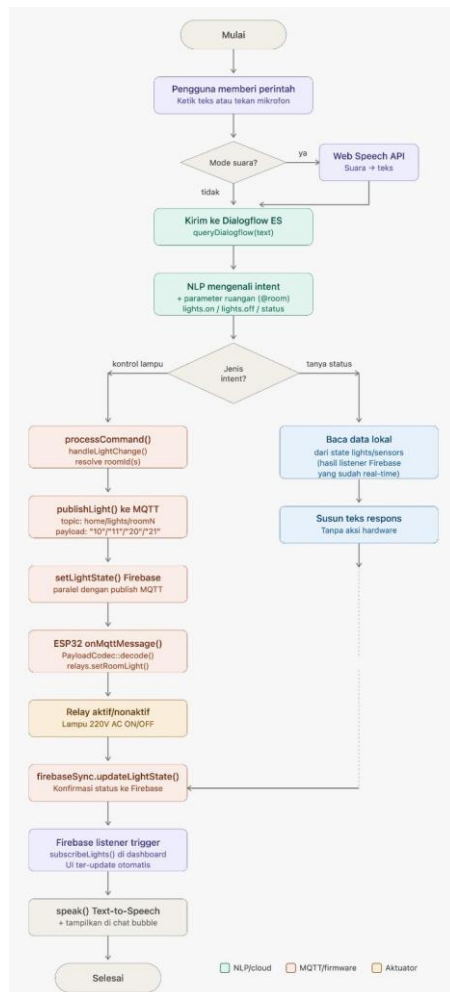
bangunan jika dibandingkan dengan penggunaan jaringan lokal konvensional [2].

- Tier 3 (Eksekusi & Umpan Balik)**
ESP32 menerima data perintah, mencocokkannya dengan logika program, dan memicu modul relay untuk memutus atau mengalirkan arus listrik. Bersamaan dengan itu, kemudian ESP32 mengirimkan status balik (*feedback*) ke aplikasi berbasis web secara *real-time*.



Gambar 4. Blok Diagram Sistem

C. Flowchart sistem

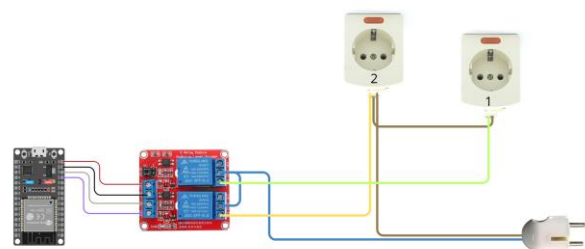


Gambar 5. Flowchart Sistem

Alur kerja sistem kendali perangkat elektronik *smart home* digambarkan pada Gambar 4. Dari Gambar dapat kita lihat bahwa Sistem kendali peralatan elektronik ini diawali ketika pengguna memberikan perintah melalui antarmuka aplikasi web, baik dengan cara mengetik teks maupun menekan tombol mikrofon untuk input suara. Apabila pengguna memilih mode suara, sistem akan memanfaatkan *Web Speech API* untuk mengonversi suara menjadi teks (*Speech-to-Text*). Namun, jika pengguna menggunakan mode teks sejak awal, input tersebut akan langsung diteruskan bersama hasil konversi suara menuju layanan *cloud* Dialogflow ES melalui fungsi *queryDialogflow(text)*. Di dalam pustaka Dialogflow, proses *Natural Language Processing* (NLP) dilakukan untuk mengenali niat pengguna (*intent*) sekaligus

mengekstrak parameter entitas ruangan (*@room*), seperti perintah menyalakan lampu (*lights.on*), mematikan lampu (*lights.off*), atau sekadar menanyakan kondisi terkini. Selanjutnya, sistem menjalankan dua proses secara paralel, yaitu mengirimkan data status baru ke database Firebase melalui fungsi *setLightState()* dan mempublikasikan pesan (*publish*) ke broker MQTT. Pesan MQTT tersebut dikirim dengan topik spesifik *home/lights/roomN* dan memuat *payload* berupa kode numerik seperti "10", "11", "20", atau "21". Di sisi perangkat keras, mikrokontroler ESP32 akan menangkap pesan melalui fungsi *onMqttMessage()*. Data tersebut kemudian didekode oleh fungsi *PayloadCodec::decode()* dan diteruskan ke fungsi *relays.setRoomLight()* untuk mengendalikan modul relay. Relay secara fisik akan memutus atau menyambungkan arus listrik AC 220V sehingga lampu berubah kondisi (menyala atau mati). Setelah aksi fisik berhasil dilakukan, mikrokontroler mengirimkan sinyal konfirmasi balik ke Firebase melalui fungsi *firebaseSync.updateLightState()* [8].

D. Schema Wiring IoT



Gambar 6. Skema Kabel IoT

Implementasi dilakukan dengan merakit komponen seperti pada gambar 5. Sistem pengendali dirancang dengan menggabungkan komponen yang dibutuhkan antara lain ESP32 yang dihubungkan dengan relay, dengan mengenali perintah suara dari aplikasi web yang berfungsi mengendalikan dan

memberikan informasi terkait monitoring setiap alat rumah tangga yang digunakan. Sinkronisasi antara aplikasi web dan ESP32 diatur menggunakan Wi-Fi yang sama (jaringan lokal) atau melalui jaringan internet berbasis *cloud* agar perangkat dapat dikendalikan dari jarak jauh tanpa batasan ruang sepanjang tersambung ke internet [6].

E. Evaluasi dan Validasi Model

Untuk menjamin keandalan sistem sebagai solusi praktis, ekonomis, dan memiliki aksesibilitas tinggi, dilakukan pengujian kinerja berdasarkan tiga parameter utama:

1. Pengujian Kontrol Manual

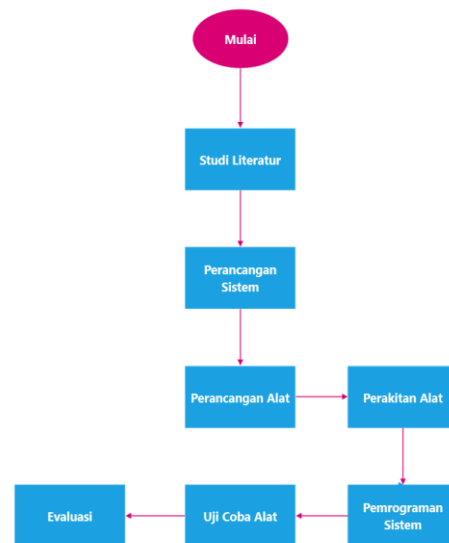
Mengukur keberhasilan sistem dalam mengeksekusi perintah lewat tombol dasbor dan memastikan sinkronisasi status beralih dengan tepat tanpa selisih kondisi.

2. Pengujian Kontrol Suara dan Teks

Mengukur kemampuan sistem dalam melaksanakan perintah berdasarkan variasi pengenalan suara (*voice recognition*) yang diucapkan oleh pengguna. Pengujian ini juga menganalisis stabilitas hubungan nirkabel pada berbagai variasi jarak dan penghalang [10][11].

3. Analisis Waktu Respon (*Latency Analysis*)

Mengukur durasi waktu (dalam satuan milidetik/detik) yang dibutuhkan sejak perintah diucapkan atau dikirim dari aplikasi hingga peralatan elektronik merespons. Pengujian kecepatan respon dilakukan untuk membandingkan performa sistem pada kondisi jaringan yang berbeda guna memastikan tingkat keberhasilan alat [10].



Gambar 7. Diagram Alur Penelitian

III. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil pengujian terhadap prototipe sistem kendali peralatan elektronik yang telah diimplementasikan pada dua ruangan, yaitu ruang tamu (room1) dan kamar tidur (room2). Pengujian dilakukan dalam dua skenario utama, yakni pengujian kontrol manual melalui antarmuka dasbor aplikasi web dan pengujian kontrol berbasis perintah suara maupun teks yang diproses melalui layanan Dialogflow ES. Selanjutnya, hasil dari kedua pengujian tersebut dianalisis untuk menilai keandalan, waktu tanggap (*response time*), serta kestabilan komunikasi data pada arsitektur sistem yang dibangun.

A. Pengujian Kontrol Manual

Pengujian kontrol manual dilakukan untuk memverifikasi bahwa tombol *toggle* pada dasbor aplikasi web mampu mengubah kondisi relay secara langsung tanpa melalui jalur *Natural Language Processing* (NLP). Pengujian ini bertujuan untuk memastikan bahwa jalur komunikasi inti, yaitu publikasi pesan melalui protokol MQTT menuju ESP32

dan sinkronisasi status ke Firebase *Realtime Database*, berfungsi dengan benar sebagai fondasi sebelum jalur NLP diuji lebih lanjut. Setiap percobaan dilakukan dengan menekan tombol *toggle* lampu pada masing-masing ruangan dari kondisi awal yang telah ditentukan, kemudian kondisi fisik relay dan lampu diamati secara langsung untuk dibandingkan dengan status yang ditampilkan pada dasbor. Hasil pengujian kontrol manual ditunjukkan pada Tabel I

Tabel 1. Hasil Pengujian Kontrol Manual pada Dasbor

No.	Ruangan	State Awal	Aksi	Hasil
1	Ruang Tamu (room1)	Mati (OFF)	Toggle ON	Lampu menyala, status dasbor sinkron
2	Ruang Tamu (room1)	Menyala (ON)	Toggle OFF	Lampu mati, status dasbor sinkron
3	Kamar Tidur (room2)	Mati (OFF)	Toggle ON	Lampu menyala, status dasbor sinkron
4	Kamar Tidur (room2)	Menyala (ON)	Toggle OFF	Lampu mati, status dasbor sinkron
5	Ruang Tamu & Kamar Tidur	Mati (OFF)	Nyalakan semua	Kedua lampu menyala bersamaan
6	Ruang Tamu & Kamar Tidur	Menyala (ON)	Matikan semua	Kedua lampu mati bersamaan

Berdasarkan Tabel I, keenam percobaan kontrol manual menunjukkan hasil yang konsisten, yaitu perubahan status relay pada ESP32 selalu sesuai dengan perintah *toggle* yang diberikan melalui dasbor, dan status tersebut langsung tersinkronisasi pada tampilan antarmuka melalui *listener* Firebase *Realtime Database*. Tidak ditemukan selisih kondisi antara status logis pada aplikasi web dan kondisi fisik lampu pada seluruh percobaan, sehingga jalur komunikasi MQTT dan Firebase pada skenario kontrol manual dapat dinyatakan berfungsi dengan baik dan layak dijadikan dasar bagi pengujian tahap berikutnya, yaitu pengujian kontrol berbasis suara dan teks.

B. Pengujian Kontrol Suara dan Teks

Pengujian kedua dilakukan untuk mengukur kinerja sistem ketika pengguna memberikan perintah melalui antarmuka percakapan (*chatbot*), baik dalam bentuk teks maupun suara yang *dikonversi* melalui *Web Speech API*. Setiap kata kunci atau perintah yang diucapkan/diketik diproses oleh layanan Dialogflow ES untuk menentukan *intent* dan parameter ruangan, kemudian diteruskan ke fungsi *processCommand()* yang menjalankan dua proses secara paralel, yaitu publikasi pesan MQTT (*mqttDispatchMs*) menuju ESP32 dan pembaruan status pada Firebase (*firebaseMs*). Waktu tanggap total (*totalMs*) dihitung sejak permintaan dikirim ke Dialogflow (*dialogflowMs*) hingga seluruh proses lokal (*processorMs*) selesai dieksekusi. Pengujian dilakukan sebanyak enam kali percobaan dengan variasi kata kunci kontrol lampu pada masing-masing ruangan serta perintah kontrol untuk seluruh ruangan sekaligus. Hasil pengukuran waktu tanggap ditunjukkan pada Tabel II.

Tabel 2. Hasil Pengujian Waktu Tanggap Kontrol Suara dan Teks

Perco baan ke-	Kata Kunci / Perintah	Dialog flow (ms)	MQTT (ms)	Fireba se (ms)	Proses Lokal (ms)	Total (ms)
1	“Nyalakan lampu ruang tamu”	4583	1	37	39	4622
2	“Nyalakan lampu kamar tidur”	2246	1	38	39	2286
3	“Matikan lampu ruang tamu”	721	1	36	37	759*
4	“Matikan lampu kamar tidur”	2258	1	40	42	2300
5	“Nyalakan semua lampu”	769	1	42	43	812
6	“Matikan semua lampu”	2379	1	49	51	2429

Pada percobaan ke-3 (matikan lampu ruang tamu), nilai *totalMs* sebesar 759 ms yang tercatat pada sistem merepresentasikan waktu pemrosesan sisi perangkat lunak (Dialogflow, *processor*, dan publikasi MQTT), namun secara

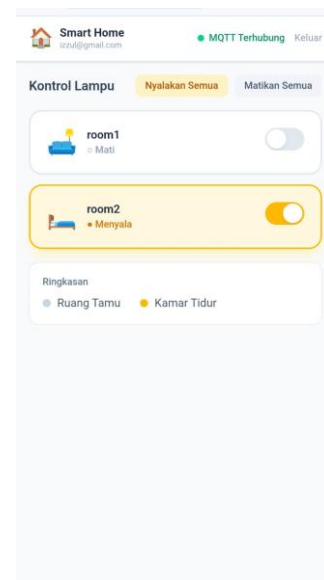
aktual perangkat ESP32 mengalami kegagalan eksekusi pada percobaan pertama akibat gangguan koneksi MQTT, sebagaimana diuraikan pada paragraf berikut.

Pada percobaan ke-3, tercatat anomali berupa terputusnya koneksi MQTT antara ESP32 dan *broker* HiveMQ Cloud tepat ketika permintaan untuk mematikan lampu ruang tamu dikirimkan dari aplikasi web. Catatan *serial monitor* pada ESP32 menunjukkan galat pada lapisan keamanan TLS/SSL, yaitu “Terminating because the ssl engine closed” yang diikuti dengan kegagalan penulisan data (“Failed while waiting for the engine to enter BR_SSL_SENDAPP”) dan diakhiri dengan waktu habis pada proses inisialisasi ulang lapisan SSL (“SSL internals timed out”). Akibatnya, pesan MQTT pertama yang dipublikasikan oleh aplikasi web tidak diterima oleh ESP32 karena koneksi dalam kondisi terputus. Mekanisme *auto-reconnect* yang diimplementasikan pada fungsi *connectMqtt()* kemudian berhasil membangun ulang koneksi dalam rentang waktu sekitar satu menit, ditandai dengan pesan “[MQTT] Menghubungkan sebagai esp32-lights-32e9bfb4 ... Terhubung!”, dan dilanjutkan dengan proses *subscribe* ulang ke topik *home/lights/room1* dan *home/lights/room2*. Perintah mematikan lampu ruang tamu baru berhasil dieksekusi setelah koneksi pulih, yang dibuktikan dengan log “[MQTT] Pesan diterima [home/lights/room1]: 10” yang diikuti perubahan status relay menjadi *OFF*. Dengan demikian, perintah dari sisi pengguna tidak hilang, melainkan tertunda hingga koneksi MQTT berhasil dipulihkan secara otomatis.

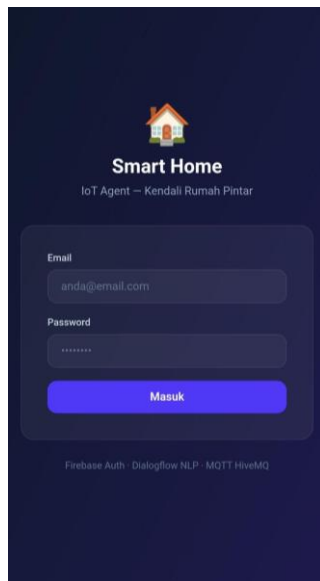
C. Hasil Analisa



Gambar 8. Fitur Chatbot Berbasis Voice Recognition dan NLP



Gambar 9. Dasbor Kontrol Lampu Manual



Gambar 10. Halaman Login Pengguna

Hasil pengujian kontrol manual dan kontrol suara/teks pada Tabel I dan Tabel II memberikan beberapa temuan yang dapat dianalisis sebagai berikut:

1. Dominasi waktu pemrosesan NLP terhadap total waktu tanggap. Dari enam percobaan kontrol suara dan teks, rata-rata waktu tanggap total (*totalMs*) tercatat sebesar 2.201,33 ms, dengan rata-rata waktu pemrosesan Dialogflow (*dialogflowMs*) sebesar 2.159,33 ms, atau setara dengan sekitar 98,1% dari keseluruhan waktu tanggap. Sebaliknya, proses lokal pada sisi aplikasi web yang mencakup publikasi MQTT dan pembaruan Firebase hanya menyumbang rata-rata 41,83 ms, atau kurang dari 2% dari total waktu. Temuan ini menunjukkan bahwa waktu tanggap sistem secara keseluruhan sangat bergantung pada latensi komunikasi menuju layanan *cloud* NLP, sedangkan jalur komunikasi MQTT dan Firebase yang berjalan secara lokal/paralel relatif tidak signifikan memengaruhi total waktu tanggap.
2. Variasi waktu respons Dialogflow antarpercobaan. Nilai *dialogflowMs* pada keenam percobaan menunjukkan rentang yang cukup lebar, yaitu dari 721 ms (percobaan ke-3) hingga 4.583 ms (percobaan ke-1). Waktu yang relatif tinggi

pada percobaan pertama diperkirakan disebabkan oleh fenomena *cold start*, yaitu kondisi ketika sesi permintaan pertama ke layanan Dialogflow ES memerlukan waktu inisialisasi lebih lama dibandingkan permintaan-permintaan berikutnya yang telah memanfaatkan koneksi atau *cache* sesi yang sama. Apabila percobaan pertama tersebut dikeluarkan dari perhitungan sebagai data pencilon (*outlier*), rata-rata waktu tanggap total pada lima percobaan tersisa menurun menjadi 1.717,20 ms dan rata-rata waktu Dialogflow menjadi 1.674,60 ms. Penurunan ini menunjukkan bahwa setelah sesi pertama terbentuk, sistem mampu memberikan waktu tanggap yang lebih stabil dan konsisten pada kisaran 0,7 hingga 2,4 detik untuk setiap kata kunci atau perintah suara/teks yang diberikan.

3. Efisiensi jalur komunikasi lokal (MQTT dan Firebase). Waktu publikasi pesan MQTT (*mqttDispatchMs*) secara konsisten tercatat 1 ms pada seluruh percobaan, sedangkan waktu pembaruan Firebase (*firebaseMs*) berkisar antara 36 hingga 49 ms. Nilai ini relatif kecil karena kedua proses dijalankan secara paralel sehingga tidak saling menunggu satu sama lain. Hal ini menunjukkan bahwa arsitektur yang memisahkan jalur komunikasi data (*real-time database*) dari jalur kendali perangkat (MQTT *broker*) terbukti efisien dan tidak menjadi sumber *bottleneck* pada keseluruhan proses kendali lampu.
4. Ketahanan sistem terhadap gangguan koneksi MQTT. Insiden terputusnya koneksi SSL/TLS pada percobaan ke-3 menunjukkan bahwa koneksi MQTT melalui jaringan internet publik menuju broker HiveMQ Cloud berpotensi mengalami gangguan sewaktu-waktu, terutama pada lapisan keamanan TLS yang memerlukan proses *handshake* tambahan dibandingkan koneksi tanpa enkripsi. Meskipun demikian, mekanisme *reconnect* otomatis yang diimplementasikan pada firmware ESP32 terbukti mampu

memulihkan koneksi tanpa intervensi manual, serta tidak menghilangkan perintah yang dikirimkan oleh pengguna meskipun terjadi penundaan eksekusi. Hal ini menunjukkan bahwa kombinasi protokol MQTT dengan parameter *Quality of Service* (QoS) 1 serta status *retain* pada pesan kontrol lampu memberikan tingkat keandalan tambahan terhadap gangguan jaringan yang bersifat sementara, sekalipun waktu tunda yang ditimbulkan, dalam kasus ini sekitar satu menit, cukup signifikan dan perlu menjadi perhatian untuk pengembangan lebih lanjut, misalnya dengan mempersingkat durasi *keep-alive* atau menambahkan mekanisme deteksi dini kegagalan koneksi.

5. Perbandingan kontrol manual dan kontrol suara/teks. Apabila dibandingkan dengan hasil pengujian kontrol manual pada Tabel I yang seluruhnya menunjukkan respons instan tanpa keterlibatan proses NLP, dapat disimpulkan bahwa selisih waktu tanggap antara kedua mode kendali tersebut terutama disebabkan oleh waktu komunikasi menuju layanan Dialogflow ES yang berada di luar infrastruktur lokal. Meskipun waktu tanggap kontrol suara dan teks secara rata-rata berada pada kisaran 0,8 hingga 4,6 detik, durasi tersebut masih berada dalam batas wajar bagi interaksi percakapan (*conversational interface*) dan tidak mengurangi fungsionalitas utama sistem dalam mengendalikan kedua titik lampu pada ruang tamu dan kamar tidur secara akurat sesuai dengan *intent* yang dikenali.

IV. KESIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan prototipe sistem kendali peralatan elektronik berbasis perintah suara dan teks pada dua ruangan, yaitu ruang tamu dan kamar tidur, dengan memanfaatkan mikrokontroler ESP32, layanan *Natural Language Processing* Dialogflow ES, protokol komunikasi MQTT melalui *broker* HiveMQ Cloud, serta *Firestore Realtime Database* sebagai

media sinkronisasi status antarperangkat. Berdasarkan hasil pengujian yang telah diuraikan pada bagian sebelumnya, dapat disimpulkan bahwa tujuan penelitian untuk mewujudkan sistem kendali peralatan elektronik rumah tangga yang dapat dioperasikan melalui perintah bahasa alami maupun secara manual telah tercapai, baik melalui jalur kontrol langsung pada dasbor aplikasi web maupun melalui jalur percakapan berbasis NLP.

Pada skenario kontrol manual, seluruh percobaan menunjukkan kesesuaian penuh antara perintah yang diberikan melalui dasbor dengan kondisi fisik relay dan status yang ditampilkan pada antarmuka, sehingga jalur komunikasi inti yang menghubungkan aplikasi web, *broker* MQTT, ESP32, dan *Firestore Realtime Database* terbukti berfungsi dengan andal sebagai fondasi sistem. Sementara itu, pada skenario kontrol berbasis suara dan teks, sistem mampu mengenali *intent* serta parameter ruangan secara konsisten melalui Dialogflow ES, dengan rata-rata waktu tanggap total sebesar 2.201,33 milidetik, di mana sekitar 98,1% dari waktu tersebut dikontribusikan oleh proses komunikasi menuju layanan NLP, sedangkan proses lokal yang mencakup publikasi MQTT dan pembaruan *Firestore* hanya menyumbang kurang dari dua persen dari keseluruhan waktu tanggap. Temuan ini memberikan validasi empiris bahwa arsitektur yang memisahkan jalur kendali perangkat dari jalur komunikasi data memiliki efisiensi struktural yang tinggi. Meskipun pengujian dilakukan dalam skala prototipe domestik, data menunjukkan secara konsisten bahwa batas atas (*bottleneck*) waktu tanggap sistem secara dominan ditentukan oleh latensi *cloud* pihak ketiga, bukan oleh kapasitas pemrosesan lokal perangkat keras maupun beban sinkronisasi basis data. Hal ini membuktikan bahwa fondasi topologi sistem yang dirancang memiliki skalabilitas yang baik untuk dikembangkan lebih lanjut.

Hasil pengujian juga mengonfirmasi keandalan fungsional (*fault tolerance*) sistem terhadap anomali jaringan yang bersifat temporer. Terjadinya insiden disosiasi koneksi aman (TLS/SSL) antara ESP32 dan *broker* MQTT

menjadi bukti autentik efektivitas mekanisme pemulihan yang dirancang. Melalui implementasi *auto-reconnect* yang dipadukan dengan parameter *Quality of Service* (QoS) 1 dan status *retain* pada pesan, sistem menjamin nol kegagalan pengiriman perintah (*zero packet loss*) meskipun terjadi penundaan eksekusi. Karakteristik ini menegaskan bahwa sistem tidak hanya andal pada kondisi ideal, melainkan memiliki retensi ketahanan yang adaptif untuk diimplementasikan pada fluktuasi jaringan internet domestik.

Dengan demikian, walaupun pengujian ini berbasis prototipe ruangan terbatas, pemodelan ini telah memenuhi kriteria minimum sebagai solusi rumah pintar yang tangguh, andal, dan siap diadaptasikan pada dinamika jaringan lingkungan rumah tangga.

REFERENSI

- [1] Cakra, M. S. Said, and Henny, "Sistem kontrol lampu menggunakan sensor suara," *Simtek : jurnal sistem informasi dan teknik komputer*, vol. 8, no. 1, pp. 77–82, Apr. 2023, doi: 10.51876/simtek.v8i1.218.
- [2] M. Amiruddin, "Implementasi Sistem Kontrol Stopkontak Elektronik Cerdas Menggunakan Aktivasi Suara dan Integrasi Pengenalan Wajah." [Online]. Available: <http://eprints.ums.ac.id/id/eprint/124391>.
- [3] G. Alexakis, S. Panagiotakis, A. Fragakakis, E. Markakis, and K. Vassilakis, "Control of Smart Home Operations Using Natural Language Processing, Voice Recognition and IoT Technologies in a Multi-Tier Architecture," *Designs*, vol. 3, no. 3, p. 32, Jul. 2019, doi: 10.3390/designs3030032.
- [4] Mohamad Salman Farizi, S. Somantri, and I. Yustiana, "IMPLEMENTASI SPEECH RECOGNITION PADA SISTEM KENDALI PERANGKAT ELEKTRONIK RUMAH BERBASIS IoT (Internet Of Things) DAN MOBILE APPLICATION," *ZONasi: Jurnal Sistem Informasi*, vol. 4, no. 2, pp. 157–166, Dec. 2022, doi: 10.31849/zn.v4i2.10662.
- [5] A. P. Launuru, G. Manu, H. K. Tupan, and R. Hutagalung, "RANCANG BANGUN SISTEM KONTROL NIRKABEL ON – OFF PERALATAN LISTRIK DENGAN PERINTAH SUARA MENGGUNAKAN SMARTPHONE ANDROID," *JURNAL SIMETRIK*, vol. 11, no. 1, pp. 388–397, Aug. 2021, doi: 10.31959/js.v11i1.570.
- [6] H. Haeruddin, "Sistem Pengendali Alat Elektronik Rumah Tangga Berbasis Internet of Things (IoT) Menggunakan Voice Recognition," *Journal of System and Computer Engineering (JSCE)*, vol. 5, no. 1, pp. 56–65, Jan. 2024, doi: 10.61628/jsce.v5i1.1063.
- [7] O. D. Winarto, I. K. Somawirata, and Sotyohadi, "DESAIN SISTEM KENDALI KURSI RODA ELEKTRIK DENGAN PERINTAH SUARA BERBASIS VOICE RECOGNITION MODULE V3," *Magnetika: Jurnal Mahasiswa Teknik Elektro*, vol. 8, no. 2, pp. 213–223, Nov. 2024.
- [8] D. P. Utomo, "Implementasi Internet of Things (IoT) Untuk Sistem Kendali Pada Smart Lamp," *Prosiding Seminar Nasional Informatika*, vol. 2, pp. 540–551, Jul. 2024.
- [9] R. Rizky Ababil, Y. Yandi Bria, S. M. Rosyid Aridho, and P. Pramono, "Sistem Smart Lighting Berbasis Internet of Things (IoT) dengan Aplikasi Blynk," *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis*, pp. 1256–1261, Jul. 2025, doi: 10.47701/5add4003.
- [10] D. Karina, "PENERAPAN INTERNET OF THINGS PADA SISTEM SMART HOME MEMANFAATKAN VOICE RECOGNITION," *Techno Bahari*, vol. 8, no. 2, May 2023, doi: 10.52234/tb.v8i2.145.
- [11] R. Wahyuningrum and L. Febrianto, "Rancang Bangun Prototype Sistem Kontrol Kunci Pintu Berbasis Voice Recognition Arduino Uno & Sensor Bluetooth," *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi dan Sistem Komputer*, vol. 7, no. 2, pp. 78–85, Oct. 2023, doi: 10.55886/infokom.v7i2.755.