

SISTEM *SMART HOME* BERBASIS IoT DENGAN INTEGRASI PENGENDALIAN SUARA DAN APLIKASI *SMARTPHONE*

Mufid Ridlo Effendi¹, Nanang Ismail^{1*)}, Nathan Rizqi Evandi¹ dan Hanif Fakhurroja²

¹Departemen Teknik Elektro, UIN Sunan Gunung Djati, Bandung, Indonesia

²Pusat Penelitian Mekatronika Cerdas, Badan Riset dan Inovasi Nasional, Bandung, Indonesia

^{*)}E-mail: nanang.is@uinsgd.ac.id

Abstrak

Salah satu penerapan dari teknologi IoT adalah sebuah sistem *smart home* yang dapat mengendalikan penggunaan peralatan rumah tangga secara lebih praktis. Penelitian ini mengusulkan sistem *smart home* yang mengontrol peralatan rumah tangga berbasis Internet of Things (IoT). Sistem yang dirancang memanfaatkan mikrokontroler NodeMCU, basis data Firebase, dan aplikasi Android berbasis Flutter. Sistem yang diusulkan memiliki 4 fitur utama. Fitur pertama memungkinkan kontrol perangkat rumah tangga melalui perintah suara menggunakan Flutter Speech API yang mendukung komunikasi berbahasa Indonesia. Fitur yang dikembangkan mencakup pengendalian lampu, televisi, dan *sound system*. Fitur pengendalian berbasis perintah suara diuji dengan variasi jarak dan lingkungan ber-noise. Fitur kedua adalah fitur EcoMode yang berfungsi untuk menyalakan-matikan lampu secara terjadwal. Fitur ini menggunakan *Network Time Protocol* (NTP) untuk menentukan waktu secara *real-time* tanpa modul *Real Time Clock* (RTC) tambahan. Fitur ketiga adalah fitur pemantauan konsumsi daya listrik menggunakan sensor arus SCT-013 yang diletakkan pada rangkaian *breaker*. Nilai daya diperoleh dengan mengalikan arus dengan tegangan 220V. Fitur keempat adalah fitur *smart remote* yang dapat mengendalikan peralatan rumah tangga melalui aplikasi pada *smartphone* layaknya *remote control*. Sistem *smart home* ini terbukti menawarkan efisiensi dan kemudahan pengelolaan perangkat rumah tangga dalam satu aplikasi.

Kata kunci: Smart home, IoT, perintah suara, smartphone.

Abstract

One application of IoT technology is a smart home system that enables more practical control of household appliances. This study proposes a smart home system that controls household appliances based on the Internet of Things (IoT). The system designed utilizes a NodeMCU microcontroller, Firebase database, and Flutter-based Android application. The proposed system has four main features. The first feature enables appliance control via voice commands using the Flutter Speech API, which supports communication in Indonesian. This feature allows users to control lights, televisions, and sound systems. The voice command-based control was tested under varying distances and noise levels. The second feature, EcoMode, schedules automatic switching of lights on and off. This feature utilizes the Network Time Protocol (NTP) to determine real-time scheduling without requiring an additional Real-Time Clock (RTC) module. The third feature monitors electricity consumption using an SCT-013 current sensor placed on the breaker circuit. The power consumption is calculated by multiplying the current by a fixed voltage of 220V. The fourth feature is the smart remote, which allows users to control household appliances through a smartphone application, functioning like a remote control. The proposed system demonstrates efficiency and convenience in managing household appliances through a single application.

Keywords: Smart home, IoT, voice command, smartphone

1. Pendahuluan

Internet of Things (IoT) adalah sebuah paradigma teknologi baru yang dapat mewujudkan komunikasi antar mesin atau antar manusia dan mesin melalui jaringan internet [1][2][3]. "Things" dalam IoT bisa berupa peralatan yang dapat dikendalikan dan dipantau melalui jaringan internet [4]. IoT dapat diaplikasikan di bidang transportasi, kesehatan, perikanan, pertanian, lingkungan, keamanan, dan otomasi rumah [5][6][7]. Salah satu aplikasi utama IoT adalah

sistem *smart home*, yang memungkinkan otomasi dan pengendalian peralatan rumah tangga untuk meningkatkan efisiensi, kenyamanan, dan keamanan bagi penghuni rumah [8][9][10].

Sistem *smart home* dapat melakukan monitoring terhadap kondisi yang terjadi pada rumah, seperti kelembapan, suhu, intensitas cahaya, kebocoran gas, konsumsi daya, hingga keamanan atau anti maling [7][11][12]. Selain itu, *smart home* dapat mewujudkan sistem integrasi antar perangkat

rumah tangga [7][12][13]. Beberapa peralatan rumah tangga yang tersedia seperti televisi, pendingin ruangan, lampu, dan komputer dapat terintegrasi oleh teknologi IoT [6][13][14]. Pengelolaan data sistem berbasis IoT memanfaatkan layanan *cloud* sebagaimana penelitian pada paper [15].

Sistem *smart home* berbasis IoT telah banyak diteliti dengan berbagai pendekatan teknologi dan tujuan. Gupta dan Johari [4] mengembangkan sistem pengendalian peralatan listrik berbasis IoT menggunakan Raspberry Pi, namun sistem ini terbatas pada kontrol berbasis web tanpa fitur pengendalian suara atau pemantauan konsumsi daya *real-time*. Yildiz dan Burunkaya [6] mengusulkan sistem pemantauan daya menggunakan mikrokontroler ESP8266, tetapi tidak mengintegrasikan otomasi berbasis jadwal atau pengendalian suara. Evandi et al. [5] mengembangkan aplikasi berbasis *smartphone* untuk mengendalikan televisi menggunakan teknologi IoT, namun sistem ini tidak mencakup pengendalian beragam perangkat atau pemantauan energi. Sementara itu, paper [16] membuat sistem pemantauan dan pengendalian beban listrik di ruangan. Rahayu dan Hendri [17] mengembangkan sistem *smart home* berbasis Voice Recognition Module V3 untuk mengendalikan peralatan elektronik, pintu, dan jendela melalui perintah suara, dengan tingkat keberhasilan pengenalan suara rata-rata 80.2%. Namun, sistem mereka tidak mengintegrasikan pemantauan konsumsi daya atau otomasi berbasis jadwal, dan akurasi pengenalan suara menurun jika pengucapan tidak jelas atau pada jarak lebih jauh. Alaa et al. [3] dalam *literature review*-nya mencatat bahwa banyak sistem *smart home* berfokus pada pengendalian perangkat atau pemantauan lingkungan, tetapi jarang yang mengintegrasikan pengendalian suara, otomasi terjadwal, dan pemantauan daya dalam satu *platform*. Kekurangan utama dari penelitian-penelitian ini adalah kurangnya solusi terintegrasi yang mencakup semua fitur tersebut, serta tantangan dalam beragam format pengendalian [3].

Berdasarkan *state of the art* tersebut, penelitian ini mengusulkan sistem *smart home* berbasis IoT yang mengintegrasikan empat fitur utama: (1) pengendalian perangkat melalui perintah suara menggunakan Flutter Speech API yang mendukung bahasa Indonesia, (2) fitur EcoMode untuk otomasi lampu berbasis jadwal menggunakan *Network Time Protocol* (NTP), (3) pemantauan konsumsi daya secara *real-time* dengan sensor arus SCT-013, dan (4) fitur *smart remote* untuk pengendalian perangkat melalui aplikasi *smartphone*. Pendekatan ini dipilih untuk mengatasi keterbatasan penelitian sebelumnya dengan menyediakan solusi yang lebih terintegrasi, efisien, dan praktis. Pemilihan NodeMCU ESP8266 sebagai mikrokontroler utama didasarkan pada kemampuan konektivitas WiFi, biaya rendah, dan kompatibilitas dengan Firebase sebagai basis data *real-time* [18]. Penggunaan Flutter untuk aplikasi Android dipilih karena kemampuan pengembangan

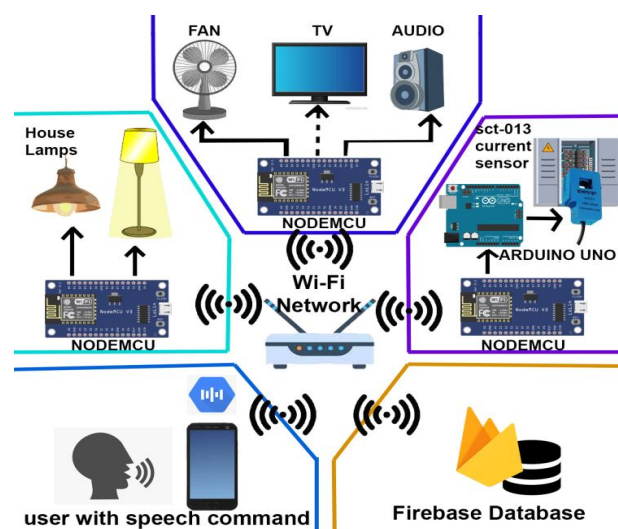
antarmuka *multi-platform* yang responsif dan efisien [19]. Pendekatan ini didukung oleh penelitian Ayeni dan Adesoba [18], yang menunjukkan bahwa sistem berbasis NodeMCU dan Firebase memungkinkan komunikasi *real-time* yang andal untuk mengendalikan peralatan rumah tangga dan memantau parameter lingkungan dalam aplikasi *smart home*.

Penelitian ini bertujuan untuk mengatasi masalah umum dalam sistem *smart home*, yaitu kurangnya integrasi antara pemantauan konsumsi daya, pengendalian otomatis berbasis jadwal, dan pengendalian suara dalam satu aplikasi. Sistem yang diusulkan diharapkan dapat meningkatkan efisiensi energi, kemudahan penggunaan, dan pengontrolan secara praktis dalam satu aplikasi.

2. Sistem *Smart home* yang Diusulkan

2.1. Model Sistem

Sistem yang dirancang pada penelitian ini bertujuan untuk mengendalikan perangkat rumah tangga seperti lampu, televisi, kipas angin, dan *sound system* sekaligus memantau konsumsi daya listrik. Model sistem memanfaatkan kombinasi perangkat keras dan perangkat lunak berbasis IoT untuk mewujudkan kendali dan monitoring yang efisien. Gambar 1 menunjukkan model sistem *smart home* yang dikembangkan, yang merupakan pengembangan dari model yang diusulkan oleh Evandi et al. [5]. Model asli dalam referensi [5] hanya mencakup pengendalian televisi melalui aplikasi *smartphone* berbasis IoT tanpa fitur pemantauan daya atau otomasi berbasis jadwal. Dalam penelitian ini, model tersebut diperluas dengan menambahkan fitur EcoMode, pemantauan konsumsi daya menggunakan sensor SCT-013, dan pengendalian suara berbasis Flutter Speech API. Penambahan ini dilakukan untuk menciptakan sistem yang lebih komprehensif dan responsif terhadap kebutuhan rumah tangga modern.

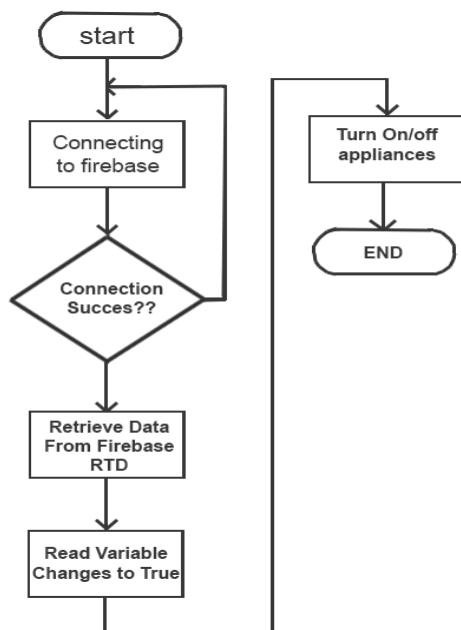


Gambar 1. Model Sistem

NodeMCU ESP8266 berfungsi sebagai pengendali utama yang menghubungkan perangkat rumah tangga dengan jaringan internet. NodeMCU saling terhubung melalui jaringan WiFi. Tidak ada setting routing khusus dalam koneksi ini, SSID dan *password* sudah ditanam dalam program NodeMCU sehingga bisa langsung terhubung ke WiFi. Konfigurasi TCP/IP yang ada berbasis DHCP *server*. Setiap NodeMCU memiliki peran spesifik, seperti mengendalikan perangkat rumah tangga atau membaca data dari sensor.

Data yang dikumpulkan akan dikirimkan ke Firebase untuk diproses dan ditampilkan melalui aplikasi Android. Firebase *realtime database* digunakan untuk menyimpan data secara *real-time*, termasuk status perangkat dan data konsumsi daya. Aplikasi Android berbasis Flutter berfungsi sebagai antarmuka pengguna untuk mengendalikan perangkat rumah tangga dan memantau data. Sensor SCT-013 bertugas membaca arus listrik untuk memantau konsumsi daya. Arduino Uno digunakan sebagai pengolah data dari sensor SCT-013 sebelum dikirimkan ke NodeMCU. Relay digunakan untuk mengontrol perangkat berbasis stop kontak, sedangkan pemancar *infrared* digunakan untuk mengendalikan perangkat seperti televisi.

Untuk mengatasi latensi dan keamanan jaringan, sistem menggunakan *Quality of Service* (QoS) level 1 pada protokol MQTT untuk komunikasi antara NodeMCU dan Firebase, memastikan pengiriman data yang andal meskipun dalam kondisi jaringan yang tidak stabil [20][21][22]. Konfigurasi ini meningkatkan efisiensi komunikasi dan mengurangi risiko kehilangan data, sebagaimana didukung oleh penelitian Zhao et al. [20].

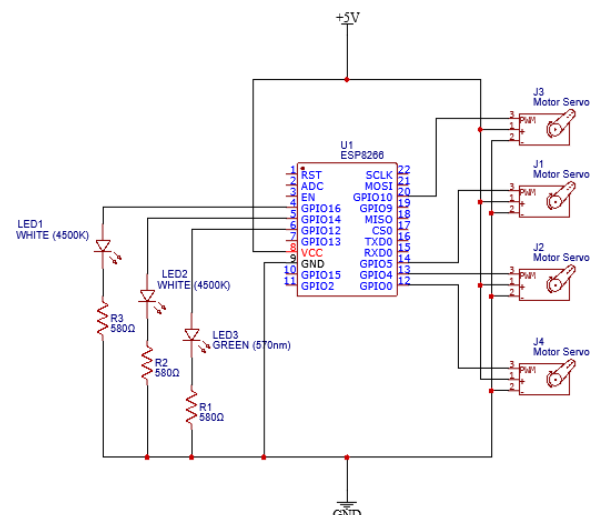


Gambar 2. Flowchart Proses Pengendalian

Gambar 2 menunjukkan *flowchart* proses pengendalian pada sistem *smart home*. Pengguna memberikan perintah melalui aplikasi Android, baik secara manual maupun menggunakan suara. Aplikasi mengubah perintah tersebut menjadi data yang dikirimkan ke Firebase. NodeMCU membaca perubahan variabel *boolean* dalam data di Firebase dan menerjemahkannya menjadi aksi pada perangkat, seperti menyalakan atau mematikan perangkat.

Sebagai contoh, perintah yang diterima misalnya “nyalakan TV”, maka variabel “TV” pada *real-time database* akan berubah menjadi *true*. Kemudian, NodeMCU yang terhubung dengan TV akan menyalakan televisi melalui pemancaran *infrared*.

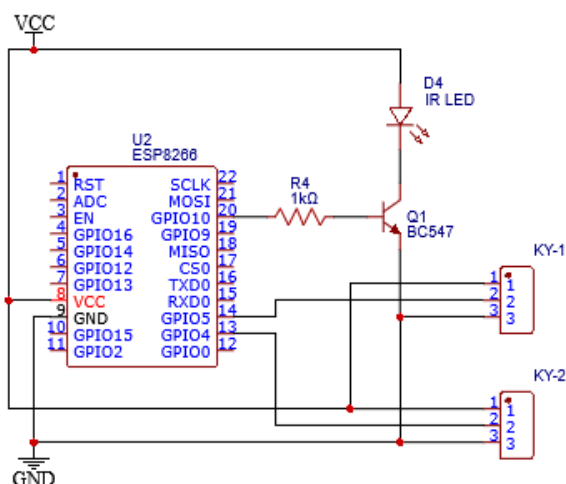
Pengendalian lampu memerlukan servo. Servo bertugas menekan saklar lampu secara mekanik. Gambar 3 menunjukkan skema rangkaian pengendali lampu dengan menggunakan servo. Servo tersebut ditempatkan pada 2 sisi saklar lampu, servo sisi atas merupakan servo untuk menyalakan dan servo sisi bawah merupakan servo yang berguna untuk mematikan. Ada 2 lampu yang dikendalikan yaitu lampu ruang tamu dan lampu teras. Terdapat 3 buah lampu LED yang berfungsi sebagai indikator apakah lampu tersebut menyala atau tidak dan indikator untuk fitur “EcoMode”. Fitur pengendali lampu tersebut berfungsi untuk menyala-matikan lampu secara otomatis pada jam tertentu menggunakan *Network Time Protocol* (NTP) untuk menentukan waktu secara *real-time* tanpa modul *Real Time Clock* (RTC) tambahan. Fitur tersebut berfungsi untuk menyala-matikan lampu secara otomatis pada jam tertentu, yaitu menyala pada jam 18.00 dan mati pada jam 07:00.



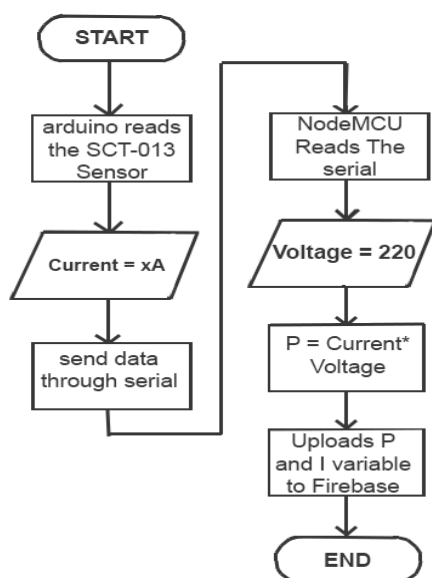
Gambar 3. Rangkaian Pengendali Lampu Menggunakan Servo

Gambar 4 menunjukkan rangkaian pengendali peralatan elektronik pada ruangan keluarga. Peralatan yang dikendalikan adalah televisi, *sound system*, dan kipas

angin. Televisi dikendalikan melalui pemancar *infrared*, sementara *sound system* dan kipas angin dikendalikan melalui modul relay pada stop kontak. Modul relay tersebut bekerja pada tegangan 5V yang bersumber dari catu daya eksternal.



Gambar 4. Skema Rangkaian Pengendali Ruang Keluarga

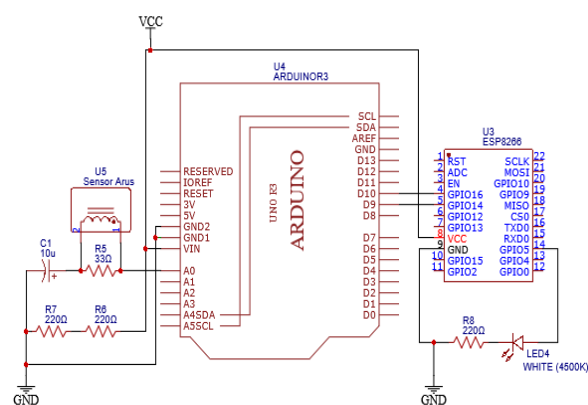


Gambar 5. Flowchart Pembacaan Sensor Arus

Selain NodeMCU yang mengendalikan peralatan, ada NodeMCU lain yang membaca sensor arus SCT-013 yang diletakan pada rangkaian *breaker*. Skema pembacaan sensor pada sistem ini dijelaskan pada Gambar 5. Pembacaan sensor arus dibantu oleh Arduino Uno yang berperan sebagai driver dari sensor arus SCT-013. Mikrokontroler yang bekerja pada bagian pembaca sensor arus ada 2. Arduino bertugas untuk mengubah sinyal analog dari SCT-013 menjadi variabel *float*. Variabel *float* selanjutnya dikirim ke NodeMCU melalui komunikasi

serial. Data arus dari Arduino selanjutnya dikalikan dengan variabel tegangan konstan 220 V untuk mendapatkan nilai daya (P). Nilai P dan I di-*upload* ke *Firestore real time database* melalui jaringan internet. Variabel P dan I yang telah terupload dapat dimonitor oleh *user* melalui antarmuka aplikasi *mobile* berbasis Flutter.

Skema rangkaian dari sensor arus ditampilkan pada Gambar 6. Komponen-komponen yang ada pada rangkaian tersebut adalah sensor arus SCT-013, rangkaian *driver* SCT-013 yang terdiri dari 3 *resistor* (2 *resistor* pembagi tegangan dan 1 *resistor* Burden) dan 1 kapasitor, Arduino uno, Node MCU, dan LED *indicator*. Rangkaian ini menggunakan *resistor* Burden sebesar 33Ω karena catu daya untuk rangkain ini menggunakan catu daya 5V yang disalurkan melalui USB Port. Terdapat indikator LED yang berfungsi sebagai penanda bahwa NodeMCU sedang melakukan komunikasi serial dengan Arduino UNO.



Gambar 6. Skema Rangkaian Pembaca Arus

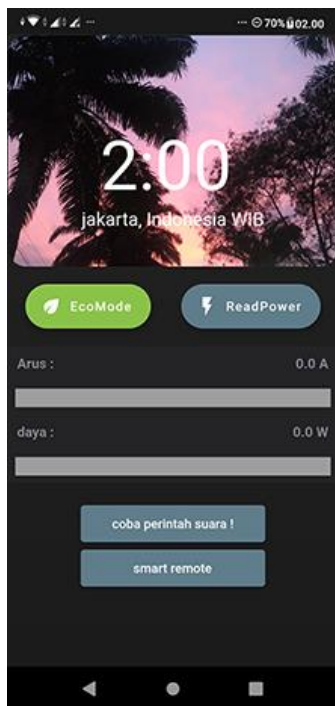
2.2. Aplikasi Berbasis Android

Aplikasi berbasis *smartphone* yang dirancang pada penelitian ini menggunakan *framework* Flutter. Flutter merupakan *framework* yang dikembangkan Google untuk membangun aplikasi *multi-platform*. Flutter menggunakan bahasa pemrograman Dart dan dapat membangun aplikasi berbasis Android dan iOS [19]. Aplikasi yang dibuat memiliki tiga bagian, yaitu halaman beranda, halaman kontrol suara, dan halaman *smart remote*. Halaman beranda menyediakan fitur EcoMode dan ReadPower dan menyediakan link ke fitur kontrol suara dan fitur *smart remote*.

A. Halaman Beranda

Halaman beranda aplikasi ditunjukkan oleh Gambar 7. Halaman ini menampilkan status perangkat rumah tangga, konsumsi daya, dan *progress bar* yang menggambarkan total penggunaan daya rumah secara keseluruhan. *Progress bar* yang ada di halaman beranda menunjukan konsumsi daya dan arus pada rumah. Data yang diambil merupakan data yang terdapat pada *real time database*. Nilai konsumsi

daya dan arus ditunjukkan oleh *progress bar* ketika *user* menekan tombol “ReadPower”. Data arus dan daya dibaca secara *real-time*, nilai yang ditampilkan akan selalu diperbaharui setiap terjadinya perubahan data pada *real time database*.



Gambar 7. Halaman Beranda Aplikasi

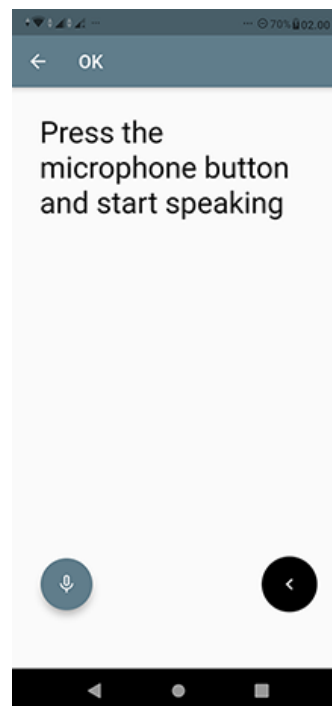
Pada halaman beranda juga terdapat tombol “EcoMode” yang berfungsi mengaktifkan lampu otomatis. Fitur ini dapat menyalakan dan mematikan lampu sesuai jadwal. Saat EcoMode diaktifkan, *user* tidak bisa mengendalikan lampu secara manual.

Pada bagian bawah dari halaman beranda, terdapat 2 tombol menu yang bertuliskan “coba perintah suara!” dan “*smart remote*”. Kedua tombol ini berfungsi untuk menjalankan fitur lain dari aplikasi ini. Kedua fitur aplikasi aplikasi tersebut berturut-turut adalah fitur kontrol ucapan dan *smart remote*.

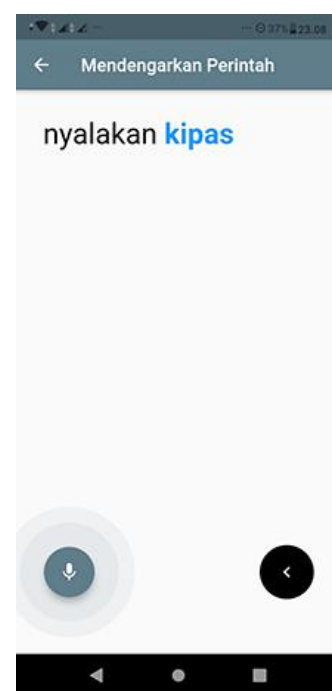
B. Fitur Kontrol Berbasis Suara

Fitur kontrol suara merupakan fitur yang dapat menerima input suara dari *user* dan menjadikannya sebagai perintah untuk mengontrol peralatan rumah tangga. Gambar 8 merupakan tampilan dari UI kontrol ucapan. Halaman ini muncul setelah *user* mengklik tombol menu “coba perintah suara!”. Kontrol ucapan menggunakan fitur dari *framework* Flutter. Pengguna dapat memberikan perintah suara melalui tombol mikrofon. Aplikasi akan mendeteksi ucapan pengguna menggunakan Flutter Speech API.

Perintah suara yang diterima ditampilkan dalam bentuk teks pada layar sehingga pengguna dapat memverifikasi input mereka, sebagaimana terlihat pada Gambar 9. Fitur ini dirancang agar tetap berfungsi dengan baik bahkan dalam kondisi dengan gangguan *noise* moderat. Beberapa frase perintah sudah diset terlebih dahulu di dalam sistem.



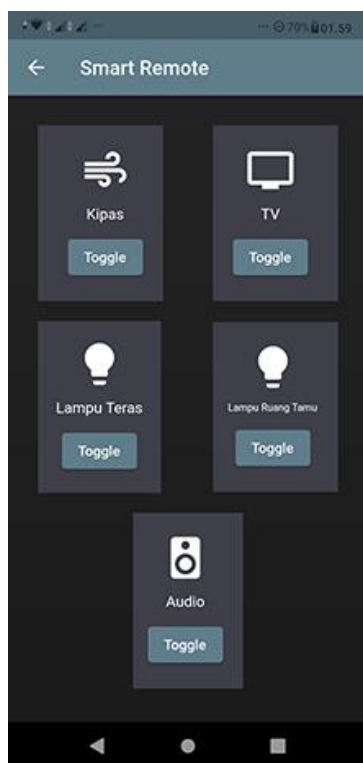
Gambar 8. Halaman Fitur Kontrol Suara



Gambar 9. Proses Mendengar Perintah Suara User

C. Fitur Smart Remote

Bagian ini berfungsi sebagai *remote control* digital untuk perangkat rumah tangga. Tombol-tombol kontrol dilengkapi indikator visual (seperti perubahan warna) untuk menunjukkan status perangkat. Gambar 10 menunjukkan halaman *smart remote control* pada aplikasi.



Gambar 10. Halaman *Smart Remote*

Peralatan rumah tangga yang tersedia pada bagian ini merupakan peralatan yang dapat dikendalikan oleh *user*. Jika *user* menekan tombol *toggle*, maka tampilan tombol akan berubah menjadi biru dan tulisan pada tombol akan berubah menjadi “on”. Jika *user* menekan kembali tombol tersebut, maka tombol akan berubah menjadi “Toggle” kembali dan peralatan akan mati. Pada sistem ini, peralatan yang dikontrol sesuai dengan tampilan pada *smart remote*.

Table 1. Tingkat Keberhasilan Kendali Berbasis Suara

Perintah	Tingkat Keberhasilan (%)							
	Orang I	Orang II	Orang III	Orang IV	Orang V	Orang VI	Orang VII	Orang VIII
“nyalakan lampu teras”	86,67	83,33	86,67	83,33	96,67	90	83,33	76,67
“matikan lampu teras”	90	80	86,67	76,67	83,33	83,33	73,33	83,33
“matikan lampu ruang tamu”	70	83,33	80	70	63,33	73,33	66,67	76,67
“nyalakan lampu ruang tamu”	90	86,67	90	83,33	83,33	90	83,33	96,67
“nyalakan kipas”	100	93,33	96,67	96,67	80	93,33	93,33	100
“matikan kipas”	96,67	93,33	100	96,67	83,33	96,67	93,33	100
“nyalakan TV”	90	96,67	100	90	86,67	100	100	96,67
“matikan TV”	93,33	96,67	93,33	100	100	90	90	93,33
“nyalakan audio”	100	80	90	100	93,33	90	96,67	83,33
“matikan audio”	96,67	86,67	96,67	93,33	90	100	93,33	90
Rata-rata	90	80	92	90,42	86	90,67	87,33	89,67

3. Hasil dan Pembahasan

Hasil pengujian sistem *Smart home* berbasis IoT ini menunjukkan performa yang baik dalam aspek pengendalian perangkat rumah tangga dan pemantauan konsumsi daya. Subbab-subbab berikut membahas pengujian dan analisis hasil untuk setiap fitur sistem.

3.1. Analisis Fitur Kontrol Berbasis Suara

Evaluasi terhadap fitur kontrol berbasis suara dilakukan dengan menguji respon sistem terhadap perintah suara. Ada 10 frase perintah yang sudah ditetapkan dalam sistem. Sembilan perintah ini diujicoba oleh 8 orang *user* dengan karakter suara yang berbeda, yaitu *user* dengan nada suara rendah, *user* dengan nada suara tinggi, dan *user* dengan nada suara sedang. Pembicara memberi perintah kepada sistem dengan ucapan sebanyak 30 kali untuk masing-masing frase perintah. Tingkat keberhasilan pengujian ini disajikan pada Tabel 1. Data uji tersebut menunjukkan bahwa tingkat keberhasilan pengendalian berbasis suara sudah tinggi dengan rata-rata akurasi mencapai 83,33% hingga 90% untuk berbagai perintah. Akurasi tinggi pada pengendalian perangkat menunjukkan bahwa sistem pengenalan suara bekerja dengan baik untuk sebagian besar perintah, meskipun terdapat kendala pada perintah dengan struktur kalimat lebih kompleks.

Fitur kontrol berbasis suara ini diuji juga dengan skenario gangguan/noise. *Noise* yang diberikan berupa suara gangguan, dimana terdapat orang yang membaca tulisan dengan keras di belakang *user* yang memberi perintah pada sistem dengan suaranya.

Pengujian dengan latar *noise* dilakukan sekali untuk setiap frase karena hanya untuk memastikan kemampuan awal sistem terhadap gangguan. Hasil pengujian dengan latar *noise* tersebut disajikan pada Tabel 2. Pengujian tersebut menunjukkan bahwa 60% perintah masih diterima oleh sistem, 10% perintah diterima setelah diulang 2 kali, 20% perintah diterima dengan frase yang tidak sesuai, dan 10% perintah tidak bisa diterima. Data ini menunjukkan bahwa sistem dapat beroperasi pada lingkungan ber-*noise* moderat.

Tabel 2. Pengujian dengan Latar Noise

Perintah	Status
"nyalakan lampu teras"	Diterima
"matikan lampu teras"	Diterima
"nyalakan lampu ruang tamu"	Diterima, tetapi membutuhkan 2 kali pengucapan
"matikan lampu ruang tamu"	Diterima
"nyalakan kipas"	Diterima
"matikan kipas"	Diterima
"nyalakan audio"	Diterima
"matikan audio"	Diterima, tetapi kalimat salah
"nyalakan TV"	Tidak diterima
"Matikan TV"	Diterima, tetapi kalimat salah

Selain diuji dengan latar *noise*, fitur kendali berbasis suara ini juga diuji dengan variasi jarak. Jarak dekat sebesar 5 - 7 cm, dan jarak jauh sekitar 3 meter. Pengujian variasi jarak ini dimaksudkan untuk melihat respon sistem terhadap perintah suara pada jarak dekat dan pada jarak yang lebih jauh. Keterbatasan utama adalah penurunan akurasi pada jarak jauh (3 meter), sebagaimana ditunjukkan pada Tabel 3.

Tabel 3. Pengujian dengan Variasi Jarak

Perintah	Jarak dekat (5-7 cm)	Jarak jauh (3 m)
"nyalakan lampu teras"	Diterima	Diterima
"matikan lampu teras"	Diterima	Tidak Diterima
"nyalakan lampu ruang tamu"	Diterima	Diterima, tetapi kalimat salah
"matikan lampu ruang tamu"	Diterima	Tidak diterima
"nyalakan kipas"	Diterima	Diterima
"matikan kipas"	Diterima	Diterima
"nyalakan audio"	Diterima	Diterima, tetapi kalimat salah
"matikan audio"	Diterima	Tidak diterima
"nyalakan TV"	Diterima	Diterima, tetapi kalimat salah
"Matikan TV"	Diterima	Diterima

Tentu saja parameter jarak bukan satu-satunya parameter yang menyebabkan perubahan respon sistem. Permasalahan jarak sangat dipengaruhi oleh kualitas *micrpphone* dari *smartphone*. Untuk mengatasi masalah ini, penggunaan mikrofon eksternal dengan sensitivitas lebih tinggi dapat dipertimbangkan.

3.2. Analisis Fitur EcoMode

Fitur ini merupakan fitur untuk menyala-matikan lampu secara terjadwal sesuai dengan waktu yang diinginkan. Fitur ini dibantu oleh *Network Time Protocol* (NTP) yang terhubung pada mikrokontroler.

Pada pengujian ini diset lampu akan menyala pada puku 18.00 dan akan mati pada pukul 7.00. Zona waktu ujicoba adalah Waktu Indonesia bagian Barat (WIB). Hasil pengujian ada delay 2 detik untuk penyalaan dan 4 detik untuk pematian lampu akibat delay komunikasi pada

jaringan WiFi. Tabel uji Fitur EcoMode disajikan pada Tabel 4. Posisi perangkat yang jauh dari pemancar Wi-Fi menyebabkan latensi pada pengendalian perangkat, yang dapat diatasi dengan meningkatkan jangkauan jaringan WiFi atau menambahkan penguat sinyal.

Tabel 4. Pengujian Fitur EcoMode

Status	Jam seharusnya	Jam sebenarnya	Selisih waktu
Lampu menyala	18:00	18:00:02	2 detik
Lampu mati	7:00	7:00:04	4 detik

Efisiensi akibat adanya fitur ini terjadi akibat adanya jadwal mematikan lampu. Jika tidak terjadwal secara otomatis, seringkali pemilik rumah terlambat mematikan lampu. Efisiensi dihitung dari selisih waktu menyala. Misalnya terdapat 6 buah lampu dengan daya masing-masing sebesar 9 watt, selisih waktu menyala selama 1 jam, maka selisih konsumsi energi oleh lampu sebesar

$$E = P \times t \quad (1)$$

$$= 6 \text{ (lampu)} \times 9 \text{ (watt)} \times 1 \text{ (jam)} = 0,054 \text{ kWh.}$$

Jika tarif listrik rumah tangga R1 sekitar Rp 1.444,7 per kWh, maka biaya tambahan sebesar $0,054 \times 1.444,7 \approx$ Rp 78 per hari. Nilai ini akan terasa besar untuk jangka waktu yang lama.

3.3. Analisis Fitur Pemantauan Konsumsi Daya Listrik

Pengujian fitur konsumsi daya listrik tidak secara langsung mengukur nilai daya, tetapi dengan mengukur nilai arus yang tercatat karena sensor yang digunakan adalah sensor arus. Daya P diperoleh dari perkalian arus dengan tegangan konstan 220 V. Tabel 5 menunjukkan pebandingan arus yang tercatat sensor (I_a) dengan arus referensi I_m yang diperoleh dengan menggunakan multimeter. Selisih pembacaan arus oleh sensor dan multimeter merupakan error pembacaan arus (ΔI). Persentasi *error* pembacaan diperoleh dengan

$$\varepsilon = \frac{\Delta I}{I_m} \times 100\%. \quad (1)$$

Pengujian dilakukan sebanyak 18 kali dengan rata-rata persentasi *error* yang relatif kecil sebesar 1,93%. Selanjutnya nilai arus yang dibaca akan dikalikan dengan tegangan konstan 220 V untuk memperoleh nilai daya yang dikonsumsi. Daya dan arus yang terbaca akan ditampilkan pada *progress bar*. Sensor SCT-013 memberikan performa yang memadai untuk pemantauan konsumsi daya listrik, meskipun terdapat kesalahan yang relatif kecil. Kesalahan ini dapat diminimalkan dengan kalibrasi ulang pada arus tinggi. Ketidakstabilan pada NodeMCU pengolah data arus disebabkan oleh beban kerja yang tinggi dalam waktu lama. Namun demikian, Tabel 5 menunjukkan tingkat kesalahan pengukuran arus (ε) berkisar antara 0% hingga

2,11% menunjukkan akurasi yang cukup tinggi. Solusi yang diusulkan adalah menggunakan NodeMCU tambahan untuk membagi beban kerja atau mengganti modul dengan perangkat keras yang lebih andal. Solusi lainnya adalah penggunaan sensor tegangan tambahan untuk meningkatkan akurasi perhitungan daya, yang dapat dipertimbangkan untuk pengembangan sistem ini.

3.4. Analisis Fitur Smart Remote

Pengujian fitur *smart remote* dilakukan dengan cara menguji halaman *smart remote* pada aplikasi berbasis *smartphone*. Pengujian dilakukan dengan *metode balck box testing*. Pengujian dilakukan sebanyak 10 kali untuk masing-masing menu yaitu:

1. menyalakan dan mematikan kipas angin
2. menyalakan dan mematikan TV
3. menyalakan dan mematikan lampu teras
4. menyalakan dan mematikan lampu ruang tamu
5. menyalakan dan mematikan *audio/sound system*

Semua pengujian dinyatakan berhasil, dimana sistem merespon sesuai input yang diberikan. Hal ini menunjukan bahwa semua alat dapat dikendalikan melalui aplikasi melalui fitur *smart remote*.

4. Kesimpulan

Penelitian ini berhasil mengembangkan sistem *Smart Home* berbasis IoT yang mampu mengendalikan perangkat rumah tangga melalui aplikasi Android dan perintah suara. Sistem ini juga dilengkapi fitur pemantauan konsumsi daya secara *real-time*, yang memberikan manfaat dalam pengelolaan energi rumah tangga. Hasil pengujian menunjukkan bahwa sistem memiliki akurasi yang baik dalam pengendalian perangkat dan pemantauan daya, meskipun masih terdapat kendala pada stabilitas perangkat keras dan latensi jaringan. Sistem ini memiliki potensi untuk diimplementasikan secara luas dan memberikan solusi praktis bagi kebutuhan rumah tangga modern yang berbasis IoT. Ke depannya, penelitian ini dapat dikembangkan lebih lanjut dengan meningkatkan stabilitas perangkat keras, optimasi jaringan, dan integrasi dengan sistem pemantauan keamanan lingkungan untuk mendukung kinerja yang lebih baik dalam berbagai kondisi lingkungan.

Ucapan Terima Kasih

Penulis mengucapkan terimakasih kepada pimpinan Pusat Penelitian dan Publikasi UIN Sunan Gunung Djati Bandung atas dukungan dana untuk publikasi melalui Skema Pengembangan Kapasitas. Penulis juga mengucapkan terimakasih kepada pimpinan Pusat Penelitian Mekatronika Cerdas, Badan Riset dan Inovasi Nasional (BRIN) atas kerjasama penelitian yang dilakukan.

Referensi

- [1]. S. Nizetić, P. Šolić, D. López-de-Ipiña González-de-Artaza, and L. Patrono, "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future," *J Clean Prod*, vol. 274, p. 122877, 2020, doi: <https://doi.org/10.1016/j.jclepro.2020.122877>.
- [2]. M. Nadil Khan, Tanvirahmedshuvo, M. R. Hossain Ontor, N. Khan, and A. Rahman, "The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises," *International Journal For Multidisciplinary Research*, vol. 6, no. 1, Feb. 2024, doi: 10.36948/ijfmr.2024.v06i01.22699.
- [3]. M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *Journal of network and computer applications*, vol. 97, pp. 48–65, 2017.
- [4]. A. K. Gupta and R. Johari, "IOT based Electrical Device Surveillance and Control System," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2019, pp. 1–5. doi: 10.1109/IoT-SIU.2019.8777342.
- [5]. N. R. Evandi, H. Fakhurroja, and N. Ismail, "Smartphone-based application to control TV using IoT technology," in *AIP Conference Proceedings*, 2023.
- [6]. S. Yıldız and M. Burunkaya, "Web Based Smart Meter for General Purpose Smart Home Systems with ESP8266," in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Oct. 2019, pp. 1–6. doi: 10.1109/ISMSIT.2019.8932931.
- [7]. T. S. Gunawan *et al.*, "Prototype design of smart home system using internet of things," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 7, no. 1, pp. 107–115, Jul. 2017, doi: 10.11591/ijeecs.v7.i1.pp107-115.
- [8]. A. Sharma, A. Gautam, A. Singh, and S. Chaurasia, "Smart Homes of the Future: A Comprehensive Review of IoT-Based Home Automation," vol. 8, pp. 142–147, May 2023.
- [9]. R. Rizal and I. Karyana, "Sistem Kendali dan Monitoring pada Smart Home Berbasis Internet of Things (IoT)," *Innovatics*, vol. 1, no. 2, pp. 43–50, 2019.
- [10]. X. Zhang, D. Luo, and X. Wu, "Based on Raspberry Pi Voice Controlled Smart Home System," *Frontiers in Computing and Intelligent Systems*, vol. 8, no. 2, pp. 38–42, 2024.
- [11]. T. Malche and P. Maheshwary, "Internet of Things (IoT) for building smart home system," in *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, Institute of Electrical and Electronics Engineers Inc., Oct. 2017, pp. 65–70. doi: 10.1109/I-SMAC.2017.8058258.
- [12]. Z. H. Rihhadatulaisy and K. D. Irianto, "Designing an Automatic Room Temperature Control System for Smart Homes for the Elderly Using IoT," *International Journal Software Engineering and Computer Science (IJSECS)*, vol. 4, no. 2, pp. 758–766, 2024.
- [13]. H. Fakhurroja, C. Machbub, A. S. Prihatmanto, and A. Purwarianti, "Multimodal interaction system for home appliances control," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 15, pp. 44–67, 2020, doi: 10.3991/IJIM.V14I15.13563.

- [14]. A. Sharma, S. Chaurasia, A. Gautam, and A. Singh, "Smart Homes of the Future: A Comprehensive Review of IoT-Based Home Automation," *International Journal of Novel Research and Development (IJNRD)*, vol. 8, no. 5, pp. 142–147, 2023, [Online]. Available: <https://www.researchgate.net/publication/371666903>.
- [15]. M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C. H. Lung, "Smart home: Integrating internet of things with web services and cloud computing," in *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, IEEE Computer Society, 2013, pp. 317–320. doi: 10.1109/CloudCom.2013.155.
- [16]. Prasetyo. Erwan Eko, "Aplikasi Internet of Things (IoT) Untuk Pemantauan dan Pengendalian Beban Listrik di Ruangan," *Jurnal Teknika STTKD*, vol. 4, no. 2, pp. 28–39, 2017.
- [17]. A. Rahayu and Hendri, "Sistem Kendali Rumah Pintar Menggunakan Voice Recognition Module V3 Berbasis Mikrokontroler dan IoT," *Jurnal Teknik Elektro dan Vokasional (JTEV)*, vol. 6, no. 2, p. 19, 2020, [Online]. Available: <http://ejournal.unp.ac.id/index.php/jtev/index>.
- [18]. P. O. Ayeni and O. C. Adesoba, "IoT-based home control system using NodeMCU and Firebase," *Journal of Edge Computing*, Nov. 2024, doi: 10.55056/jec.814.
- [19]. C. Selvarathi, P. Santhi, K. Deepa, G. R. Shri, C. Visali, and T. Abirami, "Transforming Homes into Smart Havens: Advanced Control with NodeMCU and Android Technology," in *2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, 2024, pp. 1–5.
- [20]. J. Zhao, L. Sun, and H. Fan, "MQTT-based internet of things smart home linkage control system," in *2024 3rd International Conference on Electronics and Information Technology (EIT)*, 2024, pp. 521–525.
- [21]. H. A. Rochman, R. Primananda, and H. Nurwasito, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," *Jurnal Pengembangan Teknologi Informasi dan Komputer*, vol. 1, no. 6, pp. 445–455, 2017, [Online]. Available: <http://j-ptiik.ub.ac.id>.
- [22]. A. Khakimov, A. Muthanna, R. Kirichek, A. Koucheryavy, and M. S. A. Muthanna, "Investigation of methods for remote control IoT-devices based on cloud platforms and different interaction protocols," in *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus)*, Feb. 2017, pp. 160–163. doi: 10.1109/EIconRus.2017.7910518.