

IMPLEMENTASI MIKROKONTROLER ESP32 UNTUK KONTROL BERBASIS PENGENALAN SUARA DAN CAHAYA DALAM SISTEM SMART HOME

IMPLEMENTATION OF ESP32 MICROCONTROLLER FOR VOICE AND LIGHT RECOGNITION-BASED CONTROL IN SMART HOME SYSTEM

Berliyan Sri Surani¹, Alexander Dharmawan², Cristeddy Asa Bakti³

Universitas AKI Semarang^{1,2,3}

223210019@student.unaki.ac.id¹

ABSTRACT

This study discusses the implementation of the ESP32 microcontroller in a Smart Home system to control lighting and fan devices based on voice recognition and light sensor input. The system is designed to automatically manage lighting using LDR sensors to detect ambient light intensity and respond to voice commands via Google Assistant integrated with IFTTT and Adafruit IO. The methods include hardware design, microcontroller programming using the Arduino IDE, cloud service integration, and system testing on a miniature house model. Test results show that the system responds to voice commands with an average delay of 1.9 seconds and accurately reads light intensity every 2 seconds. The system also features a manual override to prevent automatic activation if a device has been manually turned off. Therefore, the system successfully combines automatic and manual control in a flexible Smart Home ecosystem based on the Internet of Things (IoT).

Keywords: Smart Home, ESP32, Google Assistant, LDR Sensor, Internet of Things, IFTTT, Adafruit IO.

ABSTRAK

Penelitian ini membahas implementasi mikrokontroler ESP32 dalam sistem *Smart Home* untuk kontrol perangkat lampu dan kipas berbasis pengenalan suara dan sensor cahaya. Sistem dirancang agar dapat mengatur pencahayaan secara otomatis berdasarkan intensitas cahaya lingkungan menggunakan sensor LDR, serta menerima perintah suara melalui Google Assistant yang terhubung dengan IFTTT dan Adafruit IO. Metode yang digunakan meliputi perancangan perangkat keras, pemrograman mikrokontroler dengan Arduino IDE, integrasi layanan *cloud*, serta pengujian sistem pada miniatur rumah. Hasil pengujian menunjukkan bahwa sistem merespons perintah suara dengan rata-rata waktu 1,9 detik dan mampu membaca intensitas cahaya secara akurat setiap 2 detik. Sistem juga mendukung fitur *override* manual untuk mencegah penyalakan otomatis jika perangkat telah dimatikan oleh pengguna. Dengan demikian, sistem berhasil menggabungkan kontrol otomatis dan manual secara fleksibel dalam satu ekosistem *Smart Home* berbasis *Internet of Things* (IoT).

Kata Kunci: Smart Home, ESP32, Google Assistant, Sensor LDR, Internet of Things, IFTTT, Adafruit IO

PENDAHULUAN

Kemajuan teknologi informasi dan komunikasi yang sangat pesat, telah mentransformasi berbagai aspek kehidupan manusia secara signifikan. Salah satu perkembangan yang signifikan adalah munculnya konsep *Internet of Things* (IoT), yang memungkinkan interkoneksi dan pertukaran data antar perangkat pintar. Perkembangan pesat teknologi *Internet of Things* (IoT) ini telah membawa perubahan yang signifikan dalam kehidupan sehari-hari, terutama dalam ranah pemeliharaan rumah pintar (*Smart Home*) (Visayas et al., 2024).

Smart Home, sebagai sebuah konsep yang terus dikembangkan, bertujuan untuk

meningkatkan kenyamanan dan efisiensi kehidupan manusia melalui otomatisasi berbagai perangkat rumah tangga yang dapat dikendalikan dari jarak jauh melalui jaringan internet. Secara umum, teknologi *Smart Home* merupakan integrasi jaringan listrik dengan berbagai peralatan rumah tangga seperti lampu, AC, pintu, dan lain-lain (Prasetyo et al., 2022). Tujuan utama dari implementasi *Smart Home* adalah menciptakan lingkungan tempat tinggal dengan sistem otomasi elektronik yang seluas mungkin di dalam rumah sehingga perlengkapan ini dapat dikontrol dan dimonitor (Isnanto et al., 2021). Salah satu implementasi otomatisasi yang banyak dikembangkan adalah sistem kontrol

pencahayaan dan sirkulasi udara, yang berpotensi besar untuk meningkatkan efisiensi energi dan kenyamanan pengguna (Alvin R. Kedoh et al., 2019).

Namun, sistem kontrol lampu dan kipas yang umum banyak digunakan pada rumah di Indonesia saat ini masih mengandalkan kontrol manual, baik melalui saklar fisik konvensional maupun aplikasi sederhana. Ketergantungan pada metode manual ini seringkali menyebabkan pemborosan energi akibat perangkat sering kali dibiarkan menyala tanpa pengawasan, serta meningkatkan potensi terjadinya kesalahan pengguna seperti lupa mematikan alat elektronik. Hal ini masih menjadi tantangan tersendiri untuk mengatur sistem kendali lampu rumah karena praktik manual ini juga berpotensi menimbulkan risiko korsleting listrik akibat kelalaian pengguna (Mariam et al., 2024). Oleh karena itu, pengembangan sistem kontrol lampu dan kipas yang lebih cerdas dan otomatis menjadi krusial. Sistem kontrol lampu berbasis pengenalan suara dan sensor cahaya menjajikan solusi yang tidak hanya meningkatkan efisiensi energi secara signifikan tetapi juga memberikan pengalaman pengguna yang lebih sederhana dan modern. Di sisi lain, kipas angin, sebagai perangkat pendukung sirkulasi udara dalam ruangan, dapat dikendalikan secara praktis melalui *web dashboard* berbasis jaringan yang terintegrasi dengan sistem IoT. Dengan begitu, pengguna dapat menghidupkan atau mematikan kipas dari jarak jauh tanpa harus melakukan kontrol manual secara langsung (Swathi et al., 2023).

Dalam mewujudkan sistem *Smart Home* yang cerdas, teknologi mikrokontroler telah memegang peranan penting dalam arsitektur sistem tersebut. Mikrokontroler merupakan sebuah perangkat elektronik berbentuk chip yang terintegrasi dalam satu IC (*integrated circuit*), lengkap dengan prosesor, memori, dan antarmuka *input/output* (I/O), semuanya terintegrasi dalam satu paket yang dapat diprogram. Mikrokontroler

berfungsi sebagai bagian yang mengatur dan memberikan perintah ke bagian-bagian lain agar alat tersebut bisa bekerja secara otomatis (Cakra et al., 2023). Di antara berbagai jenis mikrokontroler, ESP32 menjadi pilihan yang populer dalam pengembangan sistem IoT, karena kemampuan konektivitas nirkabelnya yang terintegrasi, seperti Wi-Fi dan Bluetooth. Dengan kapasitas komputasi yang cukup, hal ini menjadikannya cocok untuk diterapkan dalam sistem *Smart Home* (Is'ad et al., 2024). Keunggulan ini memungkinkan ESP32 untuk mebangun sistem yang fleksibel, terhubung ke jaringan, dan dapat dikendalikan dari jarak jauh tanpa memerlukan modul tambahan (Nizam et al., 2022). Selain itu, ESP32 memiliki prosesor *dual-core*, monitoring lingkungan, dan sistem kendali berbasis suara (Febriansyah et al., 2024). Dalam konteks pencahayaan, kombinasi sensor cahaya untuk intesitas lampu berdasarkan kondisi lingkungan (Martins, 2023), dan teknologi pengenalan suara untuk memungkinkan kontrol verbal (Jayusman et al., 2020) menawarkan potensi otomatisasi dan efisiensi yang signifikan. Sedangkan penggunaan antarmuka *web* menjadi solusi sederhana namun efektif untuk mengontrol perangkat lain seperti kipas, yang hanya membutuhkan fungsi *on/off* dari jarak jauh (Anwar et al., 2020).

Meskipun teknologi-teknologi pendukung ini sudah tersedia, potensi pasar *Smart Home* di Indonesia sebenarnya cukup signifikan, dengan adopsi yang dilaporkan mencapai 835 juta rumah (Khoirunnisa, 2023). Selain itu, pandangan akademis juga menekankan pentingnya integrasi IoT dan CPS dalam konsep *Smart Home* di era industri 4.0 (FTMM UNAIR, 2022). Namun demikian, adopsi sistem *Smart Home* berbasis IoT di Indonesia masih tergolong rendah (Saputra et al., 2024). Terlepas dari potensi ini, aspek koneksi dan interoperabilitas sebagai tantangan utama dalam *Smart Home* berbasis IoT, dimana protokol dapat berkomunikasi dan kemampuan perangkat untuk saling

berinteraksi (Laboy et al., 2024). Beberapa sistem yang ada masih memiliki keterbatasan dalam hal kecepatan respons, keakuratan pengenalan suara, serta kestabilan sistem dalam kondisi jaringan yang tidak selalu optimal (Iwasawa et al., 2025).

Beberapa faktor menjadi penyebab utama rendahnya implementasi sistem *Smart Home* tersebut : pertama, keterbatasan pengetahuan dan keterampilan dalam pemrograman mikrokontroler, terutama bagi pengguna awam, menjadi hambatan signifikan dalam merancang dan mengembangkan sistem kustom. Kedua, perangkat *Smart Home* yang sudah tersedia di pasaran sering kali memiliki harga yang relatif tinggi, menjadikannya kurang terjangkau bagi sebagian besar masyarakat Indonesia. Ketiga, kurangnya inisiatif penelitian dan pengembangan (R&D) di tingkat lokal dalam konteks aplikasi *Smart Home* yang spesifik untuk kebutuhan dan kondisi di Indonesia menyebabkan keterbatasan dalam inovasi dan implementasi solusi IoT yang relevan dengan lingkungan sekitar (Putra Lesmana et al., 2024). Sebagai contoh, implementasi sistem pengenalan suara mungkin menghadapi tantangan terkait variasi dialek dan aksen lokal yang belum tentu didukung oleh *platform* pengenalan suara global.

Penelitian ini memiliki urgensi untuk mengembangkan solusi kontrol pencahayaan berbasis IoT yang lebih terjangkau, mudah diimplementasikan, dan efisien secara energi, dengan mempertimbangkan konteks penggunaan di Indonesia. Dengan memanfaatkan mikrokontroler ESP32 sebagai mikrokontroler utama, serta mengintegrasikan pengenalan suara dan sensor cahaya yang dioptimalkan, diharapkan dapat tercipta sebuah sistem yang tidak hanya meningkatkan efisiensi energi rumah tangga tetapi juga memberikan pengalaman pengguna yang lebih baik dan sederhana. Penelitian ini juga bertujuan untuk memberikan kontribusi pada pengembangan teknologi *Smart Home*

di tingkat lokal, menyediakan referensi bagi inovasi selanjutnya, dan membuka peluang adopsi IoT yang lebih luas dalam kehidupan sehari-hari masyarakat Indonesia. Berbeda dengan penelitian sebelumnya yang mungkin fokus pada platform mikrokontroler lain atau algoritma kontrol yang lebih kompleks, penelitian ini akan mengeksplorasi konfigurasi dan optimasi spesifik ESP32 dengan sensor cahaya dan pengenalan suara untuk mencapai keseimbangan antara biaya, kinerja, dan kemudahan penggunaan.

Dengan demikian, implementasi IoT dalam konteks *Smart Home* di Indonesia tidak hanya berpotensi mengatasi masalah efisiensi energi dan pemeliharaan, tetapi juga dapat berkontribusi pada peningkatan kualitas hidup dan adopsi teknologi secara lebih merata. Pergeseran gaya hidup menuju digitalisasi dan otomatisasi menciptakan peluang besar bagi teknologi ini untuk berkembang dan memberikan manfaat nyata bagi masyarakat luas. Dalam konteks ini, pengembangan teknologi *Smart Home* berbasis IoT yang disesuaikan dengan kebutuhan dan kondisi di Indonesia memiliki potensi signifikan untuk meningkatkan standar hidup dan kesejahteraan masyarakat (Manuhutu et al., 2025).

Berdasarkan pemasalahan yang telah diuraikan, penelitian ini berfokus pada “Implementasi Mikrokontroler ESP32 untuk Kontrol Berbasis Pengenalan Suara dan Cahaya dalam Sistem *Smart Home*”. Tujuan utama dari penelitian ini adalah untuk merancang dan mengimplementasikan sistem yang dapat mengotomatisasi pencahayaan rumah berdasarkan kondisi lingkungan, perintah suara pengguna dan kontrol sirkulasi udara. Selain itu, penelitian ini juga bertujuan untuk menguji dan mengevaluasi performa yang dikembangkan dalam hal akurasi deteksi suara, responsifitas sensor cahaya terhadap perubahan intensitas cahaya yang dievaluasi melalui pembacaan nilai ADC dari sensor LDR pada Serial Monitor Arduino IDE, serta potensi efisiensi energi

yang dapat dicapai melalui sistem otomatisasi ini.

METODE

Penelitian ini menggunakan metode eksperimen fungsional untuk menguji kinerja sistem *Smart Home* dalam merespons *input* otomatisasi melalui perintah suara, sensor cahaya (LDR), dan kontrol manual via *dashboard web*. Pengujian dilakukan menggunakan mikrokontroler ESP32 sebagai pusat kendali. Metode ini digunakan untuk menilai efektivitas dan stabilitas sistem, serta kemudahan pengguna dalam mengontrol perangkat secara otomatis dan manual, tanpa menguji hubungan antar variabel seperti dalam penelitian kuantitatif.

Metode Pengumpulan Data

Penelitian ini menggunakan tiga metode pengumpulan data, yaitu: Observasi, untuk menilai langsung fungsionalitas sistem *Smart Home* melalui pengamatan respons sistem terhadap kontrol suara, sensor cahaya (LDR), dan *dashboard web* dalam kondisi nyata, Dokumentasi, untuk mengumpulkan data sekunder berupa diagram, kode pemrograman, foto, dan hasil pengujian sistem guna mendukung keabsahan observasi, dan Studi Pustaka, untuk mengkaji literatur terkait konsep *Smart Home*, IoT, serta teknologi kontrol berbasis suara dan sensor sebagai dasar perancangan sistem.

Sumber Data

Sumber data dalam penelitian ini terdiri dari data primer dan data sekunder. Data primer diperoleh langsung dari hasil observasi dan pengujian sistem *Smart Home* oleh peneliti, mencakup respons sensor cahaya, perintah suara, dan kontrol manual melalui *dashboard web*. Data sekunder berasal dari buku, jurnal, artikel, serta datasheet resmi komponen seperti ESP32, LDR, dan relay, yang digunakan

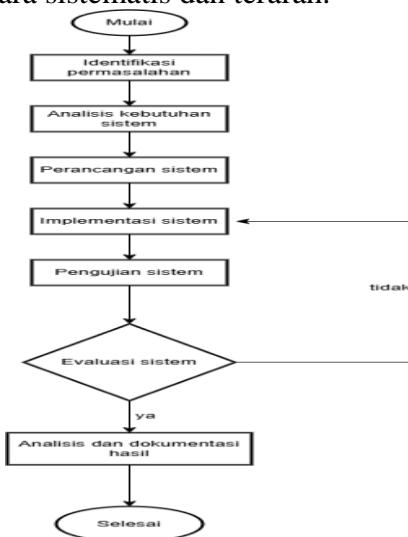
sebagai acuan dalam perancangan dan integrasi sistem.

Jenis Data

Jenis data dalam penelitian ini terdiri dari data kuantitatif dan kualitatif. Data kuantitatif berupa angka yang diperoleh dari hasil pengujian sistem, seperti pembacaan sensor cahaya (LDR), jumlah perintah suara yang berhasil dikenali, status ON/OFF perangkat, dan waktu respon sistem. Data ini dianalisis secara statistik untuk mengevaluasi performa sistem secara objektif. Sementara itu, data kualitatif bersifat deskriptif dan diperoleh dari observasi langsung, dokumentasi visual, serta catatan naratif selama pengujian. Data ini digunakan untuk menggambarkan karakteristik sistem, kenyamanan penggunaan, dan konsistensi fungsionalitas.

Rancangan Penelitian

Dalam penelitian, rancangan penelitian dapat diartikan sebagai kerangka atau struktur konseptual yang menjadi pedoman agar proses penelitian berjalan secara sistematis dan terarah.



Gambar 1. Rancangan Penelitian

Diagram alur rancangan penelitian ini mencerminkan langkah sistematis dalam mengembangkan sistem *Smart Home* berbasis IoT menggunakan mikrokontroler ESP32. Penelitian diawali dengan menetapkan ide dasar, diikuti identifikasi masalah terkait kontrol manual perangkat

rumah tangga. Selanjutnya, dilakukan analisis kebutuhan sistem untuk merinci fungsi kontrol otomatis dan manual, lalu perancangan teknis seperti *flowchart* dan *wiring diagram*. Setelah sistem diimplementasikan dengan merakit dan memprogram perangkat, dilakukan pengujian untuk mengevaluasi kecepatan respons, akurasi, dan stabilitas. Evaluasi sistem dilakukan untuk menilai kesesuaian hasil dengan tujuan, dilanjutkan dokumentasi menyeluruh sebagai bukti dan referensi pengembangan lanjutan. Proses diakhiri dengan penyusunan laporan sebagai hasil akhir dari penelitian ini.

HASIL DAN PEMBAHASAN

Implementasi Perangkat Keras

Tahap implementasi perangkat keras dilakukan melalui dua fase, yaitu perakitan terbuka dan pemasangan pada miniatur rumah sebagai media visualisasi. Pada tahap awal, seluruh komponen seperti ESP32, sensor cahaya LDR, dan modul relay 4 channel dirakit secara terbuka untuk memudahkan pengujian koneksi dan fungsionalitas. ESP32 berfungsi sebagai pusat kendali, sementara relay mengatur ON/OFF tiga lampu dan satu kipas. Setelah seluruh komponen diuji dan dipastikan terintegrasi dengan baik, sistem kemudian dipasang pada miniatur rumah untuk mensimulasikan lingkungan *Smart Home* secara visual.

Tabel 1. Rangkaian Perkabelan Sensor Cahaya (LDR) dengan ESP32

| Pin | Terhubung ke pin |
|-----|------------------|
| LDR | ESP32 |
| DO | GPIO 36 |
| VCC | 5V |
| GND | GND |

Tabel 2. Rangkaian Perkabelan Relay dengan ESP32

| Pin Relay | Terhubung ke pin ESP32 |
|-------------------|------------------------|
| IN1 (Kipas) | GPIO 14 |
| IN2 (Lampu Kamar) | GPIO 27 |

| | |
|------------------------|---------|
| IN3 (Lampu Ruang Tamu) | GPIO 26 |
| IN4 (Lampu Teras) | GPIO 25 |
| VCC | 5V |
| GND | GND |

Tabel 3. Rangkaian Perkabelan Perangkat Output dengan Relay

| Perangkat | Terhubung ke Relay |
|------------------|-----------------------------|
| Kipas | Output Relay Channel 1 (NO) |
| Lampu Kamar | Output Relay Channel 2 (NO) |
| Lampu Ruang Tamu | Output Relay Channel 3 (NO) |
| Lampu Teras | Output Relay Channel 4 (NO) |



Gambar 2. Perakitan Awal Sistem secara Terbuka



Gambar 3. Tampilan Bagian Samping Kanan Miniatur



Gambar 4. Tampilan Miniatur Rumah dari Berbagai Sisi



Gambar 5. Tampilan Bagian Samping Kiri Miniatur

Setelah semua komponen terpasang dan terintegrasi, dilakukan pengujian menyeluruh untuk memastikan sistem dapat bekerja secara stabil dan sesuai dengan logika kontrol yang telah dirancang. Dengan pendekatan berbasis miniatur ini, sistem *Smart Home* dapat divisualisasikan secara edukatif, praktis, dan representatif terhadap implementasi nyata di lingkungan rumah tinggal.

Implementasi Perangkat Lunak

Perangkat lunak berperan sebagai pengendali utama yang mengatur interaksi antara perangkat keras, layanan *cloud*, serta *input* dari pengguna. Perangkat lunak dirancang agar sistem dapat merespons secara otomatis berdasarkan intensitas cahaya dari sensor LDR, perintah suara menggunakan Google Assistant, serta secara manual melalui kontrol *web dashboard*.

Perangkat lunak dalam sistem ini tidak hanya terbatas pada program utama yang dijalankan oleh mikrokontroler, melainkan juga mencakup integrasi dengan *platform cloud* seperti Adafruit IO dan layanan otomasi IFTTT, serta pemanfaatan Google Assistant sebagai antarmuka perintah suara. Selain itu, sistem juga terhubung dengan *web dashboard*. Seluruh perangkat lunak ini diintegrasikan agar bekerja secara sinkron dalam menjalankan fungsi kendali perangkat secara cerdas dan fleksibel sesuai kebutuhan pengguna.

Arduino IDE dan Program Utama ESP32

Arduino IDE digunakan sebagai *platform* pemrograman untuk ESP32 dalam sistem *Smart Home* ini, menggunakan bahasa C++ dan berbagai pustaka seperti

WiFi.h, Adafruit_MQTT.h, ArduinoWebsockets.h, dan ArduinoJson.h untuk mendukung koneksi jaringan, komunikasi MQTT dengan Adafruit IO, WebSocket *real-time* dengan *dashboard*, serta pengolahan data JSON. Program ESP32 dibagi menjadi dua bagian utama: setup() untuk inisialisasi pin, koneksi WiFi, MQTT, dan WebSocket; serta loop() yang berjalan terus-menerus untuk membaca *input* (sensor cahaya, perintah suara, dan *dashboard*), menjalankan logika kontrol otomatis/manual, dan mengirim status perangkat ke server *backend* dalam format JSON.

Integrasi Google Assistant dan Adafruit IO pada IFTTT

Integrasi perintah suara dalam sistem ini memanfaatkan *platform* IFTTT untuk menghubungkan Google Assistant dengan Adafruit IO, sehingga pengguna dapat mengontrol lampu melalui perintah suara tanpa perlu menulis kode tambahan. IFTTT bekerja dengan konsep Trigger (perintah dari Google Assistant) dan Action (pengiriman data ke Adafruit IO). Misalnya, saat pengguna mengatakan “Hidupkan lampu kamar,” IFTTT akan mengirim nilai 1 ke feed *lampa_kamar* di Adafruit IO yang telah di-subscribe oleh ESP32, sehingga perintah segera dieksekusi. Konfigurasi *Applet* dilakukan dengan membuat delapan *Applet* terpisah untuk menangani fungsi hidup/mati pada empat perangkat berbeda, melalui tahapan pembuatan perintah di Google Assistant dan pengiriman data ke *feed* yang sesuai di Adafruit IO.



Gambar 6. Konfigurasi Applet IFTTT untuk Perintah “Hidupkan lampu kamar”

Pengujian Sistem

Pengujian sistem dilakukan untuk menilai apakah sistem *Smart Home* telah berfungsi sesuai kebutuhan dan spesifikasi yang dirancang, dengan pendekatan eksperimen melalui pemberian *input* berupa suara, cahaya, dan kontrol manual. Fokus pengujian mencakup kendali suara via Google Assistant, pembacaan cahaya dengan sensor LDR, kontrol melalui *web dashboard*, serta respons keseluruhan sistem terhadap berbagai skenario. Pengujian dilakukan pada miniatur rumah sebagai simulasi lingkungan nyata, menggunakan metode *black-box testing* untuk mengevaluasi *output* berdasarkan *input* tanpa meninjau kode program. Hasil pengujian digunakan untuk menilai keberhasilan integrasi *hardware* dan *software* secara menyeluruh.

Pengujian ini bertujuan untuk mengevaluasi kemampuan sistem dalam menerima dan mengeksekusi perintah suara melalui Google Assistant. Perintah yang diucapkan pengguna diproses oleh platform IFTTT dan diteruskan ke *feed* Adafruit IO,

yang telah *di-subscribe* oleh mikrokontroler ESP32. Selanjutnya, ESP32 mengeksekusi perintah tersebut untuk menyalakan atau mematikan perangkat sesuai dengan instruksi pengguna.



Gambar 7. Menyalakan Lampu Kamar Menggunakan Perintah Suara



Gambar 8. Mematikan Lampu Kamar Menggunakan Perintah Suara



Gambar 9. Menyalakan dan Mematikan dengan Perintah Suara

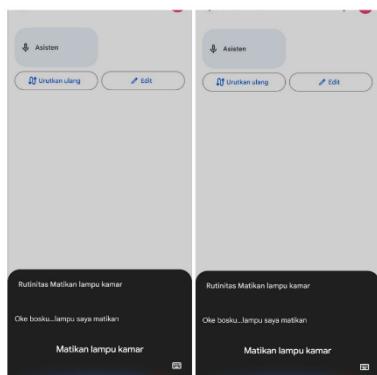
Tabel 4. Hasil Pengujian Pengenalan Suara melalui Google Assistant

| No | Perintah Suara | Feed Adafruit | Data Terkirim | Status Perangkat | Waktu Respon | Hasil |
|-----|----------------------------------|------------------|---------------|------------------|--------------|----------|
| 1. | Hidupkan lampu kamar | lampa_kamar | 1 | Kamar ON | 1,9 detik | Berhasil |
| 2. | Matikan lampu kamar | lampa_kamar | 0 | Kamar OFF | 2,2 detik | Berhasil |
| 3. | Hidupkan lampu ruang tamu | lampa_ruang_tamu | 1 | Ruang Tamu ON | 1,7 detik | Berhasil |
| 4. | Matikan lampu ruang tamu | lampa_ruang_tamu | 0 | Ruang Tamu OFF | 2 detik | Berhasil |
| 5. | Hidupkan lampu teras | lampu_teras | 1 | Teras ON | 2 detik | Berhasil |
| 6. | Matikan lampu teras | lampu_teras | 0 | Teras OFF | 2,1 detik | Berhasil |
| 7. | Hidupkan semua lampu | semua_lampu | 1 | Semua ON | 1,4 detik | Berhasil |
| 8. | Matikan semua lampu | semua_lampu | 0 | Semua OFF | 1,9 detik | Berhasil |
| 9. | Nyalakan semua lampu | - | - | Tidak merespon | - | Gagal |
| 10. | Hidupkan semua lampu (jarak 3 m) | semua_lampu | 1 | Semua ON | 1,9 detik | Berhasil |
| 11. | Nyalakan semua lampu (jarak 7 m) | semua_lampu | 1 | Semua ON | 2 detik | Berhasil |
| 12. | Matikan semua lampu (jarak 1 km) | semua_lampu | 0 | Semua OFF | 2 detik | Berhasil |

| | | | | | | |
|-----|--|-------------|---|----------|-----------|----------|
| 13. | Hidupkan lampu kamar (dengan internet berbeda) | lampu_kamar | 1 | Kamar ON | 1,9 detik | Berhasil |
|-----|--|-------------|---|----------|-----------|----------|

Rata-rata: $23 / 12 = 1,9$ detik

Tingkat keberhasilan : 100%



Gambar 10. Reaksi Google Assistant Ketika Mendeteksi Bawa Lampu Sedang Hidup atau Mati

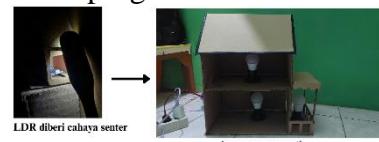
Hasil Pengujian Sensor Cahaya (LDR)

Pengujian sensor cahaya (LDR) dilakukan untuk menilai kemampuan sistem dalam mendeteksi intensitas cahaya dan mengaktifkan atau menonaktifkan

Tabel 5. Hasil Pengujian Respon Otomatis Berdasarkan Nilai LDR

| No | Kondisi Lingkungan | Waktu | Kondisi Cahaya | Nilai ADC | Status Perangkat | Hasil |
|----|---|-------|----------------|-----------|------------------|----------|
| 1. | Di luar ruangan, pagi hari (cahaya cukup) | 08.00 | Terang | 61 | Semua OFF | Berhasil |
| 2. | Di luar ruangan, siang hari (cahaya terik) | 12.30 | Terang | 55 | Semua OFF | Berhasil |
| 3. | Di luar ruangan, sore hari (mulai redup) | 17.30 | Redup | 64 | Semua OFF | Berhasil |
| 4. | Di luar ruangan, malam hari (gelap) | 19.30 | Gelap | 4095 | Semua ON | Berhasil |
| 5. | Di dalam ruangan tanpa pencahayaan buatan | 14.00 | Terang | 68 | Semua OFF | Berhasil |
| 6. | Di dalam ruangan dengan lampu menyala penuh | 18.00 | Terang | 60 | Semua OFF | Berhasil |
| 7. | Di dalam ruangan pencahayaan minim & jauh dari sumber lampu | 20.00 | Gelap | 4095 | Semua ON | Berhasil |
| 8. | Sensor ditutup dengan tangan atau benda | - | Gelap | 4095 | Semua ON | Berhasil |

perangkat secara otomatis berdasarkan nilai ADC yang terbaca. Saat sensor menerima cahaya langsung dan menghasilkan nilai ADC sekitar ± 100 , sistem mengenali kondisi terang dan secara otomatis mematikan seluruh lampu sesuai logika yang telah diprogram.



Gambar 11. Simulasi Pengujian Sensor LDR Dalam Kondisi Terang



Gambar 12. Simulasi Pengujian Sensor LDR Dalam Kondisi Gelap

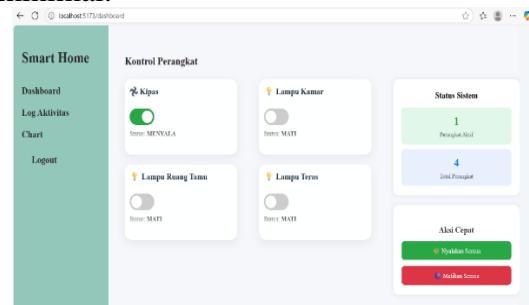
| | | | | | |
|---|---|--------|----|-----------|----------|
| Sensor disinari 9. dengan senter/cahaya buatan langsung | - | Terang | 60 | Semua OFF | Berhasil |
|---|---|--------|----|-----------|----------|

Pengujian menunjukkan bahwa mikrokontroler ESP32 mampu membaca nilai ADC dari sensor LDR secara akurat dan *real-time* setiap 2 detik. Interval ini dipilih untuk menjaga keseimbangan antara responsivitas dan efisiensi sistem, mencegah sensitivitas berlebih terhadap fluktuasi cahaya kecil, serta menjaga stabilitas mikrokontroler agar tidak terbebani oleh pembacaan data secara terus-menerus.

Hasil Pengujian Web Dashboard

Pengujian web dashboard bertujuan mengevaluasi kinerjanya sebagai antarmuka kontrol manual dalam sistem *Smart Home* berbasis ESP32. *Dashboard* ini dibangun dengan React dan terhubung ke ESP32 melalui protokol WebSocket untuk komunikasi dua arah secara *real-time*. Setiap tombol ON/OFF mengontrol satu perangkat, dan status perangkat diperbarui otomatis agar konsisten antara tampilan dan kondisi fisik. Pengujian

dilakukan dengan memberi perintah ON/OFF baik secara individual maupun serentak, serta mengukur waktu respons dan sinkronisasi status perangkat, dalam jaringan lokal untuk memastikan latensi minimal.



Gambar 13. Tampilan Antarmuka Web Dashboard Kipas Menyala



Gambar 14. Implementasi Fisik Kipas Menyala

Tabel 6 Hasil Pengujian Kendali Manual melalui Web Dashboard

| No | Perangkat | Aksi | Status Perangkat | Waktu Respon | Hasil |
|-----|------------------|------|------------------|--------------|----------|
| 1. | Kipas | ON | Kipas ON | 0,76 detik | Berhasil |
| 2. | Kipas | OFF | Kipas OFF | 0,30 detik | Berhasil |
| 3. | Lampu Kamar | ON | Kamar ON | 0,35 detik | Berhasil |
| 4. | Lampu Kamar | OFF | Kamar OFF | 0,70 detik | Berhasil |
| 5. | Lampu Ruang Tamu | ON | Ruang Tamu ON | 0,25 detik | Berhasil |
| 6. | Lampu Ruang Tamu | OFF | Ruang Tamu OFF | 0,37 detik | Berhasil |
| 7. | Lampu Teras | ON | Teras ON | 0,20 detik | Berhasil |
| 8. | Lampu Teras | OFF | Teras OFF | 0,56 detik | Berhasil |
| 9. | Nyalakan Semua | ON | Semua ON | 0,40 detik | Berhasil |
| 10. | Matikan Semua | OFF | Semua OFF | 0,80 detik | Berhasil |

Rata-Rata = $4,69 / 10 = 0,47$ detik

Tabel 7. Hasil Pengujian Trigger Web Dashboard melalui Perintah Suara

| No | Perintah Suara | Status Web Dashboard | Status Perangkat | Waktu Respon | Hasil |
|----|---------------------------|------------------------------|------------------|--------------|----------|
| 1. | Hidupkan lampu kamar | <i>Toggle</i> berubah ke ON | Kamar ON | 1,09 detik | Berhasil |
| 2. | Matikan lampu kamar | <i>Toggle</i> berubah ke OFF | Kamar OFF | 0,94 detik | Berhasil |
| 3. | Hidupkan lampu ruang tamu | <i>Toggle</i> berubah ke ON | Ruang Tamu ON | 0,66 detik | Berhasil |

| | | | | | |
|----|--------------------------|------------------------------|-----------------|------------|----------|
| 4. | Matikan lampu ruang tamu | <i>Toggle</i> berubah ke OFF | Ruang Tamu OFF | 1,22 detik | Berhasil |
| 5. | Hidupkan lampu teras | <i>Toggle</i> berubah ke ON | Teras ON | 1,12 detik | Berhasil |
| 6. | Matikan lampu teras | <i>Toggle</i> berubah ke OFF | Teras OFF | 1,72 detik | Berhasil |
| 7. | Hidupkan semua lampu | <i>Toggle</i> berubah ke ON | Semua lampu ON | 1,56 detik | Berhasil |
| 8. | Matikan semua lampu | <i>Toggle</i> berubah ke OFF | Semua lampu OFF | 1,03 detik | Berhasil |

Rata-rata = $9,34 / 8 = 1,17$ detik

Tabel 8. Hasil Pengujian Trigger Web Dashboard melalui Sensor Cahaya

| No | Kondisi Cahaya | Status Perangkat | Status Web Dashboard | Hasil |
|----|----------------|------------------|------------------------------|----------|
| 1. | Gelap | Semua lampu ON | <i>Toggle</i> berubah ke ON | Berhasil |
| 2. | Terang | Semua lampu OFF | <i>Toggle</i> berubah ke OFF | Berhasil |

Pengujian menunjukkan bahwa rata-rata waktu respons kontrol manual melalui *dashboard* adalah 0,47 detik, sementara melalui Google Assistant membutuhkan 1,17 detik karena proses melalui *cloud*. Meski ada perbedaan waktu respons, semua metode *input* dashboard, suara, dan otomatisasi sensor berhasil menjalankan kendali dengan stabil dan akurat. Status perangkat selalu tersinkronisasi dengan antarmuka *dashboard*, membuktikan bahwa sistem mampu mendukung kontrol dua arah secara *real-time* dengan performa responsif, baik secara lokal maupun melalui layanan *cloud*.

Hasil Fungsionalitas

Pengujian fungsionalitas sistem dilakukan untuk memastikan bahwa ketiga mode control otomatis via sensor LDR, perintah suara melalui Google Assistant, dan kontrol manual melalui *web dashboard* dapat beroperasi secara harmonis tanpa konflik. Fokus utama terletak pada sinkronisasi status perangkat dan validasi logika *override*, di mana jika perangkat dimatikan secara manual, sistem tidak akan menyalakannya kembali secara otomatis hingga ada perintah eksplisit dari pengguna. Hasil pengujian menunjukkan bahwa logika *override* berfungsi dengan baik, menjaga konsistensi kontrol dan mencegah intervensi otomatis yang tidak diinginkan.



Gambar 15. Simulasi Override Manual Kondisi Gelap

```
Output Serial Monitor X
Message (Enter to send message to 'ESP32 Dev Module' on COM5)
MQTT lampu_kamar_0
PENGACAUAN SENSOR
LDR: 4095
Override status Aktif
Override Ruang Tamu Tidak Aktif
Override teras: Tidak Aktif
Gelap - Lampu kamar tidak dialumakan (override aktif)
Gelap - Nyala lampu ruang tamu
Gelap nyala lampu teras
```

Gambar 16. Serial Monitor yang Menunjukkan Flag Override Aktif

Pembahasan Hasil

Hasil pengujian sistem *Smart Home* berbasis mikrokontroler ESP32 menunjukkan bahwa seluruh komponen utama yaitu kontrol otomatis berbasis sensor cahaya (LDR), kontrol suara menggunakan Google Assistant melalui integrasi IFTTT dan Adafruit IO, serta kontrol manual melalui *web dashboard* berbasis WebSocket berfungsi secara optimal dan sesuai dengan logika sistem yang telah dirancang. Integrasi antar *platform* ini membuktikan bahwa penggunaan teknologi *cloud* dapat diimplementasikan secara efektif untuk kendali perangkat jarak jauh berbasis MQTT, dengan rata-rata waktu respon

perintah suara sekitar 1,9 detik yang masih tergolong responsif.

Sensor LDR mampu mendeteksi perubahan intensitas cahaya lingkungan secara *real-time* dengan interval pembacaan setiap dua detik. Nilai ADC dari sensor dijadikan dasar pengambilan keputusan otomatis, seperti pemicu penyalaan perangkat saat kondisi gelap ($ADC > 1500$) atau pemadaman saat kondisi terang ($ADC < 100$). Hal ini menunjukkan bahwa pemetaan antara logika sistem dan *output* sensor telah bekerja secara akurat.

Sementara itu, kontrol manual melalui dashboard berbasis WebSocket juga berjalan lancar dengan waktu respon rata-rata hanya 0,47 detik. *Dashboard* ini tidak hanya berfungsi sebagai sarana kendali manual, tetapi juga sebagai antarmuka visual yang menyajikan status perangkat secara *real-time* serta mendukung mekanisme *override* yang memungkinkan pengguna untuk mengutamakan perintah manual dibandingkan otomatisasi.

Keunggulan utama sistem terletak pada kemampuannya mengintegrasikan berbagai metode *input* dalam satu kesatuan sistem tanpa menyebabkan konflik logika. *Override* yang diaktifkan melalui perintah suara mampu menahan aksi otomatisasi sensor, bahkan ketika kondisi lingkungan sebenarnya mendukung penyalaan perangkat. Hal ini dibuktikan melalui tampilan serial monitor pada Arduino IDE yang mencetak status “*override aktif*” saat sistem menunda aksi otomatis. Pendekatan ini menjadikan sistem lebih fleksibel, cerdas, dan responsif terhadap perintah pengguna tanpa mengesampingkan efisiensi pembacaan sensor.

Secara keseluruhan, implementasi sistem ini membuktikan bahwa prinsip-prinsip dasar *Internet of Things* (IoT), seperti penggabungan sensor, mikrokontroler, dan layanan *cloud*, dapat diterapkan secara harmonis tanpa memerlukan arsitektur kompleks. Ketiga metode kontrol otomatisasi berbasis sensor

cahaya, perintah suara berbasis *cloud*, dan kontrol manual berbasis *dashboard* berhasil diintegrasikan dalam satu sistem yang saling melengkapi dan mendukung adaptasi terhadap berbagai skenario penggunaan.

Jika dibandingkan, kontrol berbasis suara memiliki keunggulan dalam hal fleksibilitas dan kenyamanan bagi pengguna yang ingin mengoperasikan perangkat dari jarak jauh. Namun, metode ini bergantung pada kestabilan koneksi internet dan keakuratan frasa perintah, dengan tingkat keberhasilan pengujian mencapai 100%. Di sisi lain, metode otomatisasi cahaya lebih stabil karena tidak memerlukan intervensi langsung pengguna dan menunjukkan kinerja konsisten tanpa kegagalan selama pengujian.

Dashboard manual memberikan alternatif kendali cepat dan stabil, terutama saat digunakan dalam jaringan lokal. Kombinasi ketiga metode ini membentuk sistem kontrol perangkat rumah tangga yang tanggap, adaptif, dan cerdas, sekaligus menunjukkan validitas sistem terhadap berbagai kondisi interaksi, baik secara langsung maupun otomatis.

SIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang dilakukan terhadap sistem *Smart Home* berbasis mikrokontroler ESP32, dapat disimpulkan bahwa sistem telah mampu mengontrol perangkat rumah secara otomatis dan manual. Penggunaan sensor LDR untuk mendeteksi kondisi pencahayaan lingkungan terbukti efektif dengan keberhasilan respon 100% untuk mengatur status perangkat secara otomatis, dengan pembacaan setiap dua detik dan keputusan diambil berdasarkan ambang batas intensitas cahaya. Kondisi gelap menyebabkan perangkat menyala, sementara kondisi terang memicu pemadaman perangkat secara otomatis. Tidak ditemukan kegagalan dalam uji coba otomatisasi ini.

Di sisi lain, sistem juga mendukung kendali manual melalui perintah suara yang

dikirimkan via Google Assistant dan dihubungkan ke ESP32 melalui IFTTT dan Adafruit IO. Sistem berhasil merespon dua belas dari tiga belas perintah suara yang menghasilkan tingkat keberhasilan 100% dengan rata-rata waktu respon sekitar 1,9 detik. Kegagalan terjadi hanya pada satu skenario akibat frasa perintah yang tidak sesuai. Hal ini menunjukkan bahwa sistem cukup andal dalam komunikasi *cloud* dan responsif terhadap perintah pengguna. Selain itu, pengujian *web dashboard* menunjukkan tingkat keberhasilan 100% dengan rata-rata waktu respon 0,47 detik, membuktikan bahwa *dashboard* dapat berfungsi sebagai antarmuka kontrol lokal yang stabil dan responsif.

Integrasi antara kontrol otomatis dan manual diperkuat dengan implementasi logika *override*, yang memungkinkan pengguna untuk mematikan perangkat secara manual dan mencegah sistem menyalakannya kembali meskipun kondisi sensor mendukung penyalakan. Pengujian melalui serial monitor menunjukkan bahwa logika *override* ini berjalan sesuai perancangan dan menjaga kestabilan logika sistem.

Dengan demikian, sistem berhasil menjawab seluruh rumusan masalah yang telah ditetapkan, mulai dari otomatisasi berdasarkan cahaya, kontrol berbasis suara, integrasi keduanya secara bersamaan, hingga implementasi logika *override*. Sistem ini dinilai cukup layak dan efektif untuk diterapkan dalam skala rumah sederhana berbasis *Internet of Things* (IoT).

DAFTAR PUSTAKA

- Abdul Kholik, M., Setiawan, B., Fauzi, A., & Surakarta, U. (2023). Design and implementation of a smart home system with the internet of things (iot) using esp32. *Jurnal Teknologi Informasi Dan Komunikasi*, 13(1), 22–30.
- Akbar, A., Mutaqin, Z., Delsi Samsumar, L., & Teknologi Mataram Corresponding Author, U. (2022). IoT-Based Smart Room Using Web Server-Based Esp32 Microcontroller. *Formosa Journal of Computer and Information Science (FJCIS)*, 1(2), 91–98. <https://doi.org/10.55927/fjcis.v1i2.1151>
- Ayu Syahfitri. (2025). Internet of Things (IoT), Sejarah, Teknologi, dan Penerapannya. *Uranus : Jurnal Ilmiah Teknik Elektro, Sains Dan Informatika*, 3(1), 113–120. <https://doi.org/10.61132/uranus.v3i1.667>
- Badruzzaman, N., Bhakti, H., & Bachri, O. S. (2024). IMPLEMENTASI SISTEM MONITORING SUHU DAN PH AIR KOLAM BUDIDAYA IKAN LELE MENGGUNAKAN ARDUINO ESP8266 DAN ARDUINO IDE. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(3S1), 4423–4432. <https://doi.org/10.23960/jitet.v12i3S1.5355>
- FTMM UNAIR. (2022, April 22). Rumah Pintar : Mengintegrasikan IoT dan CPS dalam Era Industri 4.0. <https://ftmm.unair.ac.id/rumah-pintar-mengintegrasikan-iot-dan-cps-dalam-era-industri-4-0/>
- Gallileo, A., & Badrul, M. (2025). Rancangan Smarthome Berbasis Mikrokontroler Sebagai Pengendali Pintu Rumah Dan Garasi Serta Lampu. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 12(1).
- Hartono. (2019). METODOLOGI PENELITIAN. ZANAFA PUBLISHING.
- Hendrian, Y., Yudatama, Y. P., & Pratama, V. S. (2020). Jemuran Otomatis Menggunakan Sensor LDR, Sensor Hujan Dan Sensor Kelembaban Berbasis Arduino Uno. *Jurnal Khatulistiwa Informatika*, 6(1), 21–30.
- Indah Purnama, S., Sonatha, Y., & Humaira. (2021). Fitting (Tempat Lampu) Berbasis Internet of Things

- (IoT) Menggunakan Arduino. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, 2(1), 27–31. <http://jurnal-itsi.org>
- Khoirunnisa. (2023, April 5). ASIOTI: Smart Home di Indonesia Capai 835 Juta Rumah. <https://selular.id/2023/04/asioti-smart-home-di-indonesia-capai-835-juta-rumah/>
- Kimutai, M. S., Kelvin, O. K., Jasper, O. M., Kimutai, M. S., Kelvin, O. K., & Jasper, O. M. (2022). Challenges and Opportunities for Smart Homes Deployment in Developing Countries: A Case Study of the User Perspective in Kenya. *Open Access Library Journal*, 9(7), 1–8. <https://doi.org/10.4236/OALIB.1107679>
- Laboy, N. R., Vahlevi, P. A., Sutabri, T., Rizki, M., Bina, U., & Palembang, D. (2024). Analisis Penerapan Internet of Things (Iot) Dalam Smart Home System. *Jurnal Ilmu Teknik*, 1(2), 283–285. <https://doi.org/10.62017/tektonik>
- Manuhutu, A., Warella, S., Likliwatin, C., Sasauw, C., Wilson Sitopu, J., Negeri Ambon, P., & Kata Kunci, A. (2025). Mengubah Kehidupan Sehari-hari: Dampak Implementasi Internet of Things (IoT). *Indonesian Research Journal on Education Web Jurnal Indonesian Research Journal on Education*, 5(1), 1072–1078.
- Muntasiroh, L., Mariani, D., & Sumarno, R. N. (2023). Simulasi Rangkaian Catu Daya +12V dan +5V Menggunakan Multisim Pada Mata Kuliah Praktikum Dasar Elektronika Secara Daring. *Jurnal Teknik Elektro Dan Komputasi (ELKOM)*, 5(2), 238–243. <http://jurnal.unmuhjember.ac.id/index.php/ELKOM/article/view/7872>
- Niken Puspita, P., Syamsu, M., & Terisia, V. (2023). Perancangan Lampu Otomatis Menggunakan Sensor Cahaya Dan Timer Berbasis Arduino Uno. 4(1), 42–49.
- Putra Lesmana, S., Putra, A., Merah, S. B., Hermawati3, D., & Puspitasari, N. (2024). Dampak Implementasi IoT pada Sistem Smart Home untuk Efisiensi Energi dan Keamanan di Kota Berkembang. Prosiding Seminar Nasional Amikom Surakarta, 2, 1265–1278.
- Rahmatina, R., Aripin, M. N., Ikbal, M., Deolika, A., Komputer, J. T., Kalimantan, I., Rta, J., Km, M., & Raya, P. (2023). Implementasi Transistor BD139 dan Rangkaian Relay pada Mesin Air. *JIFOTECH (JOURNAL OF INFORMATION TECHNOLOGY)*, 3(1), 11–18.
- Saputra, T., & Surapati, U. (2024). Analysis of the Effectiveness of IoT-Based Automatic Street Lighting Control Using Linear Regression Method. *International Journal Software Engineering and Computer Science (IJSECS)*, 4(2), 690–701. <https://doi.org/10.35870/ijsecs.v4i2.2878>
- Wibisono Darmawan, C., U A Sompie, S. R., & Kambey, F. D. (2020). Implementasi Internet of Things pada Monitoring Kecepatan Kendaraan Bermotor. *Jurnal Teknik Elektro Dan Komputer*, 9(2), 91–100.
- Wijaya, K., & Ceng Giap, Y. (2024). DISPENSER OTOMATIS BERBASIS MIKROKONTROLER DENGAN LED INFORMASI DAN SENSOR SUHU. *Jurnal Mahasiswa Teknik Informatika*, 8(6), 11745–11751.
- Zein, A., Sita Eriana, E., Raya, J., Serpong, P., 10 Tangerang, N., & Banten, S. (2022). Perancangan Internet of Things (IoT) Smart Home. *Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*, 31(2), 46–51.