

RANCANG BANGUN OTOMATISASI LAMPU RUANGAN BERBASIS *INTERNET OF THINGS* DENGAN KENDALI WEB SERVER LOKAL

Sujono¹⁾, Erdi Masyudi²⁾

^{1,2}Prodi Informatika, Fakultas Teknologi Informasi, Universitas KH. Abdul Wahab Hasbullah

Correspondence author: E. Masyudi, erdimasyudi@gmail.com, Jombang, Indonesia

Abstract

This research aims to design and implement a room lighting control automation system that is completely independent from external internet connections. This technique addresses common issues in cloud-based Internet of Things (IoT) systems, namely high latency issues and data privacy risks. The method used is a systematic experimental approach that utilizes an ESP32 microcontroller as an Embedded Web Server operating exclusively within a local Wi-Fi network. The software implementation uses a combination of HTML, CSS, and JavaScript (AJAX) to create a responsive control interface. Functional testing results prove 100% success in ON/OFF light operation, with superior response performance. The system's average response time is 100–150 milliseconds (ms), demonstrating that it is very fast and efficient. Furthermore, this local architecture improves security because access to control and action history data is limited to devices within the local network. The developed system is an effective, stable, and secure solution that successfully mitigates the risk of reliance on third-party servers, making it ideal for users who prioritize privacy and access speed in a home environment.

Keywords: Internet of Things, local web server, ESP32, lighting control automation

Abstrak

Penelitian ini bertujuan merancang dan mengimplementasikan sistem otomasi kendali lampu ruangan yang sepenuhnya mandiri dari koneksi internet eksternal. Teknik ini untuk mengatasi masalah umum pada sistem *Internet of Things* (IoT) berbasis *cloud*, yaitu isu latensi tinggi dan risiko privasi data. Metode yang digunakan adalah pendekatan eksperimental sistematis dengan memanfaatkan mikrokontroler ESP32 sebagai *Embedded Web Server* yang beroperasi eksklusif dalam jaringan Wi-Fi lokal. Implementasi perangkat lunak menggunakan kombinasi HTML, CSS, dan JavaScript (AJAX) untuk menciptakan antarmuka kendali yang responsif. Hasil pengujian fungsionalitas membuktikan keberhasilan 100% dalam pengoperasian lampu ON/OFF, dengan kinerja respons yang superior. Waktu respons rata-rata sistem mencapai 100–150 milidetik (ms), membuktikan sistem sangat cepat dan efisien. Lebih lanjut, arsitektur lokal ini meningkatkan keamanan karena akses kontrol dan data riwayat aksi terbatas hanya pada perangkat dalam jaringan lokal. Sistem yang dikembangkan merupakan solusi efektif, stabil, dan aman yang berhasil memitigasi risiko ketergantungan pada *server* pihak ketiga, ideal bagi pengguna yang memprioritaskan privasi dan kecepatan akses di lingkungan rumah.

Kata Kunci: *Internet of Things*, web server lokal, ESP32, otomasi lampu

A. PENDAHULUAN

Perkembangan teknologi *Internet of Things* (IoT) telah membawa perubahan signifikan dalam berbagai aspek kehidupan manusia secara global. IoT memungkinkan perangkat fisik saling terhubung dan berkomunikasi melalui jaringan internet, sehingga meningkatkan efisiensi dan kemudahan pengelolaan berbagai sistem, termasuk dalam bidang otomasi rumah, kesehatan, hingga industri (Rahmah et al., 2020). Secara global, tren penerapan IoT terus meningkat, terlihat dari jumlah perangkat yang terhubung yang diprediksi mencapai miliaran unit dalam beberapa tahun terakhir, menandakan urgensi pengembangan teknologi ini dalam berbagai sektor (Shiddiqi et al., 2020).

Di tingkat nasional, perkembangan IoT di Indonesia juga mengalami kemajuan pesat, dengan penetrasi internet yang mencapai sekitar 65% dari total populasi, sehingga semakin memudahkan penerapan sistem berbasis IoT dalam kehidupan sehari-hari (Shiddiqi et al., 2020). Salah satu implementasi yang berperan penting adalah dalam *smart home*, di mana kecanggihan IoT dimanfaatkan untuk mengendalikan peralatan elektronik secara otomatis dan jarak jauh, memberikan kenyamanan dan efisiensi bagi penggunaanya (Wicaksono & Rahmatya, 2020). Namun, sebagian besar sistem IoT mengandalkan *server cloud* yang terkadang menghadapi kendala seperti ketergantungan pada internet yang stabil dan risiko keamanan data (Setiyani et al., 2019).

Khusus di tingkat lokal, banyak pengguna rumah atau institusi yang mulai menerapkan kendali otomatis pada perangkat pencahayaan sebagai bagian dari penghematan energi dan peningkatan kenyamanan, terutama pada ruangan-ruangan fungsional seperti laboratorium atau kantor (Nugroho, 2021). Sistem kendali lampu otomatis telah dikembangkan dengan menggunakan mikrokontroler yang terhubung ke jaringan internet, contohnya NodeMCU ESP8266 dan ESP32 yang populer di kalangan pengembang

IoT (Sawidin et al., 2024). Namun, pengguna juga menginginkan solusi yang tidak sepenuhnya bergantung pada koneksi internet dengan *server cloud* agar sistem lebih responsif dan aman.

Penelitian sebelumnya, seperti yang dilakukan oleh (Sawidin et al., 2024), mengembangkan sistem kendali lampu berbasis IoT menggunakan *Arduino Cloud* dengan mikrokontroler NodeMCU ESP8266 yang memungkinkan pengendalian jarak jauh melalui jaringan internet. Selain itu, (Abdussomad et al., 2022) merancang sistem kendali lampu otomatis berbasis IoT dengan aplikasi Blynk 2.0 yang juga menggunakan NodeMCU ESP8266 untuk memberikan kemudahan dalam pengontrolan melalui *smartphone*. Kedua penelitian tersebut memberikan kontribusi penting dalam pengembangan solusi kendali lampu otomatis namun masih mengandalkan *server cloud* dan koneksi internet aktif.

Berbeda dengan penelitian tersebut, penelitian ini merancang dan membangun sistem otomatisasi lampu ruangan berbasis IoT dengan kendali melalui web *server* lokal yang berjalan langsung pada mikrokontroler ESP32. Pendekatan ini menghilangkan ketergantungan pada koneksi internet dan *server cloud*, sehingga meningkatkan responsivitas dan keamanan pengendalian lampu di lingkungan lokal jaringan Wi-Fi (*local area network*) (Abdussomad et al., 2022). Sistem ini dapat diakses dari perangkat apa pun yang tersambung ke jaringan Wi-Fi lokal, seperti komputer atau *smartphone*, melalui alamat IP khusus yang meng-host halaman web pengontrol lampu. Dengan konsep ini, diharapkan pengguna dapat mengelola pencahayaan ruangan dengan lebih efisien, praktis, dan aman.

Tujuan dari penelitian ini adalah untuk merancang bangun sebuah sistem kendali lampu ruangan berbasis *Internet of Things* (IoT) dengan kendali menggunakan web *server* lokal yang berjalan pada mikrokontroler ESP32, sehingga memberikan solusi otomatisasi lampu yang responsif tanpa bergantung pada koneksi internet eksternal

untuk meningkatkan efisiensi pengelolaan pencahayaan pada ruangan.

B. METODE PENELITIAN

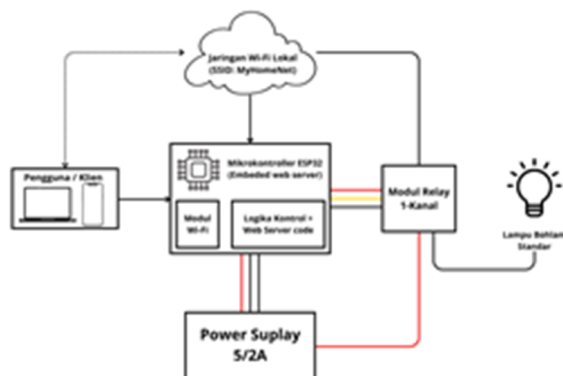
Metodologi penelitian ini mengadopsi pendekatan eksperimental sistematis untuk merancang, mengimplementasikan, dan mengevaluasi sistem otomasi lampu ruangan berbasis *Internet of Things* (IoT) dengan kendali web server lokal. Tahapan penelitian dirancang secara terstruktur untuk memastikan validitas dan reliabilitas hasil. Proses ini secara spesifik bertujuan untuk menciptakan sistem kendali lampu yang otomatis, responsif, dan independen dari layanan *cloud* eksternal, dengan memanfaatkan jaringan Wi-Fi lokal sebagai infrastruktur komunikasinya (Saputra & Surapati, 2024).

Perancangan Arsitektur Sistem

Perancangan arsitektur sistem adalah langkah awal yang krusial untuk mendefinisikan struktur dan interaksi antar komponen. Arsitektur yang diusulkan menekankan pada konsep desentralisasi kendali, di mana mikrokontroler berfungsi sebagai node cerdas yang mandiri.

Diagram Blok Sistem

Untuk memberikan gambaran menyeluruh tentang interkoneksi antar komponen, disajikan diagram blok sistem pada Gambar 1.

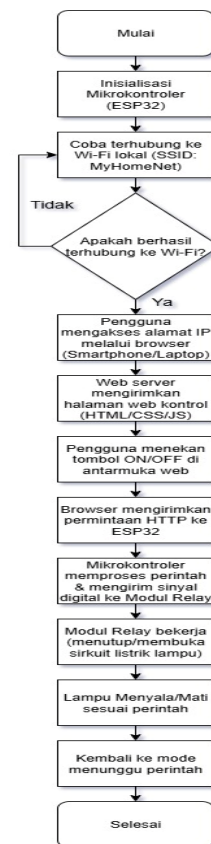


Gambar 1. Diagram blok sistem otomasi lampu ruangan.

Diagram blok tersebut secara jelas mengilustrasikan alur komunikasi dan kontrol. Pengguna (klien) dapat berupa perangkat *smartphone* atau komputer yang terhubung ke jaringan Wi-Fi lokal. Perintah kontrol dikirimkan dari perangkat pengguna ke mikrokontroler ESP32, yang berperan ganda sebagai *embedded web server* dan unit pemroses logika. Mikrokontroler kemudian menerjemahkan perintah tersebut menjadi sinyal digital untuk mengendalikan Modul Relay, yang selanjutnya berfungsi sebagai sakelar elektronik untuk mengaktifkan atau menonaktifkan Lampu. Seluruh sistem ditenagai oleh Power Supply 5V/2A yang stabil.

Alur Kerja Sistem (*Flowchart*)

Alur kerja sistem secara rinci dijelaskan melalui *flowchart* pada Gambar 2, yang menggambarkan sekuensial proses dari inisialisasi hingga eksekusi perintah kontrol



Gambar 2. *Flowchart* alur kerja sistem.

Perangkat Keras dan Perangkat Lunak

Pemilihan perangkat keras dan perangkat lunak didasarkan pada ketersediaan, biaya, dan kemampuannya untuk memenuhi persyaratan fungsional sistem.

1. Perangkat Keras

Tabel 1 menyajikan daftar komponen perangkat keras utama yang digunakan dalam pengembangan sistem ini beserta fungsi spesifiknya.

Tabel 1. Daftar Perangkat Keras Utama

Komponen Perangkat Keras	Deskripsi dan Fungsi
Mikrokontroler ESP32	Berfungsi sebagai unit pemroses utama dan <i>embedded web server</i> .
Modul Relay 1-Kanal (5V)	Bertindak sebagai sakelar elektronik untuk mengendalikan beban tegangan tinggi (220V AC).
Lampu Bohlam Standar	Sebagai aktuator utama yang dikendalikan oleh sistem.
Power Supply DC 5V / 2A	Menyediakan sumber daya listrik yang stabil untuk ESP32 dan Modul Relay.
Kabel Jumper dan Breadboard	Digunakan untuk prototipe awal dan interkoneksi antar komponen.

2. Perangkat Lunak

Implementasi perangkat lunak melibatkan beberapa platform dan bahasa pemrograman seperti yang dirinci pada Tabel 2.

Tabel 2. Daftar Perangkat Lunak

Komponen Perangkat Lunak	Deskripsi dan Fungsi
Arduino IDE (<i>Integrated Development Environment</i>)	Lingkungan pengembangan terintegrasi untuk menulis, mengkompilasi, dan mengunggah kode program ke mikrokontroler ESP32. Menyediakan framework dan library yang memudahkan pengembangan aplikasi <i>embedded</i> .

Komponen Perangkat Lunak

Deskripsi dan Fungsi

Bahasa Pemrograman C++	Digunakan untuk menulis <i>firmware</i> yang berjalan pada mikrokontroler ESP32. Termasuk logika kontrol, inisialisasi Wi-Fi, dan implementasi <i>web server</i> .
HTML (<i>HyperText Markup Language</i>)	Bahasa markup standar untuk membuat struktur halaman web. Digunakan untuk mendefinisikan elemen-elemen antarmuka kontrol seperti tombol dan teks.
CSS (<i>Cascading Style Sheets</i>)	Digunakan untuk mengatur tampilan visual dan layout halaman web agar lebih menarik dan user-friendly, serta responsif pada berbagai ukuran layar perangkat klien.
JavaScript	Bahasa pemrograman sisi klien yang digunakan untuk membuat antarmuka web menjadi interaktif. Khususnya digunakan untuk implementasi AJAX (<i>Asynchronous JavaScript and XML</i>) agar permintaan kontrol (<i>ON/OFF</i>) dapat dikirim ke <i>web server</i> ESP32 tanpa memuat ulang seluruh halaman, menghasilkan pengalaman pengguna yang lebih mulus dan responsif.
Library <i>WebServer.h</i> (untuk ESP32)	Pustaka khusus yang mempermudah implementasi fungsionalitas <i>web server</i> HTTP pada ESP32. Menyediakan fungsi untuk menangani permintaan klien, merespons dengan konten HTML, dan mengelola <i>routing</i> URL.

Tahap Implementasi

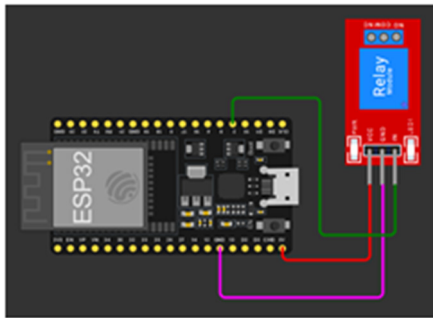
Tahap implementasi fokus pada perakitan fisik komponen dan pengembangan kode program

1. Proses Instalasi (*Hardware Assembly*)

Proses instalasi perangkat keras meliputi langkah-langkah sebagai berikut:

- Koneksi Mikrokontroler dan Relay: Pin digital pada ESP32 (misalnya GPIO2)

- dihubungkan ke pin input kontrol pada Modul Relay.
- Koneksi Daya: Power supply 5V/2A dihubungkan ke pin 5V (atau VIN) dan GND pada ESP32 serta pin VCC dan GND pada Modul Relay. Penting untuk memastikan polaritas yang benar untuk menghindari kerusakan.
 - Koneksi Beban (Lampu): Modul Relay dihubungkan secara seri dengan lampu bohlam dan sumber listrik AC 220V. Perhatian khusus pada keamanan kelistrikan harus diberikan, memastikan sambungan yang kuat dan terisolasi untuk menghindari korsleting atau bahaya sengatan listrik.



Gambar 3. Wire ESP32 dan modul relay.

2. Perancangan dan Implementasi Web Server

Pengembangan web server lokal pada ESP32 adalah inti dari sistem ini.

- Inisialisasi Wi-Fi: Kode program dalam Arduino IDE akan menginisialisasi modul Wi-Fi internal ESP32 untuk terhubung ke jaringan Wi-Fi lokal dengan SSID dan password yang telah ditentukan.
- Aktivasi Web Server: Menggunakan library `WebServer.h`, sebuah objek web server diinisialisasi pada port HTTP standar (port 80).
- Penanganan *Routing* Permintaan (Request Handling): Fungsi-fungsi handler didefinisikan untuk merespons permintaan HTTP yang masuk. Contohnya, permintaan ke URL root (/) akan mengembalikan halaman web kontrol utama, sementara permintaan ke /on atau /off akan memicu fungsi untuk

mengaktifkan atau menonaktifkan Modul Relay.

- Logika Kontrol: Di dalam fungsi handler ini, logika kontrol sederhana diterapkan untuk mengatur state pin GPIO yang terhubung ke Modul Relay. Jika permintaan /on diterima, pin diatur ke HIGH (logika aktif), dan sebaliknya untuk /off.
- Penyajian Halaman Web: Konten halaman web (HTML, CSS, JavaScript) disimpan sebagai string di dalam sketch Arduino atau dapat dimuat dari filesystem ESP32 (SPIFFS atau LittleFS) untuk halaman yang lebih kompleks.

3. Perancangan Antarmuka Web (*User Interface*)

Antarmuka web dirancang agar intuitif dan responsif:

- Struktur HTML: Kode HTML menyediakan kerangka dasar halaman web, termasuk tombol-tombol (`<button>`) untuk mengontrol status lampu (ON/OFF) serta indikator status lampu.
- Gaya CSS: CSS digunakan untuk mempercantik tampilan tombol, menyesuaikan font, warna, layout, dan memastikan halaman web terlihat konsisten dan user-friendly di berbagai perangkat (responsive design).
- Interaktivitas JavaScript (AJAX): JavaScript memainkan peran kunci dalam pengalaman pengguna. Ketika tombol ON/OFF ditekan, fungsi JavaScript akan mengirimkan permintaan HTTP asinkron (AJAX request) ke web server ESP32 tanpa memuat ulang seluruh halaman. Hal ini memastikan respons yang cepat dan pengalaman pengguna yang mulus, meniru responsivitas aplikasi native.

Tahap Pengujian

Pengujian dilakukan untuk memvalidasi fungsionalitas, konektivitas, dan kinerja sistem.

1. Pengujian Fungsionalitas:

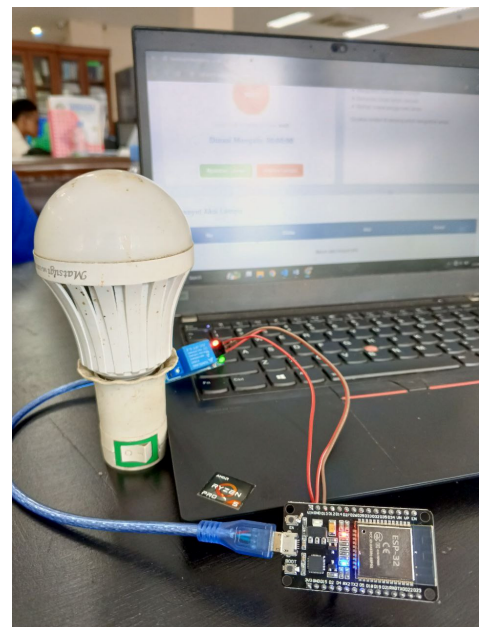
- a. Skenario: Pengguna mengirimkan perintah ON dan OFF berulang kali melalui antarmuka web.
 - b. Metrik: Observasi langsung apakah lampu merespons perubahan status sesuai dengan perintah yang diberikan. Tingkat keberhasilan diharapkan 100% dalam kondisi normal.
 - c. Tujuan: Memastikan bahwa logika kontrol dan aktuasi relay bekerja dengan benar.
2. Pengujian Konektivitas:
- a. Skenario: Percobaan koneksi ESP32 ke berbagai jaringan Wi-Fi lokal dan akses web *server* dari beberapa perangkat klien (misalnya, *smartphone*, laptop, tablet) secara bersamaan atau bergantian.
 - b. Metrik: Keberhasilan koneksi Wi-Fi ESP32, kemampuan perangkat klien untuk mengakses alamat IP web *server*, dan stabilitas koneksi selama sesi kontrol.
 - c. Tujuan: Memastikan sistem dapat beroperasi secara handal dalam lingkungan jaringan lokal yang bervariasi.
3. Pengukuran Latensi (Kinerja Respons):
- a. Skenario: Mengukur waktu tunda antara momen penekanan tombol kontrol pada antarmuka web hingga lampu menunjukkan perubahan status fisik (menyala/mati).
 - b. Metrik: Waktu diukur dalam milidetik (ms). Pengukuran dilakukan menggunakan timer pada sisi browser (JavaScript) dan/atau observasi visual yang didukung video recording berkecepatan tinggi jika diperlukan presisi ekstrem.
 - c. Tujuan: Mengevaluasi tingkat responsivitas sistem *off-cloud* dibandingkan dengan potensi sistem berbasis *cloud*, menegaskan klaim tentang latensi rendah.

Pengujian ini akan dilakukan berulang kali dalam berbagai kondisi untuk menjamin stabilitas, keandalan, dan efektivitas sistem di

bawah beban normal jaringan Wi-Fi local (Saputra & Surapati, 2024).

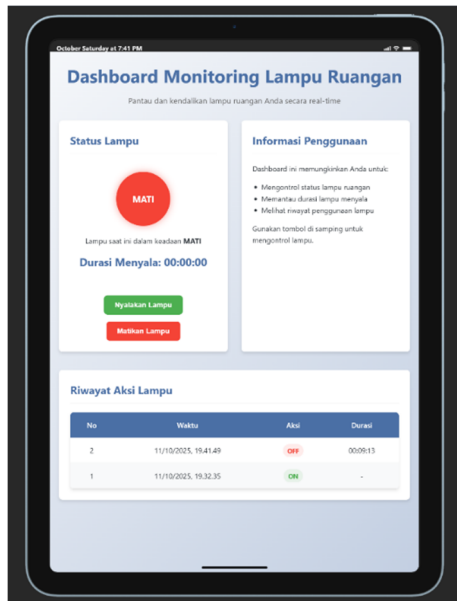
C. HASIL DAN PEMBAHASAN

Tahap implementasi berhasil mewujudkan sistem kendali lampu berbasis *Internet of Things* (IoT) yang mandiri menggunakan arsitektur *off-cloud*. Karakteristik utama subjek penelitian, yaitu perangkat keras, diilustrasikan pada Gambar 4. Perangkat terdiri dari Mikrokontroler ESP32 sebagai embedded web *server* dan Modul Relay 1-Kanal yang berfungsi sebagai saklar utama untuk lampu AC 220V.



Gambar 4. Purwarupa (prototype) implementasi perangkat keras.

Pada Gambar 4, terlihat koneksi antara ESP32 dengan Modul Relay yang mengontrol lampu bohlam, di mana lampu berada dalam kondisi MATI. Sementara itu, antarmuka visual yang menjadi titik interaksi pengguna, diimplementasikan sebagai web *server* lokal yang diakses melalui peramban (*browser*) pada perangkat klien, seperti yang ditampilkan pada Gambar 5.



Gambar 5. Antarmuka dashboard monitoring lampu ruangan.

Antarmuka dashboard pada Gambar 5 dirancang untuk memberikan informasi status lampu secara cepat dan menyediakan kontrol yang responsif. Fitur kunci dashboard ini mencakup tampilan status lampu (MENYALA/MATI), pemantauan durasi lampu menyala, dan pencatatan riwayat aksi (logging) yang dikelola langsung oleh memori lokal pada ESP32. Fitur logging ini adalah bukti nyata kemampuan sistem dalam menyimpan dan menyajikan data penggunaan tanpa bergantung pada database *cloud* eksternal.

Pengujian fungsionalitas sistem kendali lampu menunjukkan keberhasilan 100% dalam pengoperasian fungsi ON/OFF lampu, di mana status fisik lampu (Gambar 4) selalu sinkron dengan status yang tertera pada dashboard (Gambar 5). Keandalan fungsional ini didukung oleh pengujian kinerja yang berfokus pada pengukuran waktu respons (latensi), yang merupakan keunggulan utama dari arsitektur web *server* lokal. Hasil analisis menunjukkan waktu respons rata-rata yang sangat cepat, berada pada kisaran 100-150 milidetik (ms). Kecepatan ini dicapai karena perintah kontrol hanya berjalan dalam jaringan Wi-Fi lokal, menghilangkan *delay* komunikasi yang disebabkan oleh *routing*

data melalui *server cloud* di internet (Saputra et al., 2024).

Meskipun pengujian stabilitas jangka panjang (seperti sehari-hari) belum dilakukan, dalam serangkaian uji coba koneksi dan simulasi penggunaan berulang (menghidupkan dan mematikan lampu berkali-kali), mikrokontroler ESP32 berhasil terhubung dan mempertahankan koneksi yang stabil ke jaringan Wi-Fi lokal. Stabilitas ini mendukung interpretasi bahwa sistem akan beroperasi secara andal dalam lingkungan jaringan rumah standar, dan responsif kapan pun pengguna berada dalam jangkauan Wi-Fi.

D. PENUTUP

Berdasarkan hasil dan pembahasan, sistem yang dikembangkan telah berhasil mencapai tujuannya yaitu menyediakan solusi kendali lampu yang efektif dan mandiri. Keberhasilan ini ditunjukkan melalui dua temuan kunci: pertama, fungsionalitas kendali ON/OFF berjalan sempurna, dan kedua, sistem menunjukkan kinerja yang unggul dengan waktu respons rata-rata yang sangat rendah, yaitu hanya sekitar 100-150 milidetik (ms). Kecepatan ini membuktikan bahwa arsitektur *Embedded Web Server* yang bekerja di jaringan lokal jauh lebih responsif dibandingkan solusi berbasis *cloud*.

Permasalahan utama terkait ketergantungan internet dan risiko privasi pada sistem IoT tradisional berhasil diatasi. Penerapan *Web Server Lokal* secara signifikan memengaruhi aspek keamanan dan kemandirian, karena akses untuk kontrol dan data log riwayat aksi secara inheren terbatas hanya pada perangkat dalam jaringan Wi-Fi lokal hal ini secara efektif mengurangi risiko kebocoran data ke pihak ketiga. Sistem ini menyajikan solusi yang sangat responsif, stabil, dan aman yang memenuhi kebutuhan pengguna yang memprioritaskan privasi dan kecepatan akses di lingkungan rumah. Faktor pendukung utama adalah ketersediaan *System on Chip (SoC)* seperti ESP32 yang berkemampuan Wi-Fi dan *embedded web server*, serta tersedianya library pemrograman

yang memadai. Sementara itu, faktor penghambat yang teridentifikasi meliputi keterbatasan memori pada mikrokontroler untuk logging data jangka panjang dan perlunya pemeliharaan ketersediaan Power Supply 5V yang stabil untuk operasional 24 jam.

Untuk pengembangan selanjutnya, sistem kendali lampu berbasis IoT dengan web *server* lokal ini dapat ditingkatkan dengan menambahkan sensor cahaya otomatis agar lampu dapat menyala atau mati berdasarkan kondisi pencahayaan ruangan, sehingga penggunaan energi lebih efisien. Integrasi fitur pemantauan konsumsi daya juga dapat membantu pengguna memonitor penggunaan listrik secara *real-time*. Selain itu, pengembangan antarmuka web yang lebih responsif dan *user-friendly*, serta penambahan fitur kontrol suara dapat meningkatkan kenyamanan pengguna. Untuk aplikasi di lingkungan yang lebih luas, sistem dapat diperluas dengan kemampuan multi-ruangan dan integrasi protokol IoT lainnya agar lebih fleksibel dan *scalable*.

E. DAFTAR PUSTAKA

- Abdussomad, M. K., Rofiqi, M., Salehuddin, M., & Maulana, M. M. (2022). Rancang Bangun Sistem Kendali Lampu Otomatis Berbasis Internet of Things (IoT) Menggunakan Aplikasi Blynk 2.0. *Prosiding Seminar Nasional Hi-Tech (Humanity, Health, Technology)*, 1(1), 323–338.
<https://ejournal.unuja.ac.id/index.php/hitech/article/view/4996>
- Nugroho, P. A. (2021). Kontrol Lampu Gedung Melalui Wifi ESP8266 Dengan Web Server Lokal. *JEIS: Jurnal Elektro Dan Informatika Swadharma*, 1(2), 1–11.
<https://doi.org/10.56486/jeis.vol1no2.95>
- Rahmah, U., Sabara, E., & Dahr, N. (2020). Pengembangan Trainer Microcontroller Berbasis Internet of Things (IoT) Pada Mata Kuliah Microcontroller dan Interface di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Makassar. *JETC: Jurnal Elektronika Telekomunikasi & Computer*, 5(2), 1–9.
<https://ojs.unm.ac.id/JETC/article/view/23435>
- Saputra, T., & Surapati, U. (2024). Analisis Efektivitas Sistem Kendali Otomatis PJU Berbasis IoT Menggunakan Mikrokontroler ESP32 dengan Metode Regresi Linier. *JIMIK: Jurnal Indonesia Manajemen Informatika Dan Komunikasi*, 5(3), 2582–2595.
<https://doi.org/10.35870/jimik.v5i3.932>
- Sawidin, S. K., Patabo, M. D., Tuwongkesong, S., Waroh, A. P. Y., Kereh, T. M., & Lapon, J. J. (2024). Design and Building of a Room Light Control System based on IoT Arduino Cloud. *JAMBURA: Journal of Electrical and Electronics Engineering*, 6(2), 227–233.
<https://doi.org/10.37905/jjee.v6i2.26131>
- Setiyani, L., Suhada, K., & Yulindawati. (2019). Perancangan dan Implementasi IoT (Internet of Things) Pada Smarthome Menggunakan Raspberry Pi Berbasis Android. *Simetris: Jurnal Teknik Industri, Mesin, Elektro Dan Ilmu Komputer*, 10(2), 459–466.
<https://doi.org/10.24176/simet.v10i2.3059>
- Shiddiqi, A. M., Ijtihadie, R. M., Ahmad, T., Wibisono, W., Anggoro, R., & Santos, B. J. (2020). Penggunaan Internet dan Teknologi IoT untuk Meningkatkan Kualitas Pendidikan. *SEWAGATI: Jurnal Pengabdian Masyarakat*, 4(3), 235–240.
<https://doi.org/10.12962/j26139960.v4i3.369>
- Wicaksono, M. F., & Rahmatya, M. D. (2020). Implementasi Arduino dan ESP32 CAM untuk Smart Home. *JATI: Jurnal Teknologi Dan Informasi*, 10(1), 40–51.
<https://doi.org/10.34010/jati.v10i1.2836>