JURNAL TAMPIASIH
INSTITUT TEKNOLOGI DAN KESEHATAN ASPIRASI
E-ISSN : 2986 - 8742

# Perancangan dan Implementasi Sistem Internet of Things Berbasis ESP32 untuk Monitoring dan Pengendalian Perangkat Secara Real-Time

**Abdul Azis[1], Gala Al Kausar[2]**
[1]Program Studi Teknologi Informasi, Institut Teknologi dan Kesehatan Aspirasi, Indonesia
[2]Program Studi Pendidikan Informatika, STKIP Harapan Bima, Indonesia

**Article Information**

**Article history:**
Received: 12/12/2025

Approved: 31/12/2025

**Kata Kunci:**
Internet of Things, ESP32, MQTT, monitoring real-time, pengendalian perangkat.

**Corresponding author:**
abdulaziz@itka.ac.id

**ABSTRAK**

**Latar Belakang**: Pemantauan kondisi lingkungan dan pengendalian perangkat secara jarak jauh membutuhkan platform Internet of Things (IoT) yang responsif, mudah dikembangkan, dan mampu mempertahankan konsistensi status perangkat. Penelitian ini bertujuan untuk merancang dan mengimplementasikan arsitektur IoT berbasis ESP32 untuk pemantauan data sensor dan pengendalian beban listrik secara dua arah melalui jaringan Wi-Fi. Sistem menggunakan ESP32 sebagai pengendali utama, sensor DHT22 dan LDR sebagai sumber data, modul relay dua kanal sebagai aktuator, protokol MQTT sebagai media pertukaran pesan, serta dashboard web sebagai antarmuka pengguna. Metode penelitian menggunakan pendekatan prototipe yang meliputi analisis kebutuhan, perancangan perangkat keras dan topik MQTT, implementasi logika akuisisi serta kendali, dan verifikasi fungsional berbasis simulasi state machine. Telemetri dikemas dalam format JSON, sedangkan perintah kendali dipisahkan pada topik command dan dikonfirmasi melalui pesan status/acknowledgement. Hasil verifikasi menunjukkan sepuluh skenario logika utama—mulai dari inisialisasi fail-safe, koneksi Wi-Fi dan MQTT, publikasi telemetri, kendali relay, penolakan perintah yang tidak valid, hingga pemulihan koneksi—berjalan sesuai dengan keluaran yang diharapkan. Arsitektur ini menekankan komunikasi dua arah, pemisahan kanal data dan perintah, serta kondisi aman saat koneksi terputus. Penelitian ini menghasilkan rancangan implementatif yang dapat dijadikan dasar pengembangan sistem smart home, laboratorium, dan monitoring perangkat lainnya. Pengukuran latensi jaringan dan akurasi sensor fisik perlu dilakukan pada implementasi lapangan sebagai tahap validasi lanjutan.

**© 2025 TAMPIASIH**

## PENDAHULUAN

Internet of Things (IoT) berkembang sebagai pendekatan komputasi yang menghubungkan perangkat fisik, sensor, aktuator, jaringan, dan aplikasi sehingga data kondisi lapangan dapat dikumpulkan sekaligus digunakan untuk menjalankan tindakan pengendalian. Dalam sistem monitoring konvensional, pengguna sering harus berada di lokasi perangkat atau bergantung pada

pencatatan manual yang tidak kontinu. Integrasi mikrokontroler berkemampuan jaringan mengubah pola tersebut menjadi sistem terhubung yang dapat mengirim telemetri, menerima perintah, dan menyimpan status perangkat secara berkesinambungan. D'Ortona et al. (2022) menunjukkan bahwa arsitektur end-to-end berbasis MQTT dapat mengintegrasikan sensor, jaringan, broker, dan aplikasi dalam satu aliran data IoT. Sementara itu, Yasin et al. (2022) menekankan pentingnya pemilihan sensor, integrasi data, dan antarmuka pemantauan ketika perangkat komersial berbiaya rendah digunakan untuk membangun sistem monitoring yang dapat direplikasi.

ESP32 menjadi salah satu platform yang relevan untuk kebutuhan tersebut karena mengintegrasikan prosesor 32-bit, Wi-Fi 2,4 GHz, Bluetooth, GPIO, ADC, antarmuka serial, dan mekanisme manajemen daya dalam satu system-on-chip. Dokumentasi Espressif Systems (2026) menunjukkan bahwa keluarga ESP32 dirancang untuk aplikasi IoT, otomasi, sensor hub, data logger, dan perangkat rumah cerdas. Keunggulan integrasi ini mengurangi kebutuhan modul komunikasi tambahan dan menyederhanakan rancangan prototipe. Evaluasi Espinosa-Gavira et al. (2024) juga memperlihatkan bahwa ESP32 mampu digunakan pada jaringan sensor tersinkronisasi dengan kinerja akuisisi dan jaringan yang stabil, meskipun perhatian khusus tetap diperlukan pada penggunaan ADC internal untuk sensor analog berpresisi tinggi.

Pada sisi komunikasi, kebutuhan monitoring dan kontrol real-time tidak hanya menuntut konektivitas, tetapi juga mekanisme pertukaran pesan yang ringan. MQTT menggunakan pola publish/subscribe yang memisahkan pengirim dan penerima melalui broker, sehingga perangkat tidak perlu mengetahui alamat aplikasi tujuan secara langsung. Standar MQTT mendefinisikan transport pesan yang ringan dan sesuai untuk lingkungan dengan sumber daya serta bandwidth terbatas (OASIS, 2019). Manowska et al. (2023) memperlihatkan pemanfaatan MQTT pada sistem pengukuran, monitoring, dan kontrol, sedangkan Alshammari (2023) memanfaatkannya untuk pengiriman sinyal pemantauan jarak jauh dengan kebutuhan latensi rendah. Karakteristik tersebut membuat MQTT sesuai untuk ESP32 yang harus mengirim telemetri sensor dan pada saat yang sama menerima perintah aktuator.

Penelitian terdahulu telah menerapkan ESP32 dan IoT pada konteks yang beragam. Ratnasari et al. (2022) mengembangkan pemantauan daya panel surya menggunakan ESP32 dan Telegram. Abu Sneineh dan Shabaneh (2023) mengintegrasikan ESP32, beberapa sensor, dan aplikasi Blynk untuk monitoring serta pengendalian hidroponik. Sadeeq dan Zeebaree (2023) menggunakan beberapa ESP32 pada sistem manajemen energi terdistribusi yang memadukan fungsi monitoring dan controlling. Pada konteks manufaktur, Zakaria et al. (2023) menggunakan ESP32 untuk pemantauan konsumsi energi pada proses fused deposition modelling. Temuan-temuan tersebut mengindikasikan bahwa ESP32 fleksibel untuk berbagai jenis sensor dan dapat ditempatkan sebagai node pengendali pada arsitektur IoT.

Di Indonesia, penerapan serupa juga berkembang. Rahmawati et al. (2024) merancang monitoring kualitas daya secara real-time berbasis IoT, sedangkan Lianawati et al. (2024) menerapkan ESP32 pada sistem monitoring dan controlling smart waste. Penelitian yang lebih baru oleh Laagu et al. (2026) mengintegrasikan ESP32, sensor energi, relay, MQTT Dash, dan Telegram untuk monitoring sekaligus pengendalian beban rumah. Literatur tersebut memperkuat bahwa kombinasi ESP32, sensor, relay, dan aplikasi jaringan dapat digunakan untuk membangun sistem yang praktis. Namun, banyak implementasi masih berfokus pada satu domain aplikasi atau satu antarmuka tertentu sehingga struktur pertukaran pesan, status aktuator, dan perilaku saat koneksi terputus tidak selalu dijelaskan sebagai rancangan generik yang dapat digunakan kembali.

Kesenjangan yang diangkat dalam penelitian ini adalah kebutuhan rancangan IoT yang sederhana tetapi memiliki aliran komunikasi dua arah yang eksplisit, pemisahan topik telemetri dan perintah, konfirmasi status setelah aktuasi, serta mekanisme fail-safe ketika koneksi terputus. Pemisahan ini penting karena kondisi yang ditampilkan pada dashboard harus merepresentasikan

keadaan perangkat, bukan hanya perintah terakhir yang dikirimkan pengguna. Serepas et al. (2025) menunjukkan bahwa ESP32 dapat berfungsi sebagai gateway IoT ringan untuk smart home dengan tuntutan sumber daya yang relatif rendah. Di sisi protokol, Tsaqief dan Sutopo (2025) melaporkan bahwa MQTT dapat memberikan latensi rendah dan jitter kecil pada lingkungan LAN yang terkontrol, meskipun hasil tersebut tidak dapat langsung digeneralisasi ke semua jaringan. Oleh karena itu, desain sistem perlu memisahkan aspek arsitektur fungsional dari klaim performa jaringan yang harus diukur pada lingkungan implementasi nyata.

Penelitian ini merancang sistem representatif yang terdiri atas sensor DHT22 untuk suhu dan kelembapan, LDR untuk intensitas cahaya, ESP32 sebagai pengendali utama, relay dua kanal untuk mengendalikan lampu dan kipas, broker MQTT sebagai perantara pesan, serta dashboard web sebagai antarmuka monitoring dan kontrol. Data sensor dikemas dalam JSON dan dipublikasikan secara periodik, sedangkan perintah relay dikirim melalui topik command terpisah. Setelah perintah dieksekusi, ESP32 menerbitkan status aktuator sebagai acknowledgement. Ketika Wi-Fi atau broker tidak tersedia, keluaran relay diarahkan ke kondisi aman sesuai kebijakan fail-safe yang ditentukan dalam rancangan.

Tujuan penelitian adalah menghasilkan rancangan dan implementasi logika sistem IoT berbasis ESP32 yang mendukung monitoring dan pengendalian perangkat secara real-time pada tingkat aplikasi, serta memverifikasi konsistensi alur fungsi utama melalui pengujian state machine. Kontribusi penelitian terletak pada arsitektur generik yang dapat direplikasi untuk smart home, laboratorium, ruang kelas, atau monitoring perangkat sederhana tanpa bergantung pada satu platform cloud tertentu. Penelitian ini tidak mengklaim nilai latensi jaringan maupun akurasi metrologi sensor fisik karena kedua parameter tersebut memerlukan perangkat dan kondisi uji lapangan; fokus evaluasi pada naskah ini adalah integritas alur komunikasi dan fungsi kendali sebagai dasar untuk implementasi fisik berikutnya.

## METODE PENELITIAN

Penelitian menggunakan pendekatan prototipe dengan tahapan analisis kebutuhan, perancangan arsitektur, pemilihan komponen, penyusunan logika perangkat lunak, dan verifikasi fungsional. Pendekatan ini dipilih karena tujuan utama penelitian adalah menghasilkan artefak sistem yang dapat diuji dan dikembangkan secara iteratif. Kebutuhan fungsional dirumuskan menjadi lima fungsi utama, yaitu membaca data sensor, mengirim telemetri, menerima perintah kendali, mengubah kondisi relay, dan mengirim kembali status perangkat. Kebutuhan nonfungsional mencakup penggunaan Wi-Fi lokal, format data yang sederhana, mekanisme pemulihan koneksi, dan kondisi fail-safe ketika komunikasi terputus.

Perangkat keras utama terdiri atas ESP32 DevKit sebagai unit pemrosesan dan komunikasi, sensor DHT22 sebagai sumber suhu serta kelembapan, LDR dengan rangkaian pembagi tegangan sebagai sumber intensitas cahaya relatif, dan modul relay dua kanal untuk mengendalikan dua beban representatif berupa lampu dan kipas. ESP32 dipilih karena menyediakan konektivitas Wi-Fi, GPIO, ADC, dan kemampuan pemrosesan yang mencukupi untuk akuisisi sensor serta pertukaran pesan IoT dalam satu papan pengembangan (Espressif Systems, 2026). Untuk implementasi fisik, relay yang digunakan harus memiliki isolasi dan rating arus/tegangan sesuai beban; pada penelitian ini, pembahasan difokuskan pada sisi logika kendali tegangan rendah dan tidak mencakup desain instalasi listrik tegangan PLN.

Lapisan komunikasi menggunakan MQTT dengan pola publish/subscribe. Broker dapat ditempatkan pada komputer lokal, server, atau layanan cloud yang kompatibel. Tiga kelompok topik didefinisikan: iot/esp32/telemetry untuk data sensor, iot/esp32/control/relay1 dan iot/esp32/control/relay2 untuk perintah kendali, serta iot/esp32/status untuk konfirmasi kondisi aktuator. Pemisahan topik memudahkan pengaturan hak akses dan mengurangi ambiguitas antara data pengukuran dan pesan kendali. Telemetri dikirim dalam format JSON yang memuat suhu,

kelembapan, intensitas cahaya, status relay, dan penanda waktu pada sisi aplikasi. Mekanisme ini mengikuti prinsip MQTT yang ringan dan terpisah antara publisher serta subscriber (OASIS, 2019; Manowska et al., 2023).

Perangkat lunak ESP32 disusun dalam siklus nonblocking agar pembacaan sensor dan pengelolaan komunikasi tidak bergantung pada jeda panjang. Setelah boot, relay diinisialisasi pada kondisi OFF sebagai keadaan aman, kemudian perangkat melakukan koneksi Wi-Fi dan broker MQTT. Ketika koneksi broker berhasil, ESP32 melakukan subscribe pada topik kendali. Data sensor dibaca pada interval periodik dan dipublikasikan sebagai telemetri. Callback MQTT memeriksa payload perintah ON atau OFF, mengubah GPIO relay, lalu memublikasikan status aktual. Perintah dengan format tidak dikenali ditolak sehingga tidak mengubah output. Jika koneksi jaringan terputus melewati kondisi yang ditentukan, relay kembali ke kondisi aman dan proses reconnect dijalankan.

Antarmuka dashboard dirancang untuk menampilkan kartu suhu, kelembapan, intensitas cahaya, status koneksi, serta status setiap relay. Tombol kendali pada dashboard tidak langsung mengubah indikator status; tombol hanya menerbitkan command, sedangkan indikator diperbarui setelah pesan status dari ESP32 diterima. Pola acknowledgement tersebut digunakan agar antarmuka merefleksikan keadaan aktual di sisi perangkat. Esposito et al. (2023) menunjukkan bahwa framework otomasi berbasis MQTT dapat memisahkan aplikasi pengendali dari objek cerdas melalui pertukaran pesan, sedangkan Serepas et al. (2025) memperlihatkan kelayakan ESP32 sebagai komponen gateway ringan pada sistem smart home.

Verifikasi dilakukan pada level logika menggunakan simulasi state machine yang merepresentasikan kondisi boot, koneksi Wi-Fi, koneksi dan subscribe MQTT, publikasi telemetri, perubahan relay, validasi perintah, kehilangan koneksi, fail-safe, serta reconnect. Sepuluh skenario diuji terhadap keluaran yang diharapkan. Pengujian ini memvalidasi konsistensi alur perangkat lunak, bukan performa radio atau metrologi sensor. Karena tidak tersedia log pengukuran perangkat fisik dari pengguna, naskah secara sengaja tidak membuat klaim numerik mengenai latency, packet loss, error sensor, atau konsumsi daya. Parameter tersebut direkomendasikan sebagai pengujian lanjutan dengan osiloskop, timestamp broker, alat ukur referensi, dan sejumlah pengulangan yang memadai.

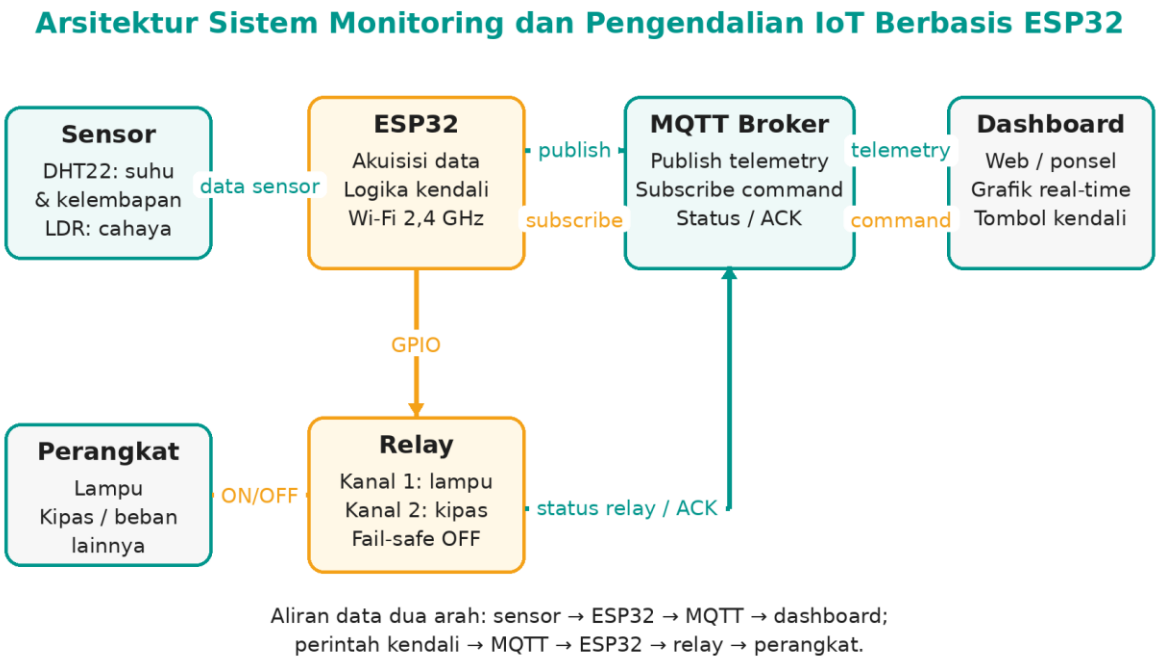
Tabel 1. Komponen dan fungsi pada prototipe sistem

| Komponen | Fungsi | Antarmuka | Keterangan |
|---|---|---|---|
| ESP32 DevKit | Pemrosesan, Wi-Fi, kendali GPIO | Wi-Fi, GPIO, ADC | Node utama IoT |
| DHT22 | Mengukur suhu dan kelembapan | Digital | Telemetri lingkungan |
| LDR | Membaca intensitas cahaya relatif | ADC | Nilai analog dikonversi relatif |
| Relay 2 kanal | Menghubung/memutus beban | GPIO digital | Kanal lampu dan kipas |
| MQTT Broker | Meneruskan pesan publish/subscribe | TCP/IP | Dapat lokal atau cloud |
| Dashboard web | Visualisasi dan kendali pengguna | MQTT/Web | Status diperbarui dari ACK |

Sumber: rancangan penelitian, 2026.

**HASIL**

Hasil utama penelitian berupa arsitektur sistem dua arah yang menempatkan ESP32 sebagai node tepi, broker MQTT sebagai perantara pertukaran pesan, dan dashboard sebagai antarmuka pengguna. Jalur monitoring dimulai dari pembacaan DHT22 dan LDR, pengolahan pada ESP32, pembentukan payload JSON, publikasi ke broker, lalu penyajian pada dashboard. Jalur kontrol berjalan pada arah sebaliknya: dashboard menerbitkan perintah, broker meneruskannya kepada ESP32 yang telah melakukan subscribe, ESP32 mengubah status GPIO relay, kemudian menerbitkan acknowledgement. Arsitektur tersebut ditunjukkan pada Gambar 1.




Gambar 1. Arsitektur sistem monitoring dan pengendalian IoT berbasis ESP32.

Pemisahan topik MQTT menghasilkan struktur komunikasi yang mudah ditelusuri. Telemetri hanya membawa data observasi dan status, sedangkan kanal control hanya menerima instruksi yang telah ditentukan. Pola ini mencegah dashboard menganggap perintah sebagai bukti bahwa aktuator benar-benar berubah. Status relay dipublikasikan kembali setelah eksekusi command, sehingga aplikasi dapat menyinkronkan tampilan dengan keadaan node. Struktur topik dan payload yang digunakan ditunjukkan pada Tabel 2.

Tabel 2. Struktur topik MQTT dan fungsi pesan

| Topik | Arah | Payload contoh | Fungsi |
|---|---|---|---|
| iot/esp32/telemetry | ESP32 → Broker | {"temp":28.4,"hum":72,"light":615} | Mengirim data sensor dan status ringkas |
| iot/esp32/control/relay1 | Dashboard → ESP32 | ON / OFF | Mengendalikan relay kanal 1 |
| iot/esp32/control/relay2 | Dashboard → ESP32 | ON / OFF | Mengendalikan relay kanal 2 |
| iot/esp32/status | ESP32 → Dashboard | {"relay1":"ON","relay2":"OFF"} | Konfirmasi keadaan aktual/ACK |

Sumber: rancangan penelitian, 2026.

Verifikasi state machine menghasilkan keberhasilan pada seluruh sepuluh skenario logika yang diuji. Keberhasilan dimaknai sebagai kesesuaian keluaran simulasi dengan aturan rancangan,

bukan sebagai bukti kinerja jaringan fisik. Pada saat boot, kedua relay berada pada kondisi OFF; setelah Wi-Fi aktif, broker dapat dihubungkan dan topik control disubscribe. Telemetri hanya dapat dipublikasikan ketika sesi MQTT aktif. Perintah ON dan OFF mengubah kondisi relay yang sesuai, sedangkan payload yang tidak dikenali ditolak. Ketika koneksi Wi-Fi diputus, sesi MQTT dinonaktifkan dan relay kembali OFF. Setelah koneksi dipulihkan, sesi MQTT dan subscription dapat dibentuk kembali.

Tabel 3. Hasil verifikasi fungsional state machine

| No. | Skenario | Keluaran yang diharapkan | Hasil |
|---|---|---|---|
| 1 | Boot sistem | Relay 1 dan Relay 2 = OFF | Sesuai |
| 2 | Koneksi Wi-Fi | Status Wi-Fi aktif | Sesuai |
| 3 | Koneksi MQTT | Session aktif dan subscribe control | Sesuai |
| 4 | Publikasi telemetri | Payload sensor diterima fungsi publish | Sesuai |
| 5 | Command Relay 1 ON | Relay 1 berubah ON dan ACK dicatat | Sesuai |
| 6 | Command Relay 1 OFF | Relay 1 berubah OFF | Sesuai |
| 7 | Command Relay 2 ON | Relay 2 berubah ON | Sesuai |
| 8 | Payload tidak valid | Perintah ditolak; output tidak berubah | Sesuai |
| 9 | Koneksi terputus | MQTT nonaktif dan relay fail-safe OFF | Sesuai |
| 10 | Reconnect | Wi-Fi, MQTT, dan subscription aktif kembali | Sesuai |

Sumber: hasil simulasi logika penelitian, 2026.

**PEMBAHASAN**

Arsitektur yang dihasilkan menunjukkan bahwa satu ESP32 dapat memegang tiga peran sekaligus, yaitu akuisisi data sensor, eksekusi logika kendali, dan komunikasi jaringan. Integrasi tersebut mengurangi jumlah komponen komunikasi dan cocok untuk prototipe berbiaya relatif rendah. Hasil ini sejalan dengan karakteristik perangkat yang dilaporkan Espinosa-Gavira et al. (2024), yang menunjukkan bahwa ESP32 mampu menangani akuisisi sensor dan komunikasi pada jaringan sinkron. Namun, penggunaan LDR melalui ADC internal perlu dipahami sebagai pengukuran relatif. Untuk aplikasi yang memerlukan ketelitian analog tinggi, kalibrasi dan penggunaan ADC eksternal dapat dipertimbangkan karena karakteristik noise dan non-linearitas ADC internal ESP32 telah dilaporkan pada penelitian tersebut.

Penerapan MQTT memberi keuntungan utama berupa decoupling antara perangkat dan aplikasi. ESP32 tidak perlu mengetahui lokasi setiap dashboard, dan dashboard tidak perlu berkomunikasi langsung dengan alamat perangkat selama keduanya terhubung pada broker dan topik yang sama. D'Ortona et al. (2022) menunjukkan pendekatan end-to-end berbasis MQTT untuk integrasi IoT, sedangkan Manowska et al. (2023) memperlihatkan relevansinya pada sistem pengukuran dan kontrol. Pada penelitian ini, pemisahan telemetry, command, dan status menambah keterlacakan aliran pesan. Struktur tersebut juga memudahkan pengembangan selanjutnya, misalnya penambahan perangkat baru dengan namespace topik yang berbeda tanpa mengubah konsep dasar sistem.

Istilah real-time pada sistem ini digunakan dalam pengertian operasional IoT, yaitu data diperbarui dan perintah diproses segera setelah pesan diterima tanpa siklus batch manual. Sistem

ini bukan hard real-time system karena tidak menetapkan deadline deterministik pada level milidetik. Performa aktual dipengaruhi kualitas Wi-Fi, lokasi broker, kepadatan jaringan, ukuran payload, Quality of Service, dan beban server. Tsaqief dan Sutopo (2025) menemukan performa MQTT yang sangat rendah latensi pada lingkungan LAN terkontrol, namun angka tersebut tidak boleh dipindahkan langsung ke rancangan ini tanpa pengukuran yang setara. Dengan demikian, klaim real-time pada penelitian ini merujuk pada arsitektur event-driven dan pembaruan langsung pada tingkat aplikasi, bukan jaminan deadline tertentu.

Mekanisme acknowledgement menjadi unsur penting karena kontrol jarak jauh memiliki risiko ketidaksinkronan antara antarmuka dan perangkat. Jika dashboard langsung mengubah indikator menjadi ON saat tombol ditekan, pengguna dapat mengira beban telah aktif meskipun pesan belum diterima ESP32 atau relay gagal berubah. Dengan menerbitkan status aktual setelah eksekusi, dashboard memiliki sumber kebenaran yang berasal dari node. Prinsip ini relevan dengan sistem otomasi berbasis pesan yang memisahkan command dan state. Framework smart home yang dibahas Esposito et al. (2023) juga menempatkan pertukaran pesan sebagai penghubung antara aplikasi dan objek cerdas, sementara Laagu et al. (2026) memperlihatkan integrasi monitoring serta relay control melalui MQTT pada lingkungan smart home.

Kondisi fail-safe OFF pada saat koneksi terputus dipilih untuk prototipe representatif karena lampu dan kipas dianggap sebagai beban yang aman bila dimatikan. Kebijakan ini tidak boleh digeneralisasi untuk seluruh aplikasi. Pada sistem tertentu, seperti ventilasi kritis, pompa proses, atau perangkat medis, kondisi aman mungkin berbeda dan harus ditetapkan melalui analisis risiko. Dengan kata lain, fail-safe merupakan kebijakan aplikasi, bukan sekadar fitur kode. Keunggulan rancangan adalah kebijakan tersebut ditempatkan secara eksplisit pada state machine sehingga perilaku saat kehilangan jaringan tidak dibiarkan tidak terdefinisi.

Dari sisi skalabilitas, penambahan node dapat dilakukan dengan memperluas identitas topik, misalnya iot/ruang1/esp32-01/telemetry atau iot/lab/esp32-02/status. Pendekatan ini lebih terstruktur dibanding menggunakan satu topik global untuk seluruh perangkat. Sadeeq dan Zeebaree (2023) menunjukkan bahwa sistem berbasis beberapa ESP32 dapat diterapkan pada lingkungan terdistribusi untuk monitoring dan controlling energi. Serepas et al. (2025) juga menunjukkan kemampuan ESP32 sebagai gateway ringan pada ekosistem smart home. Walaupun demikian, semakin banyak node akan meningkatkan kebutuhan manajemen identitas, autentikasi, penyimpanan data, dan observabilitas broker sehingga arsitektur produksi perlu menambahkan perangkat lunak backend yang lebih terstruktur.

Aspek keamanan harus menjadi bagian pengembangan lanjutan. MQTT yang digunakan pada prototipe tidak seharusnya dibiarkan terbuka pada jaringan publik tanpa autentikasi dan enkripsi. Implementasi produksi perlu menggunakan kredensial unik, Access Control List per topik, TLS bila sumber daya dan broker mendukung, serta pemisahan jaringan perangkat IoT dari jaringan pengguna umum. OASIS (2019) menyediakan mekanisme dan pertimbangan keamanan pada standar MQTT, sedangkan kemampuan keamanan ESP32 mencakup fitur seperti secure boot, flash encryption, dan akselerasi kriptografi yang dapat dipertimbangkan sesuai kebutuhan (Espressif Systems, 2026). Pengamanan juga perlu mencakup penyimpanan kredensial, mekanisme pembaruan firmware, dan pembatasan akses dashboard.

Dibanding penelitian aplikasi-spesifik, rancangan ini sengaja mempertahankan perangkat sensor dan aktuator yang sederhana untuk menonjolkan pola arsitektur. Abu Sneineh dan Shabaneh (2023) memanfaatkan ESP32 untuk sistem hidroponik, Ratnasari et al. (2022) untuk monitoring daya panel surya, Rahmawati et al. (2024) untuk monitoring kualitas daya, dan Lianawati et al. (2024) untuk smart waste. Perbedaan domain tersebut menunjukkan bahwa logika dasar akuisisi–publikasi–visualisasi–command–aktuasi bersifat dapat digunakan kembali. Karena itu, kontribusi

praktis penelitian ini adalah menyusun pola umum yang dapat menjadi titik awal pengembangan, sementara sensor, aktuator, batas kontrol, dan dashboard dapat diganti sesuai konteks.

Keterbatasan penelitian terletak pada validasi yang masih berada pada level logika dan desain. Hasil simulasi sepuluh skenario memastikan state transition konsisten, tetapi belum membuktikan reliability jaringan, packet delivery ratio, jitter, konsumsi daya, atau lifetime perangkat pada operasi jangka panjang. Penelitian Yasin et al. (2022), Zakaria et al. (2023), dan studi-studi monitoring lainnya menunjukkan bahwa pengujian perangkat fisik diperlukan untuk memperoleh karakteristik sensor dan komunikasi yang sebenarnya. Tahap lanjutan sebaiknya menguji minimal beberapa kondisi RSSI Wi-Fi, lokasi broker lokal dan cloud, QoS MQTT, interval publish, restart perangkat, kehilangan daya, serta kalibrasi sensor terhadap alat referensi. Hasil tersebut kemudian dapat digunakan untuk mengubah rancangan ini dari prototipe fungsional menjadi sistem terukur yang siap digunakan pada lingkungan operasional.

## KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan logika sistem Internet of Things berbasis ESP32 untuk monitoring suhu, kelembapan, dan intensitas cahaya serta pengendalian dua perangkat melalui relay secara real-time. Sistem menggunakan komunikasi dua arah berbasis MQTT dengan pemisahan topik telemetri, perintah, dan status, sehingga proses pertukaran data lebih terstruktur dan kondisi aktuator dapat dikonfirmasi kembali melalui dashboard. Hasil verifikasi terhadap sepuluh skenario state machine menunjukkan bahwa fungsi inisialisasi fail-safe, konektivitas, publikasi data, kendali relay, validasi perintah, penanganan kehilangan koneksi, dan proses reconnect telah berjalan sesuai dengan rancangan.

Berdasarkan hasil tersebut, rancangan sistem dinilai layak digunakan sebagai dasar pengembangan prototipe IoT berbasis ESP32. Namun, penelitian ini belum mengukur latensi, packet loss, akurasi sensor, dan reliabilitas perangkat fisik pada lingkungan nyata. Oleh karena itu, penelitian selanjutnya perlu melakukan implementasi perangkat secara langsung, pengujian kinerja komunikasi, kalibrasi sensor, uji operasi jangka panjang, penerapan keamanan MQTT seperti TLS dan ACL, serta penyimpanan data historis agar sistem dapat dikembangkan menjadi platform monitoring dan pengendalian yang lebih aman, terukur, dan siap diterapkan pada berbagai kebutuhan IoT.

## UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada semua pihak yang mendukung penyusunan serta pengembangan prototipe.

## DAFTAR PUSTAKA

Abu Sneineh, A., & Shabaneh, A. A. A. (2023). Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things. MethodsX, 11, 102401. https://doi.org/10.1016/j.mex.2023.102401

Alshammari, H. H. (2023). The internet of things healthcare monitoring system based on MQTT protocol. Alexandria Engineering Journal, 69, 275–287. https://doi.org/10.1016/j.aej.2023.01.065

D'Ortona, C., Tarchi, D., & Raffaelli, C. (2022). Open-source MQTT-based end-to-end IoT system for smart city scenarios. Future Internet, 14(2), 57. https://doi.org/10.3390/fi14020057

Espinosa-Gavira, M. J., Agüera-Pérez, A., Palomares-Salas, J. C., Sierra-Fernández, J. M., Remigio-Carmona, P., & González-de-la-Rosa, J. J. (2024). Characterization and performance evaluation of ESP32 for real-time synchronized sensor networks. Procedia Computer Science, 237, 261–268. https://doi.org/10.1016/j.procs.2024.05.104

Espressif Systems. (2026). ESP32 Series Datasheet (Version 5.3). Espressif Systems.

Esposito, M., Belli, A., Palma, L., & Pierleoni, P. (2023). Design and implementation of a framework for smart home automation based on cellular IoT, MQTT, and serverless functions. Sensors, 23(9), 4459. https://doi.org/10.3390/s23094459

Laagu, M. A., Nendriyo, B. R. J., Anshori, I. F., Wicaksono, I., & Chaidir, A. R. (2026). Design and implementation of an ESP32-based smart home electricity monitoring and control module using MQTT Dash and Telegram integration. SMATIKA Jurnal, 16(2), 470–480. https://doi.org/10.32664/smatika.v16i02.2321

Lianawati, Y., Mahendra, C., Sugianto, G. M., Mendrofa, S. J., Setiani, A. L., & Baraga, B. Y. (2024). Sistem monitoring dan controlling “Smart Waste” berbasis Internet of Things menggunakan modul ESP 32. Journal of Telecommunication Electronics and Control Engineering, 6(2), 163–175. https://doi.org/10.20895/jtece.v6i2.1400

Manowska, A., Wycisk, A., Nowrot, A., & Pielot, J. (2023). The use of the MQTT protocol in measurement, monitoring and control systems as part of the implementation of energy management systems. Electronics, 12(1), 17. https://doi.org/10.3390/electronics12010017

OASIS. (2019). MQTT Version 5.0. OASIS Standard.

Rahmawati, Y., Ramadhan, A. N., & Kadarina, T. M. (2024). Perancangan sistem monitoring kualitas daya pada panel saklar tegangan rendah secara real-time berbasis Internet of Things (IoT). Journal of Telecommunication Electronics and Control Engineering, 6(1), 60–73. https://doi.org/10.20895/jtece.v6i1.1327

Ratnasari, D. A., Suprianto, B., & Baskoro, F. (2022). Monitoring daya listrik pada panel surya berbasis Internet of Things (IoT) menggunakan aplikasi Telegram. Indonesian Journal of Engineering and Technology, 5(1), 1–10. https://doi.org/10.26740/inajet.v5n1.p1-10

Sadeeq, M. A. M., & Zeebaree, S. R. M. (2023). Design and implementation of an energy management system based on distributed IoT. Computers & Electrical Engineering, 109, 108775. https://doi.org/10.1016/j.compeleceng.2023.108775

Serepas, F., Papias, I., Christakis, K., Dimitropoulos, N., & Marinakis, V. (2025). Lightweight embedded IoT gateway for smart homes based on an ESP32 microcontroller. Computers, 14(9), 391. https://doi.org/10.3390/computers14090391

Tsaqief, M. F., & Sutopo, J. (2025). Comparative performance analysis between the MQTT and WebSocket protocols. bit-Tech, 8(2), 2227–2237. https://doi.org/10.32877/bt.v8i2.3223

Yasin, A., Delaney, J., Cheng, C.-T., & Pang, T. Y. (2022). The design and implementation of an IoT sensor-based indoor air quality monitoring system using off-the-shelf devices. Applied Sciences, 12(19), 9450. https://doi.org/10.3390/app12199450

Zakaria, S., Mativenga, P. T., & Ariff, E. A. R. E. (2023). An investigation of energy consumption in fused deposition modelling using ESP32 IoT monitoring system. Procedia CIRP, 116, 263–268. https://doi.org/10.1016/j.procir.2023.02.045