

Design and Implementation of a Laravel-Based Hybrid Monitoring and Control System for the Cah Fungi Mushroom House

by Revi Mario Ilhamsyah

Submission date: 28-Sep-2026 10:29AM (UTC+0700)

Submission ID: 3044137129

File name: 22_Artikel_JTIK_V12N2_Revi_3804.pdf (551.1K)

Word count: 6718

Character count: 43630

Design and Implementation of a Laravel-Based Hybrid Monitoring and Control System for the Cah Fungi Mushroom House

Revi Mario Ilhamsyah^{1*)}, Anang Faktchur Rachman²⁾, Sholeh Rachmatullah³⁾, Jamilatul Badriyah⁴⁾

¹⁾²⁾³⁾⁴⁾Program Studi Informatika, Fakultas Teknik, Universitas Madura

^{*)}Correspondence author: revimarioilhamsyah@gmail.com, Pamekasan, Indonesia

DOI: <https://doi.org/10.37012/jtik.v12i2.3804>

Abstract

Oyster mushroom cultivation requires stable temperature and humidity conditions to maintain a suitable growing-house environment. This study aims to design and develop a Laravel-based hybrid monitoring and control system for the Cah Fungi mushroom growing house by integrating environmental monitoring, device control, and parameter configuration into a single web application. The system uses an ESP32 and DHT22 sensor for data acquisition, MQTT for communication, MySQL for data storage, and Laravel as the central platform for data processing and control logic. Automatic control applies temperature thresholds of 22–28 °C and relative humidity thresholds of 80–90%, while manual control allows users to perform a manual override through the dashboard. The system was developed using the Waterfall method and evaluated using Black Box Testing and response-time measurements. The results show that monitoring, MQTT communication, automatic and manual control, manual override, dynamic configuration, and data visualization operated as designed. The system achieved an average response time of 0.407 seconds, while control parameters could be updated without reflashing the ESP32 firmware. Future development should incorporate CO₂ and light-intensity sensors to provide more comprehensive environmental monitoring.

Keywords: Internet of Things, Hybrid Control, Laravel, MQTT, Mushroom Growing House

Abstrak

Budidaya jamur tiram memerlukan kestabilan suhu dan kelembapan agar kondisi lingkungan kumbung mendukung pertumbuhan jamur. Penelitian ini bertujuan merancang dan membangun sistem monitoring dan kendali hibrida pada kumbung jamur Cah Fungi berbasis Laravel yang mengintegrasikan pemantauan, pengendalian perangkat, dan konfigurasi parameter dalam satu aplikasi web. Sistem menggunakan ESP32 dan sensor DHT22 untuk akuisisi data, MQTT sebagai protokol komunikasi, MySQL sebagai basis data, serta Laravel sebagai pusat pengolahan data dan logika kendali. Kendali otomatis menggunakan ambang suhu 22–28 °C dan kelembapan 80–90% RH, sedangkan kendali manual memungkinkan pengguna melakukan manual override melalui dashboard. Pengembangan sistem menggunakan metode Waterfall dan pengujian dilakukan menggunakan Black Box Testing serta pengukuran waktu respons. Hasil pengujian menunjukkan fungsi monitoring, komunikasi MQTT, kendali otomatis dan manual, manual override, konfigurasi dinamis, serta visualisasi data berjalan sesuai rancangan. Sistem menghasilkan waktu respons rata-rata 0,407 detik dan perubahan parameter dapat diterapkan tanpa re-flash ESP32. Pengembangan selanjutnya direkomendasikan menambahkan sensor CO₂ dan intensitas cahaya agar pemantauan kondisi kumbung lebih komprehensif.

Kata Kunci: Internet of Things, Kendali Hibrida, Laravel, MQTT, Kumbung Jamur

PENDAHULUAN

Pemanfaatan teknologi Internet of Things (IoT) dalam sektor pertanian berkembang sebagai pendekatan untuk meningkatkan kemampuan pemantauan dan pengendalian kondisi lingkungan secara berkelanjutan. Integrasi sensor, mikrokontroler, jaringan komunikasi, dan aplikasi memungkinkan data lingkungan diperoleh secara berkala serta digunakan sebagai dasar pengambilan keputusan terhadap perangkat yang dikendalikan. Pada budidaya jamur, pendekatan IoT telah digunakan untuk membangun sistem monitoring dan pengendalian lingkungan secara terintegrasi, termasuk pada budidaya jamur tiram yang membutuhkan pemantauan kondisi lingkungan secara kontinu (Chong et al., 2023; Guragain et al., 2024; Mulyanto et al., 2024).

Jamur tiram (*Pleurotus ostreatus*) memerlukan kondisi lingkungan yang sesuai agar pertumbuhan dan pembentukan tubuh buah dapat berlangsung dengan baik. Suhu dan kelembapan merupakan parameter penting karena perubahan keduanya dapat memengaruhi pertumbuhan jamur serta kondisi media budidaya. Beberapa penelitian menunjukkan bahwa jamur tiram memerlukan lingkungan dengan kelembapan relatif tinggi, terutama sekitar 80–90%, sedangkan suhu pertumbuhan dapat berada pada kisaran sekitar 20–28 °C bergantung pada fase pertumbuhan dan kondisi budidaya (Desnanjaya & Sugiartawan, 2022; Mesele et al., 2024; Olaniyan et al., 2025). Berdasarkan literatur tersebut dan kebutuhan operasional lokasi penelitian, sistem dalam penelitian ini menetapkan rentang suhu 22–28 °C dan kelembapan relatif 80–90% sebagai parameter kendali.

Pemeliharaan kondisi lingkungan tersebut menjadi tantangan apabila pengelolaan kumbung masih bergantung pada pengamatan dan intervensi pengguna secara langsung. Perubahan suhu dan kelembapan dapat terjadi akibat kondisi cuaca, ventilasi, aktivitas penyiraman, maupun akumulasi panas di dalam kumbung. Sistem monitoring berbasis IoT memungkinkan kondisi lingkungan dipantau secara berkelanjutan dan menyediakan data historis tanpa mengharuskan pengguna selalu berada di lokasi (M. E. N. Ntihung & Sugiartawan, 2024). Penelitian lain juga menunjukkan bahwa integrasi ESP32, sensor lingkungan, dan aktuator dapat dimanfaatkan untuk mendukung otomatisasi kondisi budidaya jamur (Andrianto, 2025; Elewi et al., 2024). Dengan demikian, penerapan monitoring dan pengendalian berbasis IoT dapat mengurangi ketergantungan terhadap pemeriksaan manual sekaligus mempercepat respons terhadap perubahan kondisi lingkungan.

Berbagai penelitian telah dikembangkan untuk menerapkan IoT pada budidaya jamur. (Kristiyanti et al., 2022) mengembangkan sistem monitoring suhu dan kelembapan menggunakan MQTT yang mampu menyampaikan kondisi kumbung secara real-time. (Saputra et al., 2022) menerapkan Fuzzy Logic dalam pengendalian suhu dan kelembapan, sedangkan (Andrianto et al., n.d.) mengembangkan monitoring dan otomatisasi penyiraman menggunakan ESP32 dan DHT22. Pada penelitian internasional, (Chong et al., 2023) mengembangkan sistem monitoring dan pengendalian lingkungan budidaya jamur berbasis IoT melalui aplikasi web dan perangkat bergerak. (Irwanto et al., 2024) mengintegrasikan IoT dengan fuzzy logic untuk pengelolaan lingkungan budidaya, sementara (Elewi et al., 2024) mengembangkan sistem budidaya jamur tiram cerdas yang mengintegrasikan sensor, aktuator, dan antarmuka monitoring.

Penelitian-penelitian tersebut menunjukkan bahwa monitoring dan otomatisasi lingkungan budidaya jamur telah berkembang dari sistem pemantauan sederhana menuju sistem yang mengintegrasikan sensor dan aktuator. Namun, masih terdapat ruang pengembangan pada aspek fleksibilitas kendali dan pengelolaan sistem. Sebagian sistem lebih berorientasi pada monitoring atau kendali otomatis, sedangkan mekanisme yang memungkinkan pengguna mengambil alih aktuator tertentu melalui manual override dan mengubah parameter kendali secara langsung melalui satu aplikasi belum menjadi fokus utama. (Guragain et al., 2024) telah menunjukkan penerapan ekosistem IoT terpusat pada budidaya jamur tiram, sedangkan (M. E. N. Ntihung & Sugiartawan, 2024) mengembangkan sistem informasi berbasis web untuk monitoring kondisi kumbung dan penyimpanan data historis. Penelitian ini mengembangkan pendekatan tersebut lebih lanjut dengan memusatkan fungsi monitoring, logika kendali, konfigurasi parameter, dan pengelolaan perangkat dalam backend Laravel.

Permasalahan serupa ditemukan pada sistem awal yang sebelumnya diterapkan pada kumbung jamur Cah Fungi. Sistem tersebut menggunakan ESP32 dan DHT22 dengan Node-RED sebagai pengolah alur data, InfluxDB sebagai penyimpanan data deret waktu, Grafana sebagai media visualisasi, serta Virtual Private Server (VPS). Sistem telah mampu melakukan monitoring kondisi lingkungan, tetapi pengelolaan fungsi masih tersebar pada beberapa platform dan perubahan parameter kendali belum dapat dilakukan secara langsung melalui satu aplikasi. Kondisi tersebut menjadi dasar pengembangan sistem baru yang lebih terintegrasi dan dapat beroperasi pada jaringan lokal.

Pada penelitian ini, Laravel digunakan sebagai aplikasi utama untuk mengintegrasikan pengelolaan data, antarmuka monitoring, konfigurasi parameter, dan logika kendali. Laravel menerapkan arsitektur Model-View-Controller yang mendukung pemisahan pengelolaan data, logika aplikasi, dan antarmuka secara terstruktur serta dapat diintegrasikan dengan perangkat IoT dan komunikasi MQTT (Ardhana et al., 2023). Data suhu dan kelembapan diperoleh menggunakan ESP32 dan DHT22, sedangkan MQTT digunakan sebagai media komunikasi antara perangkat dan aplikasi. Penggunaan ESP32 dan DHT22 juga telah diterapkan dalam penelitian pengendalian lingkungan kumbung jamur karena mampu mendukung proses akuisisi suhu dan kelembapan serta komunikasi pada sistem IoT (Fatkhur Rohman et al., n.d.; Sanhaji et al., 2023).

Sistem yang dikembangkan menerapkan mekanisme kendali hibrida yang menggabungkan kendali otomatis dan manual pada setiap aktuator. Seluruh keputusan otomatis diproses pada backend Laravel berdasarkan data sensor dan parameter yang tersimpan dalam basis data. Mode manual memungkinkan pengguna melakukan manual override terhadap perangkat tertentu tanpa menghentikan proses otomatis pada aktuator lainnya. Selain itu, konfigurasi dinamis memungkinkan nilai ambang suhu dan kelembapan diubah melalui dashboard tanpa melakukan perubahan atau re-flash firmware ESP32. Pendekatan tersebut membedakan penelitian ini dari sistem yang hanya menggunakan kendali otomatis berbasis aturan atau fuzzy logic karena pengguna tetap memperoleh kewenangan untuk menentukan mode operasi masing-masing aktuator.

Kontribusi utama penelitian ini adalah integrasi monitoring suhu dan kelembapan, komunikasi MQTT, kendali otomatis dan manual per aktuator, manual override, konfigurasi ambang batas secara dinamis, penyimpanan histori, dan visualisasi data dalam satu aplikasi Laravel yang beroperasi pada jaringan lokal. Berdasarkan permasalahan dan kesenjangan tersebut, penelitian ini bertujuan merancang dan membangun sistem monitoring dan kendali hibrida pada kumbung jamur *Cah Fungi* berbasis web yang mampu melakukan monitoring suhu dan kelembapan secara real-time, menjalankan kendali otomatis dan manual, menyediakan konfigurasi parameter secara dinamis, serta mengintegrasikan ESP32 dan aplikasi melalui protokol MQTT dalam satu sistem yang terpusat dan responsif.

METODE PENELITIAN

Penelitian ini merupakan penelitian pengembangan sistem yang bertujuan merancang dan membangun sistem monitoring dan kendali hibrida pada kumbung jamur Cah Fungi berbasis web. Pengembangan sistem menggunakan model Waterfall yang terdiri atas tahapan requirements analysis, system design, implementation, testing, dan maintenance. Model tersebut digunakan karena kebutuhan sistem telah ditentukan sejak tahap awal dan setiap proses pengembangan dilakukan secara berurutan dan terdokumentasi (Ardhana et al., 2023). Sistem mengintegrasikan sensor DHT22 dan ESP32 sebagai perangkat akuisisi data, MQTT sebagai media komunikasi, Laravel sebagai pusat pengolahan data dan logika kendali, serta MySQL sebagai media penyimpanan data.

Desain dan Lokasi Penelitian

Penelitian ini merupakan penelitian pengembangan sistem yang bertujuan menghasilkan aplikasi monitoring dan kendali hibrida untuk pengelolaan kondisi lingkungan kumbung jamur Cah Fungi. Pengembangan sistem menggunakan model Waterfall karena kebutuhan utama sistem telah ditentukan sejak awal dan proses pengembangan dilakukan secara bertahap mulai dari analisis kebutuhan hingga pengujian. Model Waterfall memberikan alur pengembangan yang terstruktur melalui tahapan requirements analysis, system design, implementation, testing, dan maintenance (Ardhana et al., 2023).

Penelitian dilaksanakan pada kumbung jamur Cah Fungi yang berlokasi di Kabupaten Sampang, Madura. Pelaksanaan penelitian berlangsung secara bertahap selama Januari–Juni 2026, dimulai dari analisis kebutuhan, perancangan sistem, implementasi perangkat keras dan aplikasi berbasis web, hingga pengujian pada lingkungan pengembangan dan lokasi budidaya. Pengumpulan kebutuhan dilakukan melalui observasi kondisi kumbung, wawancara dengan pengelola Cah Fungi, serta studi literatur mengenai kebutuhan iklim mikro jamur tiram, teknologi IoT, protokol MQTT, dan pengembangan aplikasi web.

Tahapan Pengembangan Sistem

Tahap requirements analysis dilakukan untuk mengidentifikasi kebutuhan fungsional dan nonfungsional sistem. Kebutuhan fungsional meliputi monitoring suhu dan kelembapan secara berkala, penyimpanan histori pengukuran, visualisasi data melalui dashboard, kendali otomatis dan manual terhadap aktuator, manual override, serta perubahan nilai ambang batas melalui antarmuka web. Kebutuhan nonfungsional meliputi penggunaan jaringan lokal,

waktu komunikasi yang rendah, penyimpanan konfigurasi secara persisten, autentikasi pengguna, dan kemudahan pengoperasian sistem.

Tahap system design mencakup perancangan arsitektur komunikasi antara sensor DHT22, ESP32, broker MQTT Mosquitto, aplikasi Laravel, basis data MySQL, dan aktuator. Pada tahap ini juga dilakukan perancangan basis data, antarmuka dashboard, topik komunikasi MQTT, serta mekanisme kendali hibrida. Tahap implementation dilakukan melalui pemrograman ESP32 menggunakan Arduino IDE, pengembangan backend dan antarmuka menggunakan Laravel, penyimpanan data menggunakan MySQL, serta visualisasi data menggunakan Chart.js.

Tahap testing dilakukan setelah seluruh komponen terintegrasi. Pengujian diarahkan pada fungsi autentikasi, monitoring sensor, komunikasi MQTT, kendali manual, kendali otomatis, manual override, konfigurasi dinamis, dan waktu respons sistem. Tahap maintenance dalam penelitian ini dilakukan dalam bentuk identifikasi keterbatasan hasil implementasi serta penyusunan rekomendasi pengembangan sistem lebih lanjut.

Perangkat Keras dan Perangkat Lunak

Sistem menggunakan ⁶ perangkat keras dan perangkat lunak yang saling terintegrasi untuk menjalankan proses akuisisi data, komunikasi, pemrosesan logika kendali, penyimpanan, dan pengendalian aktuator. Komponen utama yang digunakan terdiri atas ESP32 DevKit V1, sensor DHT22, modul relay empat kanal, kipas, sistem misting, lampu, dan pompa air. Pada sisi perangkat lunak digunakan Laravel, PHP, MySQL, Eclipse Mosquitto, Arduino IDE, JavaScript, AJAX, dan Chart.js.

Rincian perangkat utama dan fungsi setiap komponen perlu disajikan secara ringkas agar konfigurasi sistem yang diuji dapat direplikasi. Spesifikasi perangkat keras dan perangkat lunak yang digunakan ditunjukkan pada Tabel 1.

Tabel 1.
Spesifikasi Perangkat Keras dan Lunak

Kategori	Komponen	Spesifikasi / Fungsi Utama
Perangkat Keras	Mikrokontroler	ESP32 DevKit V1 dengan integrasi Wi-Fi 2.4GHz
Perangkat Keras	Sensor Lingkungan	DHT22 dengan akurasi suhu $\pm 0.5^{\circ}\text{C}$ dan kelembaban $\pm 2\%$ RH
Perangkat Keras	Modul Relai	Relay 4 Channel sebagai saklar aktuator elektronik
Perangkat Keras	Aktuator	Kipas AC, Pompa Misting, Lampu LED, dan Pompa Air
Perangkat Lunak	Kerangka Kerja Backend	Laravel versi 10.x dengan arsitektur MVC
Perangkat Lunak	Bahasa Pemrograman	PHP 8.x, JavaScript (Chart.js & AJAX), C++ (Arduino IDE)
Perangkat Lunak	Basis Data	MySQL untuk penyimpanan konfigurasi dan histori sensor
Perangkat Lunak	Protokol Komunikasi	MQTT dengan Broker Mosquitto yang berjalan secara lokal
Perangkat Lunak	Visualisasi Data	Chart.js untuk menampilkan grafik time-series
Perangkat Lunak	Protokol Komunikasi	Eclipse Mosquitto 2.x

Berdasarkan Tabel 1, ESP32 berfungsi sebagai perangkat akuisisi data dan eksekutor perintah aktuator, sedangkan pengolahan data dan pengambilan keputusan kendali dipusatkan pada backend Laravel. Sensor DHT22 digunakan untuk memperoleh nilai suhu dan kelembapan karena mampu mengukur kedua parameter dalam satu perangkat dan telah banyak diterapkan pada sistem monitoring lingkungan kumbung berbasis IoT (Fatkhur Rohman et al., n.d.; Sanhaji et al., 2023).

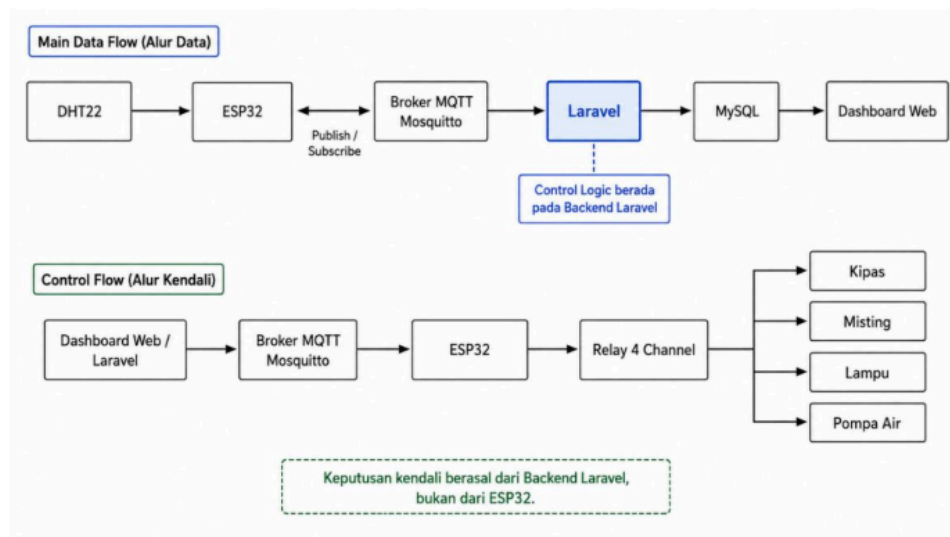
Arsitektur Sistem

Arsitektur sistem menggunakan jaringan lokal yang menghubungkan perangkat IoT dengan aplikasi web melalui protokol MQTT. Data suhu dan kelembapan dibaca oleh DHT22 dan diteruskan kepada ESP32. Selanjutnya, ESP32 melakukan publish data sensor menuju broker Mosquitto. Laravel berlangganan pada topik sensor, menerima data tersebut, melakukan validasi, menyimpannya ke basis data MySQL, kemudian menyediakan data bagi dashboard monitoring. Jalur komunikasi sebaliknya digunakan untuk pengendalian aktuator, yaitu Laravel melakukan publish perintah kendali menuju broker MQTT, kemudian ESP32 menerima perintah dan mengubah kondisi relay sesuai instruksi.

Penggunaan MQTT memungkinkan komunikasi dilakukan menggunakan mekanisme publish/subscribe sehingga perangkat pengirim dan penerima bertukar informasi melalui

broker. Pendekatan serupa telah digunakan dalam monitoring kumbung jamur karena mendukung pertukaran data sensor secara berkelanjutan dan komunikasi yang relatif ringan pada sistem IoT (Kristiyanti et al., 2022).

Hubungan antarbagian sistem perlu divisualisasikan agar alur monitoring dan pengendalian dapat dipahami secara jelas. Arsitektur sistem yang digunakan dalam penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Alur Arsitektur Sistem

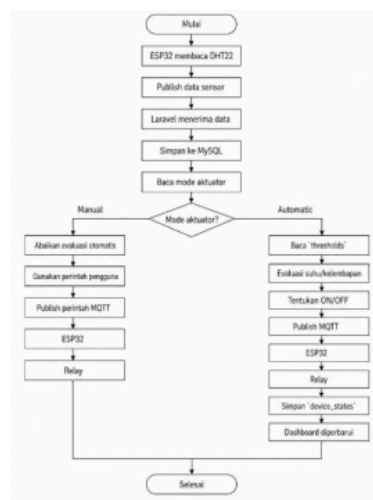
Berdasarkan Gambar 1, Laravel menjadi pusat aplikasi dan pusat keputusan kendali, sedangkan ESP32 tidak melakukan evaluasi nilai ambang batas secara mandiri. Pendekatan tersebut menjaga fungsi firmware tetap sederhana karena perangkat hanya bertanggung jawab membaca sensor, bertukar pesan melalui MQTT, dan mengeksekusi status relay. Dengan demikian, perubahan aturan kendali maupun konfigurasi dapat dilakukan pada sisi aplikasi tanpa melakukan re-flash firmware ESP32.

Basis data yang digunakan terdiri atas enam tabel utama, yaitu users, devices, sensor_readings, device_states, thresholds, dan system_settings. Tabel users menyimpan akun pengguna, devices menyimpan identitas serta mode operasi aktuator, sensor_readings menyimpan histori suhu dan kelembapan, device_states mencatat perubahan status aktuator, thresholds menyimpan konfigurasi nilai ambang batas, sedangkan system_settings digunakan untuk konfigurasi umum aplikasi. Struktur tersebut digunakan untuk mendukung proses autentikasi, monitoring, kendali hibrida, pencatatan histori, dan konfigurasi dinamis dalam satu aplikasi.

Mekanisme Kendali Hibrida

Kendali hibrida dirancang dengan menggabungkan mode otomatis dan manual pada setiap aktuator secara independen. Setiap perangkat memiliki informasi mode operasi yang tersimpan dalam basis data sehingga satu aktuator dapat berjalan secara otomatis sementara aktuator lainnya berada pada mode manual. Seluruh proses evaluasi mode dan kondisi sensor dilakukan pada backend Laravel. Pada mode otomatis, setiap data baru yang diterima Laravel melalui MQTT dibandingkan dengan nilai ambang batas yang aktif pada tabel thresholds. Rentang suhu acuan ditetapkan sebesar 22–28 °C, sedangkan kelembapan relatif ditetapkan sebesar 80–90%. Rentang tersebut digunakan sebagai parameter operasional penelitian dan tetap dapat diperbarui melalui fitur konfigurasi dinamis pada dashboard.

Aturan kendali suhu dan kelembapan disusun berdasarkan hubungan antara nilai sensor dengan fungsi aktuator. Apabila suhu melebihi 28 °C, backend Laravel mengirimkan perintah untuk mengaktifkan kipas. Apabila suhu berada di bawah 22 °C, kipas dinonaktifkan. Apabila kelembapan berada di bawah 80%, sistem mengaktifkan misting dan pompa air untuk meningkatkan kelembapan. Sebaliknya, apabila kelembapan melebihi 90%, misting dan pompa air dinonaktifkan dan kipas dapat diaktifkan untuk membantu sirkulasi udara. Lampu digunakan sebagai perangkat pencahayaan operasional sehingga pengendaliannya dilakukan secara manual melalui dashboard. Mekanisme pengambilan keputusan tersebut dapat diperjelas melalui diagram alur sehingga hubungan antara pembacaan sensor, pemeriksaan mode, dan pengendalian aktuator terlihat secara sistematis. Alur kendali hibrida ditunjukkan pada Gambar 2.



Gambar 2. Flowchart Kendali Hibrida

Berdasarkan Gambar 2, prioritas kendali ditentukan oleh mode setiap perangkat. Ketika perangkat berada pada mode manual, keputusan otomatis terhadap perangkat tersebut tidak dijalankan sehingga pengguna dapat melakukan intervensi tanpa konflik dengan logika backend. Ketika mode dikembalikan menjadi otomatis, backend kembali menggunakan nilai sensor dan konfigurasi ambang batas sebagai dasar pengambilan keputusan.

Konfigurasi Dinamis

Konfigurasi dinamis diterapkan untuk memungkinkan pengguna mengubah nilai ambang batas melalui dashboard tanpa melakukan perubahan pada firmware ESP32. Nilai minimum dan maksimum suhu maupun kelembapan disimpan pada tabel thresholds. Setiap perubahan yang dilakukan melalui antarmuka terlebih dahulu divalidasi oleh Laravel untuk memastikan bahwa nilai yang dimasukkan bersifat numerik dan nilai minimum tidak melebihi nilai maksimum.

Setelah konfigurasi berhasil disimpan, backend menggunakan nilai tersebut secara langsung pada proses evaluasi sensor berikutnya. Konfigurasi tidak diteruskan sebagai logika kendali ke ESP32 karena seluruh proses pengambilan keputusan tetap dipusatkan pada Laravel. ESP32 hanya menerima hasil keputusan berupa perintah kondisi aktuator. Mekanisme tersebut memungkinkan perubahan parameter dilakukan tanpa penghentian sistem maupun pengunggahan ulang firmware.

Teknik Pengujian dan Analisis Data

Pengujian dilakukan melalui dua tahap. Tahap pertama dilaksanakan pada lingkungan pengembangan untuk memastikan bahwa setiap komponen bekerja sesuai rancangan. Setelah fungsi dasar dinyatakan berjalan, pengujian dilanjutkan pada lingkungan kumbung jamur Cah Fungi untuk mengevaluasi sistem dalam kondisi operasional sebenarnya. Skenario pengujian mencakup kondisi normal, kondisi batas (boundary), dan kondisi di luar ambang untuk mengetahui respons sistem terhadap berbagai masukan.

Metode utama yang digunakan adalah Black Box Testing, yaitu pengujian terhadap hubungan antara masukan dan keluaran tanpa meninjau struktur internal program. Aspek yang diuji meliputi autentikasi, komunikasi MQTT, monitoring suhu dan kelembapan, kendali manual, kendali otomatis, manual override, konfigurasi dinamis, visualisasi data, dan pencatatan histori. Hasil setiap skenario dibandingkan dengan keluaran yang diharapkan

dan dinyatakan berhasil apabila perilaku aplikasi, pesan MQTT, status aktuator, serta data yang tersimpan sesuai dengan rancangan.

Selain pengujian fungsional, dilakukan pengujian komunikasi dan waktu respons sistem. Pengujian komunikasi digunakan untuk memastikan data sensor dari ESP32 dapat diterima dan disimpan oleh Laravel serta perintah dari Laravel dapat diterima dan dieksekusi oleh ESP32. Waktu respons dianalisis secara deskriptif berdasarkan hasil beberapa percobaan pengiriman perintah terhadap aktuator, kemudian dihitung nilai rata-ratanya untuk menggambarkan respons komunikasi sistem.

Keabsahan hasil pengujian diperiksa melalui kesesuaian antara kondisi masukan, keluaran yang diharapkan, status aktual perangkat, serta data yang tercatat pada aplikasi. Hasil pengujian juga diperiksa pada sisi broker MQTT dan basis data untuk memastikan bahwa perubahan yang terlihat pada dashboard sesuai dengan pesan yang dipertukarkan dan kondisi perangkat yang dijalankan. Data hasil monitoring disajikan dalam bentuk nilai terkini, tabel histori, dan grafik deret waktu, sedangkan hasil pengujian fungsi dan waktu respons disajikan dalam bentuk tabel serta dijelaskan secara naratif.

HASIL DAN PEMBAHASAN

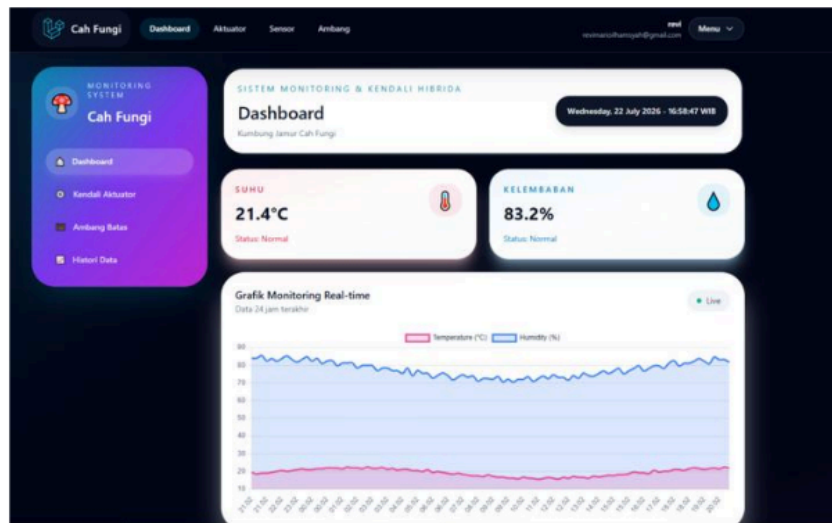
Hasil penelitian menunjukkan bahwa sistem monitoring dan kendali hibrida berhasil diimplementasikan dengan mengintegrasikan ESP32, sensor DHT22, broker MQTT Mosquitto, aplikasi berbasis Laravel, dan basis data MySQL dalam satu jaringan lokal. Sistem mampu menjalankan fungsi monitoring suhu dan kelembapan, menyimpan histori pengukuran, menampilkan visualisasi data, mengendalikan aktuator secara otomatis maupun manual, serta mengubah parameter kendali melalui dashboard. Seluruh keputusan kendali otomatis diproses pada backend Laravel, sedangkan ESP32 berfungsi sebagai perangkat akuisisi data dan eksekutor perintah aktuator.

Hasil Implementasi Dashboard Monitoring

Aplikasi yang dihasilkan menyediakan satu dashboard terintegrasi sebagai pusat monitoring dan pengendalian kumbung jamur. Informasi utama yang ditampilkan meliputi suhu dan kelembapan terkini, waktu pembacaan terakhir, grafik histori pengukuran, serta kondisi perangkat yang terhubung dengan sistem. Dashboard juga menyediakan akses terhadap

histori data, pengendalian aktuator, konfigurasi nilai ambang batas, dan pengelolaan akun pengguna.

Tampilan dashboard utama perlu ditunjukkan untuk memperlihatkan implementasi antarmuka yang digunakan pengguna selama proses monitoring. Hasil implementasi dashboard ⁵ ditunjukkan pada Gambar 3.



Gambar 3. Hasil Implementasi Dashboard Monitoring

Berdasarkan Gambar 3, informasi monitoring dan kondisi perangkat dapat diakses melalui satu antarmuka tanpa menggunakan Node-RED maupun Grafana sebagai aplikasi terpisah. Data yang telah tersimpan ditampilkan dalam bentuk nilai terkini dan grafik deret waktu sehingga pengguna dapat mengamati perubahan kondisi lingkungan secara lebih mudah. Pembaruan informasi pada dashboard dilakukan menggunakan AJAX sehingga perubahan data dapat ditampilkan tanpa melakukan pemuatan ulang seluruh halaman.

Proses monitoring dimulai ketika DHT22 membaca suhu dan kelembapan, kemudian ESP32 mengirimkan data tersebut ke broker MQTT. Laravel menerima pesan sebagai subscriber, melakukan validasi, dan menyimpannya pada tabel `sensor_readings`. Mekanisme ini menghasilkan aliran data yang terintegrasi antara perangkat fisik dan aplikasi web.

Hasil Monitoring dan Komunikasi MQTT

Pengujian monitoring dilakukan untuk mengetahui keberhasilan proses pengiriman, penerimaan, penyimpanan, dan penyajian data sensor. Selama periode pengujian, sistem berhasil menyimpan 165 data pembacaan yang terdiri atas waktu pencatatan, suhu, dan

kelembapan. Sebagian data hasil monitoring dapat digunakan sebagai sampel untuk menunjukkan kontinuitas penerimaan data.

Hasil pengujian monitoring disajikan pada Tabel 2.

Tabel 2.

Sampel Hasil Monitoring Suhu dan Kelembapan

No.	Waktu Pembacaan	Suhu (°C)	Kelembapan (%)	Status
1	23-07-2026 13:53:06	27,1	81,8	Berhasil diterima
2	23-07-2026 13:53:16	27,1	82,0	Berhasil diterima
3	23-07-2026 13:53:26	27,0	82,1	Berhasil diterima
4	23-07-2026 13:53:36	27,0	82,1	Berhasil diterima
5	23-07-2026 13:53:46	27,0	81,9	Berhasil diterima

Tabel 2 menunjukkan bahwa data suhu dan kelembapan dapat diterima secara berurutan dan disimpan pada basis data berdasarkan waktu pencatatan. Lima baris tersebut merupakan sampel dari total 165 pembacaan yang berhasil tersimpan selama proses pengujian. menunjukkan bahwa alur DHT22–ESP32–MQTT–Laravel–MySQL mampu menjalankan proses penerimaan dan penyimpanan data secara kontinu. Selama pengujian, sebanyak 165 data pembacaan berhasil disimpan dalam basis data. Selain menampilkan kondisi terkini, penyimpanan data historis memungkinkan pengguna mengamati perubahan suhu dan kelembapan berdasarkan waktu pengukuran.

Kemampuan tersebut sejalan dengan penelitian (Kristiyanti et al., 2022), yang menunjukkan bahwa MQTT dapat diterapkan untuk monitoring suhu dan kelembapan pada budidaya jamur tiram. (M. E. Ntuhung et al., 2024) juga menunjukkan bahwa sistem monitoring berbasis IoT dapat mengintegrasikan data lingkungan dengan aplikasi berbasis web serta menyimpan informasi historis. Berbeda dengan kedua penelitian tersebut, MQTT pada penelitian ini tidak hanya digunakan untuk pengiriman data monitoring, tetapi juga sebagai media komunikasi dua arah untuk mengirimkan perintah pengendalian dari Laravel menuju ESP32.

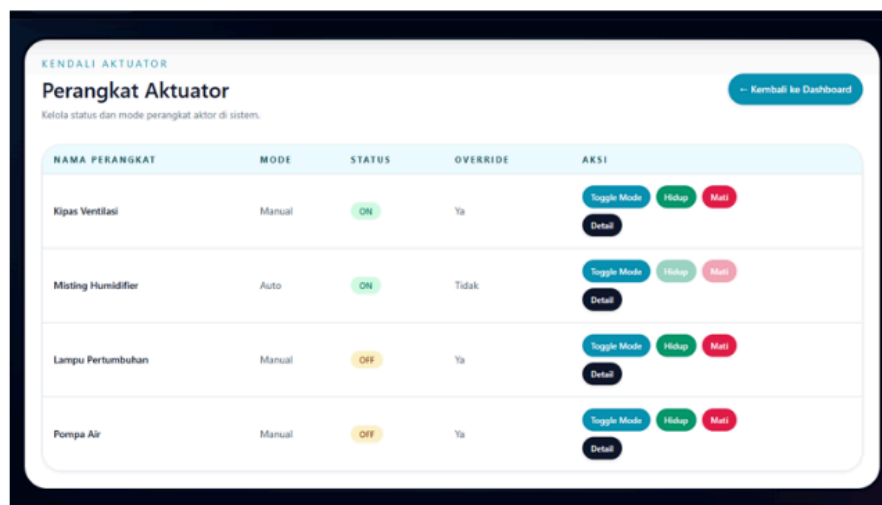
Sistem monitoring dan kontrol dua arah juga dikembangkan oleh (Chong et al., 2023), yang mengintegrasikan sensor, aktuator, serta aplikasi dalam pengelolaan lingkungan budidaya jamur. Hasil penelitian ini menunjukkan bahwa pola komunikasi serupa dapat diterapkan pada arsitektur berbasis jaringan lokal dengan Laravel sebagai pusat pengelolaan aplikasi dan keputusan kendali.

Hasil Kendali Hibrida

Implementasi kendali hibrida memungkinkan setiap aktuator memiliki mode operasi secara independen. Aktuator yang digunakan terdiri atas kipas, sistem misting, lampu, dan pompa air. Backend Laravel memeriksa mode masing-masing perangkat sebelum menjalankan keputusan kendali sehingga satu perangkat dapat tetap berada pada mode otomatis sementara perangkat lain dioperasikan secara manual.

Pada mode manual, seluruh aktuator diuji menggunakan perintah ON dan OFF melalui dashboard. Hasil pengujian menunjukkan bahwa delapan skenario yang terdiri atas ON dan OFF pada kipas, misting, lampu, dan pompa air berhasil dieksekusi sesuai dengan perintah pengguna. Perintah dari dashboard diteruskan oleh Laravel melalui MQTT dan diterima oleh ESP32 untuk mengubah kondisi relay.

Mekanisme pengendalian perangkat pada dashboard perlu ditampilkan karena fitur tersebut merupakan salah satu kontribusi utama sistem. Implementasi halaman kendali ditunjukkan pada Gambar 4.



Gambar 4. Implementasi Kendali Hibrida dan Manual Override

Berdasarkan Gambar 4, mode operasi dapat ditentukan untuk setiap perangkat secara terpisah. Mekanisme tersebut memberikan fleksibilitas kepada pengguna untuk melakukan intervensi hanya terhadap aktuator tertentu tanpa menghentikan fungsi otomatis pada perangkat lainnya.

Pada mode otomatis, backend Laravel membandingkan data sensor dengan rentang kendali aktif sebesar 22–28 °C untuk suhu dan 80–90% untuk kelembapan relatif. Penetapan tersebut

didasarkan pada kebutuhan operasional penelitian dengan mempertimbangkan literatur budidaya *Pleurotus ostreatus*. (Mesele et al., 2024) melaporkan pertumbuhan jamur tiram pada kelembapan relatif ideal 80–90%, sedangkan (Olaniyan et al., 2025) menjelaskan budidaya jamur tiram pada suhu sekitar 20–28 °C dengan kelembapan tinggi sekitar 85–90%. (Desnanjaya & Sugiartawan, 2022) juga menekankan pentingnya pengendalian suhu dan kelembapan pada budidaya jamur tiram di lingkungan tropis. Dengan demikian, rentang 22–28 °C dan 80–90% digunakan sebagai nilai operasional sistem, bukan sebagai nilai universal untuk seluruh fase pertumbuhan jamur.

Pengujian pada parameter suhu telah menunjukkan bahwa ketika suhu mencapai 29,0 °C atau melebihi batas maksimum 28 °C, kipas berhasil diaktifkan secara otomatis. Pada suhu 27,0 °C yang masih berada pada rentang 22–28 °C, kipas tetap dalam kondisi tidak aktif. Untuk menjaga konsistensi dengan konfigurasi final penelitian, hasil pengujian kelembapan perlu menggunakan rentang 80–90% seperti ditunjukkan pada Tabel 3.

Tabel 3.

Hasil Pengujian Kendali Otomatis dengan Threshold Suhu 22–28 °C dan Kelembapan 80–90% RH

Suhu (°C)	Kelembapan (%)	Kondisi	Kipas	Misting	Pompa Air	Hasil
27,2	85,4	Suhu dan kelembapan normal	OFF	OFF	OFF	Sesuai
28,6	84,7	Suhu > 28 °C	ON	OFF	OFF	Sesuai
29,1	87,2	Suhu > 28 °C	ON	OFF	OFF	Sesuai
26,8	78,6	Kelembapan < 80%	OFF	ON	ON	Sesuai
27,5	79,3	Kelembapan < 80%	OFF	ON	ON	Sesuai
25,9	88,1	Kelembapan 80–90%	OFF	OFF	OFF	Sesuai
26,4	91,7	Kelembapan > 90%	ON	OFF	OFF	Sesuai
29,3	92,4	Suhu > 28 °C dan kelembapan > 90%	ON	OFF	OFF	Sesuai

Hasil pengujian kendali otomatis menunjukkan bahwa mekanisme otomatis dapat menghasilkan perintah terhadap aktuator berdasarkan kondisi sensor, sedangkan mode manual memungkinkan pengguna memberikan perintah ON atau OFF secara langsung. Fitur manual override memungkinkan keputusan otomatis dihentikan sementara hanya pada perangkat yang diambil alih pengguna. Setelah manual override dinonaktifkan, perangkat kembali mengikuti keputusan backend berdasarkan data sensor.

(Saputra et al., 2022) dan (Irwanto et al., 2024) menunjukkan bahwa pendekatan fuzzy logic dapat digunakan untuk menghasilkan keputusan otomatis terhadap kondisi lingkungan budidaya jamur. (Elewi et al., 2024) juga mengembangkan otomatisasi melalui integrasi

sejumlah sensor dan aktuator pada budidaya jamur tiram. Perbedaan utama penelitian ini terletak pada mekanisme kendali hibrida per aktuator, yaitu pengguna dapat menentukan perangkat mana yang dikendalikan otomatis dan perangkat mana yang dikendalikan secara manual. Pendekatan tersebut memberikan fleksibilitas ketika kondisi operasional membutuhkan intervensi pengguna tanpa menghentikan keseluruhan fungsi otomatisasi.

Hasil Konfigurasi Dinamis

Konfigurasi dinamis diimplementasikan melalui halaman pengaturan threshold pada dashboard. Pengguna dapat mengubah nilai minimum dan maksimum suhu maupun kelembapan tanpa melakukan perubahan pada firmware ESP32. Setiap masukan divalidasi oleh Laravel sebelum disimpan pada basis data dan digunakan sebagai parameter baru dalam proses evaluasi otomatis.

Tampilan konfigurasi perlu disertakan untuk menunjukkan bahwa perubahan parameter memang dilakukan melalui antarmuka pengguna. Implementasi fitur konfigurasi dinamis ditunjukkan pada Gambar 5.

KONFIGURASI AMBANG BATAS

Threshold Rules

Atur rentang suhu dan kelembapan yang akan mengendalikan perangkat secara otomatis.

[Kembali ke Dashboard](#)

TIPE

Temperature

Nilai Minimum

26

Nilai Maksimum

28

[Simpan Perubahan](#)

TIPE

Humidity

Nilai Minimum

80

Nilai Maksimum

90

[Simpan Perubahan](#)

Gambar 5. Hasil Implementasi Halaman Konfigurasi Dinamis

Berdasarkan Gambar 5, parameter dapat dikelola dari aplikasi web dan disimpan secara persisten pada basis data. Pengujian menunjukkan bahwa nilai minimum dan maksimum suhu maupun kelembapan dapat diubah melalui dashboard dan disimpan pada basis data. Konfigurasi terbaru langsung digunakan oleh backend Laravel pada proses evaluasi data

berikutnya tanpa memerlukan perubahan atau re-flash firmware ESP32. Dengan demikian, parameter pengendalian dipisahkan dari firmware perangkat.

Pendekatan tersebut memperkuat karakter aplikasi sebagai pusat pengelolaan sistem. (Ardhana et al., 2023) menunjukkan bahwa framework Laravel dapat diintegrasikan dengan sistem IoT sehingga pengelolaan aplikasi dan perangkat dapat dilakukan dalam arsitektur berbasis web. Pada penelitian ini, konsep tersebut dikembangkan dengan menempatkan parameter kendali secara persisten pada basis data dan menggunakannya secara dinamis oleh backend.

Hasil Pengujian Fungsional

Pengujian Black Box digunakan untuk memastikan kesesuaian fungsi aplikasi dengan kebutuhan sistem. Pengujian meliputi proses registrasi dan autentikasi, dashboard monitoring, pembaruan data menggunakan AJAX, histori sensor, penyaringan data, paginasi, ekspor CSV, kendali manual, mode otomatis, manual override, pengelolaan threshold, serta profil pengguna.

Ringkasan ⁵ hasil pengujian ditunjukkan pada Tabel 4.

Tabel 4.

Ringkasan Hasil Pengujian Fungsional

No.	Kelompok Pengujian	Hasil
1	Autentikasi dan pengelolaan pengguna	Berhasil
2	Dashboard dan monitoring sensor	Berhasil
3	Histori, filter, paginasi, dan ekspor data	Berhasil
4	Komunikasi MQTT	Berhasil
5	Kendali manual	Berhasil
6	Kendali otomatis	Berhasil
7	Manual override	Berhasil
8	Konfigurasi dinamis dan validasi	Berhasil

Berdasarkan Tabel 4, seluruh kelompok fungsi menghasilkan keluaran sesuai dengan rancangan. Pada tabel pengujian lengkap terdapat 20 skenario Black Box dan seluruhnya berstatus berhasil. Hasil tersebut menunjukkan bahwa integrasi fungsi aplikasi, monitoring, penyimpanan data, dan pengendalian dapat dijalankan melalui satu aplikasi berbasis web. Walaupun seluruh skenario fungsional berhasil, hasil ini perlu dipahami sebagai pengujian terhadap kesesuaian fungsi sistem, bukan sebagai pengukuran akurasi sensor maupun peningkatan produktivitas budidaya. Penelitian ini belum menguji perbandingan hasil panen sebelum dan sesudah sistem diterapkan sehingga manfaat terhadap produktivitas jamur belum dapat dinyatakan secara kuantitatif.

Hasil Pengujian Waktu Respons

Pengujian waktu respons dilakukan untuk mengetahui selang waktu sejak perintah diberikan melalui dashboard hingga perangkat memberikan respons. Pengujian mencakup perintah ON dan OFF pada beberapa aktuator. Hasil pengukuran ditunjukkan pada Tabel 5.

Tabel 5 Hasil Pengujian Waktu Respons Sistem

No.	Perangkat	Perintah	Waktu Respons (detik)
1	Kipas	ON	0,41
2	Kipas	OFF	0,39
3	Misting	ON	0,43
4	Misting	OFF	0,40
5	Lampu	ON/OFF	0,42
6	Pompa Air	ON	0,41
7	Pompa Air	OFF	0,39
Rata-rata			0,407

Berdasarkan Tabel 5, waktu respons berada pada rentang 0,39–0,43 detik dengan rata-rata 0,407 detik. Nilai terendah sebesar 0,39 detik diperoleh pada beberapa perintah OFF, sedangkan nilai tertinggi sebesar 0,43 detik terjadi ketika mengaktifkan misting. Seluruh perintah tetap berhasil dieksekusi.

Waktu respons tersebut menunjukkan bahwa komunikasi melalui broker MQTT pada jaringan lokal mampu memberikan respons kurang dari satu detik pada skenario pengujian yang dilakukan. Penggunaan jaringan lokal mengurangi ketergantungan terhadap koneksi internet eksternal, sedangkan MQTT menyediakan mekanisme pertukaran pesan yang relatif ringan. Namun, nilai 0,407 detik hanya merepresentasikan lingkungan dan skenario pengujian penelitian ini sehingga tidak dapat langsung digeneralisasi untuk jaringan atau konfigurasi perangkat yang berbeda.

Pembahasan

Secara keseluruhan, hasil penelitian menunjukkan bahwa integrasi Laravel, MQTT, ESP32, dan MySQL dapat digunakan untuk membangun sistem monitoring dan kendali lingkungan kumbung dalam satu aplikasi. Sebanyak 165 data pembacaan berhasil diterima selama pengujian, fungsi utama sistem berjalan sesuai skenario Black Box Testing, komunikasi MQTT dapat dilakukan dua arah, serta aktuator memberikan waktu respons rata-rata 0,407 detik pada lingkungan jaringan lokal yang digunakan.

Dari sisi monitoring, sistem mempunyai kesamaan dengan penelitian (Kristiyanti et al., 2022), (Mulyanto et al., 2024), dan (M. E. Ntihung et al., 2024) dalam penggunaan IoT untuk

memperoleh dan menyajikan data kondisi lingkungan kumbung. Namun, penelitian ini tidak berhenti pada fungsi monitoring karena dashboard juga berfungsi sebagai pusat konfigurasi dan kendali perangkat.

Dari sisi arsitektur, (Guragain et al., 2024) menunjukkan bahwa pendekatan IoT terpusat dapat digunakan untuk mengintegrasikan berbagai fungsi pada budidaya jamur tiram. Penelitian ini menerapkan prinsip integrasi serupa dengan memusatkan pengelolaan data, histori, konfigurasi, dan keputusan kendali pada backend Laravel. Broker Mosquitto tetap digunakan sebagai media komunikasi antara aplikasi dan perangkat, tetapi fungsi bisnis utama tidak lagi tersebar pada Node-RED, InfluxDB, Grafana, dan VPS seperti pada sistem awal Cah Fungi.

Dari sisi kendali, penelitian (Saputra et al., 2022), (Irwanto et al., 2024), dan (Elewi et al., 2024) menunjukkan bahwa otomatisasi dapat digunakan untuk mempertahankan kondisi lingkungan budidaya jamur berdasarkan informasi sensor. Penelitian ini menambahkan fleksibilitas melalui mode otomatis dan manual yang dapat ditentukan secara independen pada setiap aktuator serta mekanisme manual override. Dengan demikian, otomatisasi tidak menghilangkan kemampuan pengguna untuk mengambil alih perangkat ketika diperlukan.

Penerapan ESP32, DHT22, dan aktuator pada penelitian ini juga menunjukkan hasil yang sejalan dengan berbagai penelitian monitoring dan pengendalian kumbung berbasis IoT (Andrianto et al., n.d.; Fatkhur Rohman et al., n.d.; Sanhaji et al., 2023). Perbedaannya terletak pada pemusatan pengambilan keputusan di sisi backend. ESP32 hanya menjalankan fungsi akuisisi data, komunikasi, dan eksekusi relay sehingga perubahan parameter maupun logika aplikasi tidak mengharuskan perubahan firmware perangkat.

Meskipun fungsi sistem dapat berjalan sesuai rancangan, penelitian ini masih memiliki keterbatasan. Parameter yang dipantau hanya suhu dan kelembapan, sedangkan penelitian lain menunjukkan bahwa kondisi budidaya jamur juga dapat dipantau melalui parameter tambahan seperti CO₂ dan intensitas cahaya (Chong et al., 2023; M. E. Ntuhung et al., 2024). Penelitian selanjutnya dapat menambahkan parameter tersebut, memperpanjang periode pengujian, meningkatkan jumlah pengukuran waktu respons, serta mengevaluasi pengaruh sistem terhadap produktivitas dan kualitas hasil panen secara kuantitatif.

KESIMPULAN DAN REKOMENDASI

Berdasarkan hasil penelitian dan pengujian, sistem monitoring dan kendali hibrida pada kumbung jamur Cah Fungi berbasis web berhasil dirancang dan diimplementasikan dengan mengintegrasikan ESP32, sensor DHT22, protokol MQTT, Laravel, dan basis data MySQL dalam satu sistem pada jaringan lokal. Sistem mampu melakukan monitoring suhu dan kelembapan, menyimpan histori pembacaan sensor, serta menyajikan kondisi lingkungan melalui dashboard berbasis web. Komunikasi MQTT dapat mendukung pertukaran data dua arah antara ESP32 dan aplikasi, sehingga data sensor dapat diterima oleh Laravel dan perintah pengendalian dapat diteruskan kembali kepada perangkat.

Mekanisme kendali hibrida berhasil menggabungkan kendali otomatis dan kendali manual pada setiap aktuator secara independen. Kendali otomatis menggunakan parameter suhu 22–28 °C dan kelembapan relatif 80–90%, sedangkan fitur manual override memungkinkan pengguna mengambil alih pengendalian perangkat tertentu tanpa memengaruhi mode operasi aktuator lainnya. Konfigurasi dinamis juga berhasil diterapkan sehingga nilai ambang batas dapat diubah melalui dashboard dan langsung digunakan oleh logika kendali pada backend Laravel tanpa melakukan re-flash firmware ESP32. Hasil pengujian Black Box menunjukkan seluruh fungsi utama sistem berjalan sesuai dengan kebutuhan yang dirancang. Pengujian waktu respons menghasilkan nilai antara 0,39–0,43 detik dengan rata-rata 0,407 detik, yang menunjukkan bahwa sistem mampu memberikan respons pengendalian dalam waktu kurang dari satu detik pada lingkungan jaringan lokal yang digunakan selama pengujian.

Pengembangan selanjutnya direkomendasikan untuk menambahkan parameter lingkungan lain, seperti konsentrasi karbon dioksida (CO₂), intensitas cahaya, dan kelembapan media tanam agar kondisi kumbung dapat dipantau secara lebih komprehensif. Pengujian juga dapat diperluas pada periode operasional yang lebih panjang dan jumlah data yang lebih besar, serta dilengkapi dengan evaluasi pengaruh penerapan sistem terhadap produktivitas dan kualitas hasil panen jamur. Selain itu, pengembangan akses jarak jauh dapat dipertimbangkan dengan tetap memperhatikan keamanan komunikasi dan autentikasi agar sistem tidak hanya dapat digunakan pada jaringan lokal.

REFERENSI

1. Andrianto, C. (2025). Sistem Penyiraman Otomatis Pada Budidaya Jamur Tiram Berbasis Iot (Internet of Things). repository.unpkediri.ac.id. <https://repository.unpkediri.ac.id/20613/>
2. Andrianto, C., Maulidina, M., Dewi, M., Puspitasari, M., & Merita Indrawati, E. (n.d.). Sistem Penyiraman Otomatis Pada Budidaya Jamur Tiram Berbasis IoT (Internet of Things). Retrieved <https://ojs.unpkediri.ac.id/index.php/noe>
3. Ardhana, V. Y. P., Hidayat, M. T., Jannah, M., Sumiati, S., Rini, P., & Sari, N. (2023). Implementasi RESTful API Pada Laravel dan Simulator IoT Wokwi Untuk Pengukuran Suhu dan Kelembaban Menggunakan Metode Waterfall. Arcitech: Journal of Computer Science and Artificial Intelligence, 3(2), 93. <https://doi.org/10.29240/arcitech.v3i2.9334>
4. Chong, J. L., Chew, K. W., Peter, A. P., Ting, H. Y., & Show, P. L. (2023). Internet of Things (IoT)-Based Environmental Monitoring and Control System for Home-Based Mushroom Cultivation. Biosensors, 13(1). <https://doi.org/10.3390/bios13010098>
5. Desnanjaya, I. G. M. N., & Sugiartawan, P. (2022). Controlling and Monitoring of Temperature and Humidity of Oyster Mushrooms in Tropical Climates. IJEIS (Indonesian Journal of Electronics and Instrumentation Systems), 12(1), 69. <https://doi.org/10.22146/ijeis.73346>
6. Elewi, A., Hajhamed, A., Khankan, R., Duman, S., Souag, A., & Ahmed, A. (2024). Design and implementation of a cost-aware and smart oyster mushroom cultivation system. Smart Agricultural Technology, 8. <https://doi.org/10.1016/j.atech.2024.100439>
7. Fatkhur Rohman, R., Idhom, M., & Endah Wahanani, H. (n.d.). Seminar Nasional Informatika Bela Negara (SANTIKA) Adaptive PID Kontrol Berbasis IoT pada ESP32 untuk Kontrol Suhu dan Kelembapan Kumbung Jamur Tiram Putih.
8. Guragain, D. P., Shrestha, B., & Bajracharya, I. (2024). A low-cost centralized IoT ecosystem for enhancing oyster mushroom cultivation. Journal of Agriculture and Food Research, 15. <https://doi.org/10.1016/j.jafr.2023.100952>
9. Irwanto, F., Hasan, U., Lays, E. S., De La Croix, N. J., Mukanyiligira, D., Sibomana, L., & Ahmad, T. (2024). IoT and fuzzy logic integration for improved substrate environment

-
- management in mushroom cultivation. *Smart Agricultural Technology*, 7. <https://doi.org/10.1016/j.atech.2024.100427>
10. Kristiyanti, D. R., Wijayanto, A., & Aziz, A. (2022). Sistem Monitoring Suhu dan Kelembaban pada Budidaya Jamur Tiram Berbasis Internet of Things Menggunakan MQTT dan Telegram BOT. *Adopsi Teknologi Dan Sistem Informasi (ATASI)*, 1(1), 61–73. <https://doi.org/10.30872/atasi.v1i1.60>
11. Mesele, E., Yaekob, A. T., & Zeslassie, A. (2024). Valuation of the growth response of oyster (*Pleurotus ostreatus*) mushroom on partially composted sesame stalk with different blends of wheat straw. *Discover Food*, 4(1). <https://doi.org/10.1007/s44187-024-00147-y>
12. Mulyanto, Y., Idifitriani, F., Susanto, E. S., & Sulastris, S. (2024). Implementasi Sistem Monitoring Berbasis Internet Of Things (IoT) pada Rumah Budidaya Jamur Tiram. *Digital Transformation Technology*, 3(2), 871–878. <https://doi.org/10.47709/digitech.v3i2.3404>

Design and Implementation of a Laravel-Based Hybrid Monitoring and Control System for the Cah Fungi Mushroom House

ORIGINALITY REPORT

12%
SIMILARITY INDEX

11%
INTERNET SOURCES

8%
PUBLICATIONS

9%
STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Tengah Student Paper	6%
2	repository.bsi.ac.id Internet Source	3%
3	repository.polibatam.ac.id Internet Source	1%
4	Endang Sukarna, Sofy Fitriani, Mina Naidah Gani. "Meningkatkan Produktivitas Budidaya Jamur Tiram dengan Sistem Kendali", Indonesian Journal of Multidisciplinary on Social and Technology, 2026 Publication	1%
5	proceeding.unuha.ac.id Internet Source	1%
6	Putra Pangihutan Nababan, Marzuarman. "Sistem Smartkey Berbasis ESP8266 Menggunakan Local Web Server untuk Kendali Sepeda Motor dan Monitoring Bahan Bakar", INOVTEK - Seri Elektro, 2026 Publication	1%

Exclude quotes

Off

Exclude matches

< 1%

Exclude bibliography

On