BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1114-1122
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.1108

Simulasi dan Uji Fungsional Dashboard Monitoring Kualitas Air Kolam Ikan Berbasis MQTT

Anjelita Trully Carolina Losung, Riyandy Kumalaka, Yuscar Agsbastian, Maureen Langie, Marike Kondoj*

Program Studi D-4 Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Manado, Manado, Indonesia
Email: [1]anjelita12@elektro.polimdo.ac.id, [2]kumalakariyandy@gmail.com, [3]yuscarags@gmail.com, [4]maureen.langie@gmail.com, [5,*]marikekondoj@polimdo.ac.id
Email Penulis Korespondensi: marikekondoj@polimdo.ac.id

Abstrak-Kualitas air berperan penting dalam budidaya ikan karena perubahan pH, suhu, kadar air, kekeruhan, dan total padatan terlarut (TDS) dapat mempengaruhi kondisi dan kesehatan ikan. Pemantauan manual masih memiliki keterbatasan karena data tidak tersedia secara kontinyu dan peringatan dini sulit diberikan ketika kondisi udara mulai keluar dari batas aman. Penelitian ini bertujuan membangun dan menguji secara fungsional dashboard monitoring kualitas kolam ikan berbasis simulasi Internet of Things (IoT) menggunakan protokol MQTT. Pengujian dilakukan pada simulasi lingkungan untuk alur pengiriman data, penyimpanan data, visualisasi dashboard, klasifikasi status, dan pencatatan peringatan. Hasil pengujian menunjukkan sistem mampu menerima dan menyimpan 204 record data sensor serta menghasilkan 248 data alert pada tabel alert_log. Dashboard berhasil menampilkan status NORMAL, WASPADA, BAHAYA, dan KRITIS sesuai aturan ambang batas yang ditentukan. Penelitian ini berkontribusi pada pengembangan prototipe awal sistem monitoring kualitas air berbasis simulasi yang dapat digunakan untuk memeriksa integrasi alur data dan respons dashboard sebelum diterapkan pada perangkat nyata. Penelitian ini tidak dimaksudkan sebagai validasi sensor fisik maupun evaluasi kinerja jaringan lapangan, melainkan sebagai pengujian awal terhadap fungsionalitas sistem simulasi.

Kata Kunci: Internet of Things; MQTT; Simulasi IoT; Dashboard Monitoring; Kualitas Air

Abstract-Water quality plays a crucial role in fish farming because changes in pH, temperature, water content, turbidity, and total dissolved solids (TDS) can affect the condition and health of fish. Manual monitoring still has limitations because data is not available continuously, and early warnings are difficult to provide when water conditions begin to exceed safe limits. This study aims to develop and functionally test an Internet of Things (IoT)-based fish pond water quality monitoring dashboard using the MQTT protocol. Testing was conducted in a simulated environment to evaluate data transmission, data storage, dashboard visualization, status classification, and alert logging. Test results show that the system is capable of receiving and storing 204 sensor data records and generating 248 alert data entries in the `alert_log` table. The dashboard successfully displays the statuses NORMAL, WARNING, DANGER, and CRITICAL in accordance with the defined threshold rules. This research contributes to the development of an initial prototype of a simulation-based water quality monitoring system that can be used to verify data flow integration and dashboard responses before implementation on real devices. This research is not intended as a validation of physical sensors or an evaluation of field network performance, but rather as an initial test of the simulation system's functionality.

Keywords: Internet of Things; MQTT; IoT Simulation; Monitoring Dashboard; Water Quality

1. PENDAHULUAN

Kualitas air menjadi salah satu faktor yang menentukan keberhasilan budidaya ikan. Kondisi air yang tidak stabil dapat memengaruhi metabolisme, pertumbuhan, nafsu makan, dan tingkat kelangsungan hidup ikan. Pada kolam budidaya, perubahan nilai pH, suhu, kekeruhan, total dissolved solids (TDS), dan level air tidak selalu terlihat secara langsung oleh pembudidaya. Beberapa perubahan bahkan baru disadari setelah air terlihat keruh, berbau, atau ikan menunjukkan penurunan aktivitas. Keadaan ini membuat pemantauan kualitas air perlu dilakukan secara lebih teratur agar pengguna dapat mengetahui kondisi kolam sebelum masalah berkembang menjadi lebih serius.

Pemantauan manual masih banyak digunakan pada skala kecil karena sederhana dan tidak membutuhkan perangkat yang rumit. Namun, cara tersebut memiliki beberapa kelemahan. Data tidak tercatat secara otomatis, perubahan nilai tidak dapat dilihat sebagai riwayat, dan pembudidaya harus berada di lokasi untuk memeriksa kondisi kolam. Selain itu, pengukuran manual sering dilakukan pada waktu tertentu saja, sehingga perubahan yang terjadi di antara waktu pengukuran dapat terlewat. Keterbatasan ini mendorong kebutuhan terhadap sistem yang dapat menampilkan data secara kontinu, menyimpan riwayat, dan memberikan peringatan ketika parameter air keluar dari batas yang ditentukan.

Internet of Things (IoT) digunakan untuk menghubungkan sensor, mikrokontroler, server, basis data, dan antarmuka pengguna dalam satu alur kerja. Pada pemantauan kualitas air, sensor membaca kondisi lingkungan, mikrokontroler mengirimkan data, server menyimpan serta memproses data, sedangkan dashboard menampilkan informasi dalam bentuk yang lebih mudah dipahami. Pemanfaatan IoT pada fish farm telah menunjukkan bahwa data kualitas air dapat dipantau secara lebih terstruktur[1]. Pada sisi lain, penggunaan sensor berbiaya rendah tetap membutuhkan perhatian terhadap ketelitian pembacaan data, khususnya ketika sistem diterapkan pada lingkungan akuakultur yang sebenarnya [2].

Di sisi komunikasi, Message Queuing Telemetry Transport (MQTT) sering dipakai pada sistem IoT karena menerapkan pola publish/subscribe. Pola ini memisahkan perangkat pengirim data, broker, dan penerima data, sehingga dashboard tidak perlu berkomunikasi langsung dengan mikrokontroler. Karakter MQTT yang ringan dan berbasis topik sesuai dengan standar OASIS untuk pengiriman data periodik pada sistem IoT [21]. Dalam penelitian ini, MQTT dipakai untuk menguji alur pengiriman data dari ESP32 simulasi menuju server dan dashboard, bukan untuk menyimpulkan performa jaringan lapangan.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1114-1122
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.1108

Penelitian terkait pemantauan kualitas air menunjukkan arah pengembangan yang beragam. Pada lingkungan akuakultur, sistem berbasis NB-IoT dapat digunakan untuk memantau kualitas air kolam secara jarak jauh [3]. Pendekatan IoT juga telah digunakan pada budidaya udang vaname untuk membantu pencatatan parameter air secara lebih teratur [4]. Kedua arah penelitian tersebut memperlihatkan bahwa pemantauan berbasis jaringan dapat mengurangi ketergantungan pada pemeriksaan manual yang dilakukan pada waktu tertentu saja.

Dalam konteks kolam ikan, beberapa penelitian memusatkan perhatian pada parameter pH, suhu, kekeruhan, TDS, dan kualitas air untuk budidaya koi [5]. Rancangan monitoring akuakultur berbiaya rendah juga menunjukkan bahwa sistem pemantauan dapat dibuat dengan komponen yang relatif sederhana [6]. Selain itu, penggunaan ESP32 dan perangkat IoT lain pada kolam ikan memperlihatkan bahwa data sensor dapat dikirim ke platform pemantauan digital untuk mendukung pengawasan kondisi air.

Beberapa penelitian tidak hanya menampilkan data sensor, tetapi juga menambahkan mekanisme penilaian kondisi air. Pendekatan fuzzy inference dapat digunakan untuk mengubah nilai parameter menjadi kategori kualitas air yang lebih mudah dipahami [7]. Penggunaan metode serupa pada sistem pemantauan kualitas air menunjukkan bahwa data sensor dapat diolah menjadi informasi peringatan atau rekomendasi awal bagi pengguna [15]. Pada pengembangan lain, logika fuzzy juga dipakai untuk klasifikasi kualitas air secara real-time [16].

Pada aspek pengambilan keputusan, sistem IoT dapat dikembangkan menjadi pendukung interpretasi kualitas air, tidak hanya sebagai alat pencatat nilai sensor [13]. Pemantauan real-time pada akuakultur juga menegaskan bahwa sistem harus memperhatikan hubungan antara perangkat, jaringan, dan kestabilan data [8]. Kajian lain pada pemantauan akuakultur menekankan pentingnya pembacaan parameter secara terus-menerus agar perubahan kondisi air dapat diketahui lebih cepat [20].

Kajian literatur terbaru memperlihatkan bahwa integrasi sensor dan IoT menjadi arah penting dalam pemantauan kualitas air [11]. Tinjauan sistematis lain juga menekankan bahwa pemilihan sensor, konteks pengukuran, dan arsitektur komunikasi berpengaruh terhadap kualitas sistem pemantauan [17]. Selain itu, perkembangan pemantauan kualitas air mulai dikaitkan dengan analisis data, prediksi, dan pengambilan keputusan berbasis informasi historis [12], [19]. Penerapan monitoring real-time pada instalasi pengolahan air menunjukkan bahwa konsep serupa juga relevan di luar konteks kolam ikan [14].

Meskipun penelitian sebelumnya sudah banyak membahas perangkat fisik, dashboard, dan klasifikasi kualitas air, masih terdapat ruang untuk membahas pengujian awal alur end-to-end melalui lingkungan simulasi. Simulasi berbasis MQTT dapat digunakan untuk memeriksa format payload, alur publish/subscribe, dan integrasi dashboard sebelum perangkat fisik diterapkan [18]. Wokwi menyediakan lingkungan simulasi ESP32 dan komponen elektronik yang dapat dipakai untuk menguji rancangan awal sistem IoT [22]. Karena itu, penelitian ini menempatkan simulasi sebagai tahap awal untuk memeriksa integrasi publisher MQTT, server, database, WebSocket, dashboard, dan alert.

Berdasarkan celah penelitian tersebut, penelitian ini membangun simulasi dashboard monitoring kualitas air kolam ikan berbasis MQTT sebagai tahap awal pengujian integrasi sistem IoT. Istilah simulasi digunakan secara tegas karena seluruh data dan pengujian dilakukan pada lingkungan Wokwi, bukan menggunakan sensor fisik yang diterapkan langsung pada kolam ikan nyata. Penelitian ini difokuskan pada uji fungsional arsitektur sistem end-to-end, yaitu memastikan data dari ESP32 simulasi dapat dikirim melalui MQTT, diterima server, disimpan ke basis data, diteruskan secara real-time ke dashboard, serta menghasilkan status dan peringatan sesuai aturan ambang batas yang ditentukan. Dengan ruang lingkup tersebut, kontribusi penelitian terletak pada validasi awal alur komunikasi, penyimpanan, visualisasi, dan mekanisme alert pada prototipe monitoring berbasis simulasi. Oleh karena itu, penelitian ini tidak diarahkan untuk mengevaluasi akurasi sensor fisik maupun performa jaringan lapangan, melainkan untuk menilai kesiapan rancangan sistem sebelum diimplementasikan pada lingkungan akuakultur nyata.

2. METODOLOGI PENELITIAN

2.1 Tahapan 1Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang disusun untuk menghasilkan sistem simulasi yang dapat diuji secara fungsional. Tahap pertama adalah studi literatur mengenai pemantauan kualitas air, IoT pada akuakultur, MQTT, dashboard real-time, serta penggunaan Wokwi sebagai lingkungan simulasi. Tahap kedua adalah analisis kebutuhan untuk menetapkan parameter yang dipantau, yaitu pH, suhu air, level air, kekeruhan, dan TDS. Parameter tersebut dipilih karena umum digunakan untuk menggambarkan kondisi dasar air pada kegiatan budidaya.

Tahap berikutnya adalah perancangan arsitektur sistem. Pada tahap ini ditentukan alur data dari ESP32 simulasi menuju broker MQTT, kemudian ke server Node.js, MySQL, WebSocket, dan dashboard browser. Setelah arsitektur ditentukan, perangkat simulasi dibuat pada Wokwi menggunakan ESP32 DevKit V1, LCD I2C 20x4, DS18B20, HC-SR04, dan tiga potensiometer. Potensiometer digunakan untuk membentuk variasi nilai pH, kekeruhan, dan TDS selama pengujian. Perlu ditegaskan bahwa penggunaan potensiometer hanya merepresentasikan perubahan nilai analog, bukan karakteristik sensor kimia sesungguhnya seperti noise, drift, kalibrasi, atau respons terhadap larutan.

Tahap implementasi dilakukan dengan membuat komunikasi MQTT, server penerima data, penyimpanan MySQL, dan dashboard. Tahap pengujian dilakukan untuk memastikan data dapat dikirim, diterima, disimpan, divisualisasikan, dan memunculkan alert sesuai aturan. Tahap terakhir adalah pembahasan hasil dengan menempatkan data sebagai hasil

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1114-1122
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.1108

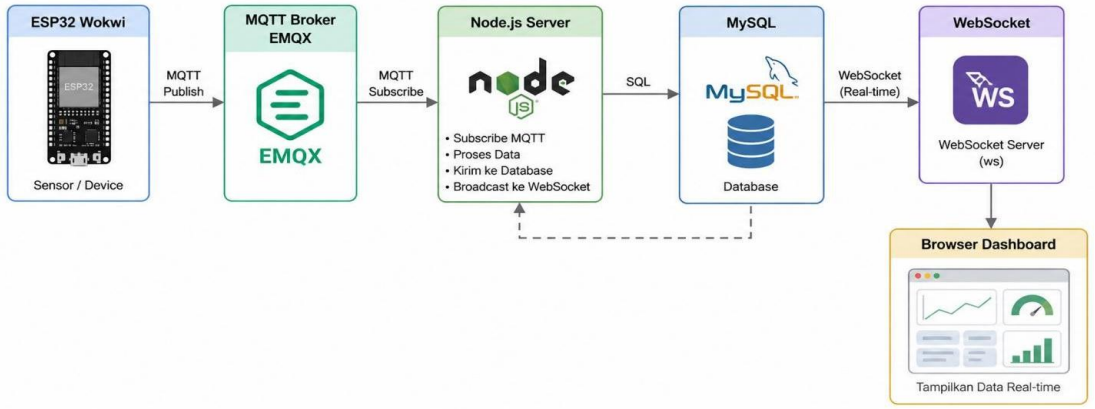
uji simulasi. Oleh karena itu, interpretasi hasil tidak diarahkan untuk menyimpulkan performa jaringan nyata atau akurasi sensor fisik, melainkan untuk menilai kesiapan rancangan sistem sebelum dilakukan pengujian lapangan.

Tabel 1. Tahapan penelitian

No	Tahapan	Keterangan
1	Studi literatur	Mengumpulkan referensi terkait IoT, MQTT, dashboard, Wokwi, dan monitoring kualitas air.
2	Analisis kebutuhan	Menentukan parameter, komponen simulasi, format data, dan kebutuhan penyimpanan.
3	Perancangan sistem	Menyusun arsitektur end-to-end dari ESP32 simulasi sampai dashboard.
4	Implementasi simulasi	Membangun rangkaian Wokwi dan kode publish data sensor.
5	Implementasi server	Menerima data MQTT, menyimpan ke MySQL, dan mengirim ke dashboard melalui WebSocket.
6	Pengujian fungsional	Menguji alur data, visualisasi, penyimpanan, status, dan alert.
7	Analisis hasil	Membahas hasil dengan batasan sebagai pengujian simulasi, bukan validasi lapangan.

2.2 Arsitektur Sistem

Sistem dibangun menggunakan arsitektur IoT end-to-end berbasis simulasi. ESP32 pada Wokwi berperan sebagai publisher yang membaca data sensor simulasi dan mengirimkannya ke EMQX Public Broker melalui MQTT. Pemilihan Wokwi didasarkan pada kemampuannya menyimulasikan ESP32 dan komponen elektronik dalam lingkungan pengembangan embedded system [22], sedangkan MQTT digunakan karena mendukung pola publish/subscribe yang sesuai untuk komunikasi data periodik [21]. Server Node.js menerima data dari broker, menyimpannya ke MySQL melalui driver mysql2, kemudian mengirimkan data terbaru ke dashboard menggunakan WebSocket.



Gambar 1. Arsitektur sistem monitoring kualitas air berbasis simulasi MQTT

Gambar 1 menunjukkan bahwa dashboard tidak mengambil data langsung dari perangkat, tetapi menerima pembaruan dari server setelah data masuk melalui MQTT. Alur ini dipilih agar proses penyimpanan data, pencatatan alert, dan pengiriman data real-time dapat dikelola dari satu server. Dengan cara tersebut, data terbaru dapat ditampilkan pada browser tanpa refresh halaman, sementara data historis tetap tersedia di MySQL.

2.3 Perangkat dan Teknologi Pengembangan Sistem

Teknologi yang digunakan dipilih untuk mendukung simulasi perangkat, komunikasi data, penyimpanan, dan visualisasi. Node.js digunakan sebagai runtime server, Express.js sebagai kerangka backend, WebSocket sebagai jalur pembaruan real-time, dan MySQL sebagai basis data lokal. Pada sisi antarmuka, dashboard dibuat menggunakan HTML5, CSS3, JavaScript murni, dan Chart.js. Simulator Wokwi digunakan untuk menjalankan ESP32 dan komponen sensor secara virtual, sedangkan PlatformIO digunakan untuk pengembangan kode perangkat.

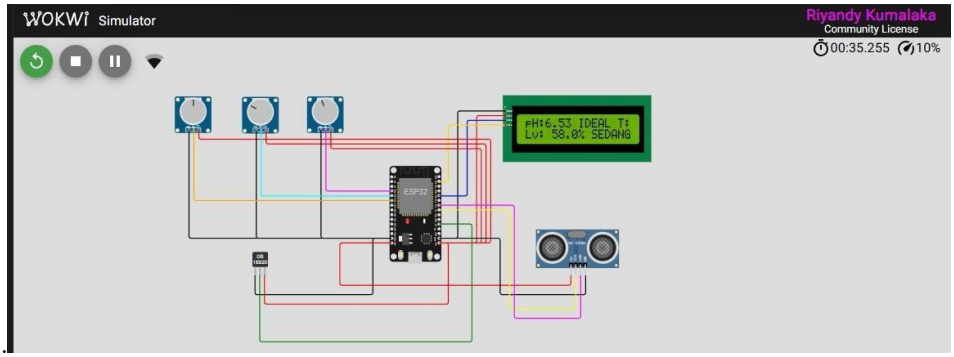
Tabel 2. Teknologi pengembangan sistem

Lapisan	Teknologi
Runtime server	Node.js v24.11.0
Kerangka backend	Express.js v4.18.2
Komunikasi real-time	WebSocket v8.16.0
Frontend	HTML5, CSS3, JavaScript murni
Visualisasi grafik	Chart.js
Basis data	MySQL melalui Laragon dan phpMyAdmin
Driver database	mysql2 v3.9.2
Simulasi perangkat	Wokwi melalui ekstensi VS Code
Alat pengembangan	PlatformIO

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1114-1122
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.1108

2.4 Rangkaian Simulasi Wokwi dan MQTT

Rangkaian simulasi menggunakan ESP32 DevKit V1 sebagai mikrokontroler utama. LCD I2C 20x4 dengan alamat 0x27 menampilkan nilai sensor dan status pada sisi perangkat. Sensor DS18B20 digunakan untuk pembacaan suhu air, sedangkan HC-SR04 digunakan untuk simulasi level air. Tiga potensiometer digunakan sebagai input analog untuk menyimulasikan nilai pH, kekeruhan, dan TDS. Penggunaan potensiometer tidak dimaksudkan sebagai pengganti karakteristik sensor kimia fisik, karena kajian sensor kualitas air menunjukkan bahwa sensor nyata dapat dipengaruhi noise, kalibrasi, dan kondisi lingkungan [11], [17]. Rangkaian Wokwi tetap digunakan sebagai media uji awal alur perangkat dan komunikasi data [22].



Gambar 2. Rangkaian simulasi IoT pada Wokwi

Komunikasi MQTT menggunakan EMQX Public Broker dengan host broker.emqx.io pada port 1883 melalui koneksi TCP. Sistem tidak menggunakan username dan password karena broker bersifat public free broker. Quality of Service yang digunakan adalah QoS 0 dengan pola at most once. Pemilihan QoS 0 mengikuti kebutuhan pengujian fungsional yang mengutamakan komunikasi ringan dan alur publish/subscribe, bukan jaminan pengiriman pesan seperti pada QoS yang lebih tinggi [21]. Konfigurasi ini dirangkum pada Tabel 3.

Tabel 3. Konfigurasi MQTT

Konfigurasi	Nilai
Host	broker.emqx.io
Port	1883
Tipe koneksi	TCP
Jenis broker	Public Free Broker EMQX
Autentikasi	Tanpa username/password
QoS	0
Topik	kolam-ikan/sensors/data
Interval publikasi	2000 ms

2.5 Format Payload dan Batas Parameter

Data dikirim dalam format JSON agar mudah dibaca server dan dashboard. Payload memuat identitas perangkat, nilai sensor, status umum, label per parameter, uptime, dan timestamp dari sisi perangkat. Contoh payload yang digunakan adalah sebagai berikut:

{"device":"FishPond_278d1b55","ph":7.24,"temperature":27.5,"level":63.0,"turbidity":45.5,"tds":282,"status":"NORMAL","ph_label":"IDEAL","temp_label":"IDEAL","level_label":"SEDANG","turbidity_label":"SEDANG","tds_label":"BAGUS","uptime":145,"ts":1234567}

Tabel 4. Batas parameter kualitas air pada sistem

Parameter	Kritis	Bahaya	Waspada	Normal	Ideal/Optimal
pH	<5,5 / >9,0	<5,5 / >9,0	<6,0 / >8,5	6,0-8,5	6,5-7,8
Suhu (°C)	<18 / >38	<22 / >35	<22 / >35	22-35	25-30
Level air (%)	<15	<15	<25	25-75	>=70
Kekeruhan (NTU)	>85	>85	>60	25-60	<=25
TDS (ppm)	>900	-	-	300-600	<=300

2.6 Lingkup Validasi dan Metode Pengujian

Validasi pada penelitian ini dibatasi pada uji fungsional sistem. Pengujian tidak dimaksudkan untuk menilai akurasi sensor fisik karena pH, kekeruhan, dan TDS masih disimulasikan menggunakan potensiometer. Pengujian juga tidak digunakan untuk menyimpulkan performa jaringan secara umum, sebab broker yang dipakai bersifat publik dan perangkat dijalankan pada komputer lokal. Pembatasan ini perlu ditegaskan agar hasil penelitian tidak disamakan dengan implementasi fisik pada kolam nyata.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1114-1122
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.1108

Dalam metodologi ini, simulator dipakai untuk memeriksa apakah data dari ESP32 dapat dikirim secara periodik, diterima server, disimpan ke database, ditampilkan pada dashboard, dan memicu alert sesuai aturan. Pendekatan tersebut sejalan dengan penggunaan simulasi sebagai tahap awal perancangan sistem IoT sebelum perangkat fisik digunakan [18]. Selain itu, Wokwi digunakan karena mendukung simulasi ESP32 dan komponen elektronik yang relevan untuk pengujian alur sistem [22]. Nilai waktu seperti pembaruan WebSocket dan respons API diperlakukan sebagai pengamatan teknis pada lingkungan pengujian. Nilai tersebut belum ditempatkan sebagai pengukuran presisi end-to-end karena penghitungan delay MQTT yang akurat membutuhkan sinkronisasi waktu antara publisher dan server, atau pencatatan timestamp menggunakan sumber waktu yang sama. Dalam penelitian ini, sinkronisasi waktu tersebut belum diterapkan, sehingga nilai delay tidak dijadikan dasar utama kesimpulan.

Keterbatasan tersebut juga berkaitan dengan karakteristik sistem IoT lapangan. Pemantauan real-time pada akuakultur dapat dipengaruhi oleh kondisi jaringan, karakteristik perangkat, dan kestabilan sensor [8]. Pengujian pada lingkungan akuakultur nyata juga menunjukkan bahwa pembacaan parameter air perlu mempertimbangkan kondisi perangkat dan lingkungan pengukuran [20]. Karena itu, pembahasan pada penelitian ini diarahkan pada keberhasilan integrasi alur data dan respons dashboard, bukan pada generalisasi performa jaringan. Dari sisi interpretasi data, penelitian ini menggunakan aturan ambang batas untuk menghasilkan status OPTIMAL, NORMAL, WASPADA, BAHAYA, dan KRITIS. Pemilihan pendekatan rule based dilakukan agar status mudah dipahami pada tahap prototipe. Kajian mengenai sensor IoT menekankan bahwa kualitas pembacaan dan konteks pengukuran tetap perlu diperhatikan ketika sistem diterapkan pada kondisi nyata [11], [17]. Sementara itu, pengembangan sistem pemantauan berbasis data menunjukkan peluang lanjutan untuk analisis, prediksi, dan pendukung keputusan [13], [19]. Dengan batasan tersebut, hasil penelitian ini lebih tepat dipahami sebagai evaluasi awal fungsionalitas sistem simulasi.

Tabel 5. Lingkup pengujian dan batasan validasi

Aspek	Diuji dalam penelitian	Batasan
Alur MQTT	Data dikirim dari ESP32 simulasi ke broker dan diterima server	Tidak menghitung packet loss presisi pada jaringan nyata
Penyimpanan data	Data sensor dan alert tersimpan di MySQL	Tidak membahas optimasi basis data untuk beban besar
Dashboard real-time	Data terbaru didorong melalui WebSocket	Waktu tampil hanya diamati pada lingkungan lokal
Klasifikasi status	Status dibentuk berdasarkan ambang batas rule-based	Bukan model prediksi atau klasifikasi berbasis dataset seimbang
Sensor simulasi	Nilai dapat diubah melalui Wokwi dan potensiometer	Tidak menggambarkan noise, drift, dan kalibrasi sensor fisik

3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Dashboard

Dashboard monitoring berhasil dibuat untuk menampilkan data sensor secara ringkas dan mudah dipahami. Tampilan utama memuat identitas perangkat, uptime, jumlah data, status kolam, indikator server, MQTT, dan live data. Setiap parameter ditampilkan melalui kartu yang memuat nilai numerik, satuan, label kondisi, batas ideal, serta grafik kecil. Pada bagian bawah dashboard tersedia grafik historis sehingga pengguna dapat melihat perubahan parameter dalam rentang waktu tertentu.

Gambar 3. Tampilan dashboard monitoring pada kondisi normal

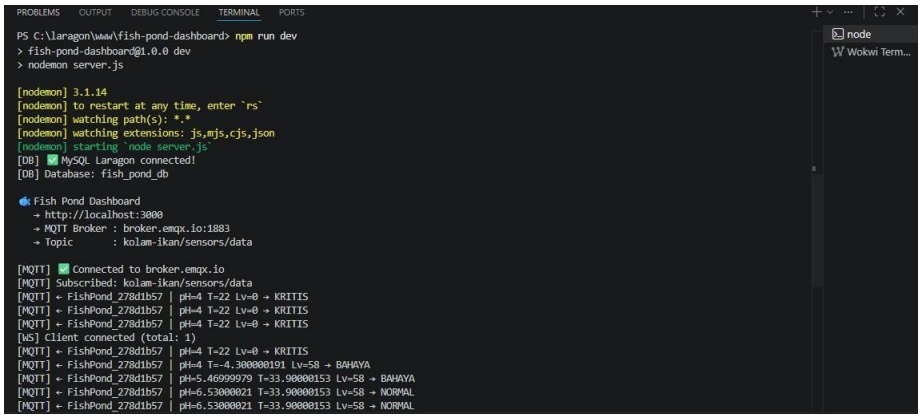
Pada Gambar 3, sistem menampilkan status kolam NORMAL. Nilai pH sebesar 6,53 berada pada rentang ideal 6,5-7,8. Suhu air sebesar 33,9°C masih termasuk normal, meskipun tidak berada pada rentang ideal 25-30°C. Level air

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1114-1122
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.1108

sebesar 58,0% berada pada kategori sedang, kekeruhan sebesar 27,4 NTU berada pada kategori sedang, dan TDS sebesar 493 ppm juga berada pada kategori sedang. Tampilan tersebut menunjukkan bahwa dashboard tidak hanya menampilkan angka, tetapi juga memberi interpretasi sederhana terhadap kondisi setiap parameter.

3.2 Hasil Implementasi Simulasi IoT dan Koneksi Server

Simulasi perangkat IoT dijalankan melalui Wokwi. Nilai dari sensor dan potensiometer diproses oleh ESP32, ditampilkan pada LCD, kemudian dikirim ke broker MQTT. Pada sisi server, Node.js menerima data dari topik kolam-ikan/sensors/data, melakukan parsing JSON, menyimpan data ke tabel sensor_readings, dan mengirimkan data terbaru ke dashboard melalui WebSocket. Alur tersebut menunjukkan bahwa sistem sudah berjalan sebagai rangkaian end-to-end dari perangkat simulasi hingga browser.



Gambar 4. Log koneksi MQTT, MySQL, dan WebSocket pada server Node.js

Gambar 4 memperlihatkan server berhasil terhubung ke MySQL lokal dengan database fish_pond_db, terkoneksi ke broker.emqx.io:1883, dan melakukan subscribe ke topik MQTT yang digunakan. Log juga menunjukkan data diterima dari perangkat FishPond_278d1b57 serta perubahan status seperti KRITIS, BAHAYA, dan NORMAL. Ketika client WebSocket terhubung, data terbaru dapat diteruskan ke browser tanpa refresh halaman.

3.3 Data Sensor Hasil Pengujian

Data pengujian berasal dari tabel sensor_readings yang tersimpan pada MySQL. Total data yang dianalisis sebanyak 204 record dari beberapa sesi pengujian. Data berisi identitas perangkat, nilai pH, suhu, level air, kekeruhan, TDS, status umum kolam, label setiap parameter, uptime, dan waktu pencatatan. Tabel 6 menampilkan sampel data aktual yang mewakili beberapa status sistem.

Tabel 6. Sampel data sensor aktual berdasarkan status

Status	pH	Suhu	Level	Keruh	TDS	Label pH	Label suhu	Label level	Label keruh	Label TDS
NORMAL	6.53	33.9	58.0	27.4	226	IDEAL	NORMAL	SEDANG	SEDANG	BAGUS
WASPADA	5.58	25.0	40.0	21.5	200	ASAM	IDEAL	SEDANG	JERNIH	BAGUS
BAHAYA	5.42	41.9	37.0	26.3	259	ASAM	KRITIS	RENDAH	SEDANG	BAGUS
KRITIS	4.00	25.0	0.0	0.0	0	ASAM	IDEAL	KRITIS	JERNIH	BAGUS

Berdasarkan Tabel 6, sistem menghasilkan status yang berbeda ketika nilai parameter berubah. Status WASPADA muncul ketika pH berada di bawah batas normal, status BAHAYA muncul ketika terdapat parameter yang melewati batas bahaya, dan status KRITIS muncul ketika level air sangat rendah. Hasil ini memperlihatkan bahwa aturan berbasis ambang batas sudah berjalan pada data simulasi.

Tabel 7. Statistik data sensor pada tabel sensor_readings

Parameter	Minimum	Maksimum	Rata-rata	Std. deviasi
pH	4.00	8.64	5.35	1.11
Suhu (°C)	-15.50	78.50	42.17	23.58
Level air (%)	0.00	98.00	10.84	19.50
Kekeruhan (NTU)	0.00	92.20	25.39	19.67
TDS (ppm)	0.00	841.00	247.14	179.14

Tabel 7 memperlihatkan bahwa nilai parameter memiliki variasi yang lebar. Variasi tersebut muncul karena potensiometer dan kondisi simulasi sengaja diubah untuk menguji respons dashboard dan alert. Nilai suhu atau level air yang berada jauh dari rentang ideal tidak dimaknai sebagai kondisi kolam nyata, melainkan bagian dari pengujian fungsi sistem. Penjelasan ini penting agar data simulasi tidak disalahartikan sebagai data lingkungan budidaya yang sebenarnya.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1114-1122
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.1108

Tabel 8. Distribusi status pada data sensor

Status	Jumlah data	Persentase
KRITIS	163	79.90%
BAHAYA	25	12.25%
WASPADA	10	4.90%
NORMAL	6	2.94%

Distribusi pada Tabel 8 menunjukkan bahwa data KRITIS berjumlah paling besar. Kondisi ini bukan menunjukkan kegagalan sistem atau kondisi kolam aktual, melainkan karena pengujian diarahkan sebagai stress test untuk memastikan alert banner, border card, grafik, LCD, dan pencatatan alert dapat bereaksi pada kondisi ekstrem. Dengan demikian, data ini tidak digunakan sebagai dataset klasifikasi yang harus seimbang, tetapi sebagai log pengujian fungsional untuk memeriksa apakah aturan status dapat memicu respons sistem.

3.4 Hasil Implementasi Dashboard pada Kondisi Kritis

Gambar 5. Tampilan dashboard saat status kolam KRITIS

Gambar 5 menunjukkan tampilan dashboard ketika sistem mendeteksi status KRITIS. Pada kondisi tersebut, level air bernilai 9,0% sehingga berada di bawah batas kritis 15%. Parameter lain juga berada di luar rentang ideal, yaitu pH 8,64, suhu 65,8°C, kekeruhan 88,2 NTU, dan TDS 841 ppm. Dashboard merespons kondisi ini dengan menampilkan alert banner berwarna merah, status kolam KRITIS, perubahan warna pada kartu parameter, dan grafik historis yang memperlihatkan perubahan nilai. Tampilan tersebut memperlihatkan fungsi dashboard sebagai media peringatan awal pada skenario simulasi.

3.5 Hasil Implementasi Alert Log

Tabel 9. Distribusi data pada tabel alert_log

Status alert	Jumlah data	Persentase
KRITIS	193	77.82%
BAHAYA	34	13.71%
WASPADA	21	8.47%

Selain data sensor, sistem juga menyimpan peringatan ke tabel alert_log. Dari pengujian yang dilakukan, terdapat 248 data alert. Status KRITIS menjadi alert terbanyak, diikuti BAHAYA dan WASPADA. Dominasi KRITIS sejalan dengan skenario stress test yang sengaja menekan sistem pada kondisi ekstrem. Alert log menyimpan status dan pesan peringatan sehingga pengguna dapat meninjau kembali kejadian yang pernah muncul selama pengujian berlangsung.

3.6 Hasil Pengamatan Fungsional Sistem

Pengamatan sistem dilakukan pada konfigurasi publish MQTT setiap 2 detik. Server menerima data, menyimpannya ke MySQL, dan mengirim pembaruan ke dashboard melalui WebSocket. Karena penelitian ini belum menerapkan sinkronisasi waktu antara ESP32 simulasi dan server, nilai delay MQTT, jitter, throughput, dan packet loss tidak dihitung sebagai metrik presisi. Tabel 10 menyajikan hasil pengamatan fungsional yang dapat diverifikasi dari log dan basis data.

Tabel 10. Hasil pengamatan fungsional sistem simulasi

Indikator	Hasil	Keterangan
Interval publish MQTT	2000 ms	Ditetapkan pada kode ESP32 simulasi.
LCD refresh	600 ms	LCD diperbarui lebih cepat daripada publish MQTT.
Koneksi broker	Berhasil	Server terhubung ke broker.emqx.io:1883 dan subscribe ke topik.
Penyimpanan sensor	204 record	Data tersimpan pada tabel sensor_readings.
Penyimpanan alert	248 record	Data peringatan tersimpan pada tabel alert_log.
WebSocket	Berjalan	Dashboard menerima push data ketika client terhubung.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1114-1122
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.1108

Indikator	Hasil	Keterangan
REST API	Berjalan	Endpoint digunakan untuk mengambil data terbaru, riwayat, dan statistik.
Status rule-based	Berjalan	Sistem menampilkan NORMAL, WASPADA, BAHAYA, dan KRITIS.
Delay/jitter/packet loss	Belum dihitung presisi	Membutuhkan sinkronisasi timestamp dan pengujian jaringan nyata.

Hasil pada Tabel 10 memperlihatkan bahwa bagian utama sistem telah bekerja pada lingkungan simulasi. Data berhasil melewati tahapan publish, subscribe, parsing, penyimpanan, broadcast, dan visualisasi. Akan tetapi, hasil ini tidak dapat digeneralisasi sebagai performa jaringan IoT yang sebenarnya. Untuk mendapatkan analisis kinerja yang lebih kuat, pengujian berikutnya perlu mencatat timestamp pada publisher dan subscriber dengan sumber waktu yang sama, kemudian menghitung rata-rata delay, jitter, throughput, packet loss, dan perbandingan QoS MQTT.

3.7 Pembahasan

Hasil penelitian menunjukkan bahwa simulasi dapat digunakan untuk memeriksa kesiapan alur sistem sebelum perangkat fisik digunakan. Dalam konteks pengembangan, pendekatan ini bermanfaat karena kesalahan pada format payload, topic MQTT, penyimpanan basis data, dan tampilan dashboard dapat ditemukan lebih awal. Penggunaan simulasi pada sistem monitoring berbasis MQTT relevan sebagai tahap awal sebelum implementasi perangkat nyata [18]. Selain itu, Wokwi dapat digunakan sebagai lingkungan uji untuk ESP32 dan komponen IoT sehingga sesuai dengan kebutuhan pemeriksaan alur perangkat simulasi [22].

Jika dibandingkan dengan penelitian yang menggunakan perangkat fisik, kontribusi penelitian ini berada pada sisi integrasi alur dashboard berbasis web secara end-to-end. Beberapa penelitian pada kolam ikan menunjukkan kebutuhan pemantauan kualitas air yang mampu menampilkan data sensor secara teratur [10], [11]. Penelitian lain pada akuakultur menekankan pentingnya pemantauan real-time untuk menjaga kestabilan parameter air [8], [20]. Sementara itu, penelitian ini memperlihatkan bagaimana alur simulasi dari ESP32, MQTT, server, basis data, WebSocket, sampai dashboard dapat diuji sebagai prototipe awal.

Penggunaan QoS 0 pada MQTT memberikan komunikasi yang ringan, tetapi tidak memberikan jaminan pesan diterima. Pada penelitian ini, QoS 0 masih memadai untuk melihat alur publish/subscribe karena data dikirim periodik dan pengujian belum difokuskan pada kehilangan paket. Namun, untuk implementasi nyata, pemilihan QoS perlu diuji ulang dengan mengacu pada karakteristik MQTT dan kebutuhan data yang dikirim [21]. QoS 1 atau QoS 2 dapat dipertimbangkan ketika data peringatan harus lebih terjamin sampai ke server, meskipun konsekuensinya beban komunikasi dapat meningkat.

Kritik terhadap distribusi data yang tidak seimbang juga perlu diperjelas. Data pengujian memang didominasi status KRITIS karena skenario diarahkan untuk memastikan sistem memberikan respons pada kondisi ekstrem. Jika tujuan penelitian adalah membangun model klasifikasi, data seperti ini tidak memadai karena tidak seimbang. Akan tetapi, pada penelitian ini status ditentukan oleh aturan ambang batas, bukan model machine learning. Karena itu, data tidak dipakai untuk melatih atau menguji akurasi klasifikasi, melainkan untuk membuktikan bahwa alert dan perubahan tampilan bekerja ketika parameter melewati batas tertentu.

Keterbatasan penelitian terletak pada lingkungan simulasi. Potensiometer tidak merepresentasikan karakteristik sensor pH, turbidity, dan TDS yang sebenarnya. Sensor fisik dapat mengalami noise, drift, kesalahan kalibrasi, keterlambatan respons, dan pengaruh lingkungan. Kajian mengenai sensor kualitas air menunjukkan bahwa konteks pengukuran dan karakteristik perangkat harus diperhatikan ketika sistem diterapkan pada kondisi lapangan[12], [17]. Selain itu, pemantauan kualitas air berbasis IoT masih dipengaruhi oleh kualitas data, sensor, dan lingkungan pengujian[13]. Broker publik EMQX dan server lokal juga tidak menggambarkan kondisi jaringan lapangan yang mungkin memiliki interferensi, bandwidth terbatas, atau packet loss. Oleh karena itu, hasil penelitian ini lebih tepat dibaca sebagai evaluasi awal pada prototipe simulasi, bukan sebagai validasi akhir sistem monitoring di kolam ikan nyata.

4. KESIMPULAN

Penelitian ini menghasilkan prototipe simulasi dashboard monitoring kualitas air kolam ikan berbasis MQTT yang mampu menjalankan alur data secara end-to-end dari ESP32 pada Wokwi hingga dashboard web. Sistem berhasil melakukan pengiriman, penerimaan, penyimpanan, visualisasi, dan pencatatan data peringatan berdasarkan aturan ambang batas yang telah ditentukan. Hasil pengujian fungsional menunjukkan bahwa sistem mampu mencatat 204 data sensor dan 248 data alert, serta menampilkan status NORMAL, WASPADA, BAHAYA, dan KRITIS sesuai kondisi parameter yang diuji pada lingkungan simulasi. Kontribusi praktis penelitian ini terletak pada penyediaan rancangan awal dan validasi fungsional arsitektur monitoring berbasis simulasi yang dapat digunakan untuk memeriksa integrasi komunikasi data, penyimpanan, visualisasi dashboard, dan mekanisme alert sebelum diterapkan pada perangkat fisik. Namun, penelitian ini memiliki keterbatasan karena seluruh pengujian dilakukan menggunakan simulasi Wokwi dan sebagian parameter masih direpresentasikan menggunakan potensiometer, sehingga hasil penelitian belum dapat digunakan untuk menyimpulkan akurasi sensor maupun performa jaringan pada kondisi lapangan nyata. Selain itu, penelitian ini belum menerapkan sinkronisasi timestamp antara publisher dan server sehingga analisis delay, jitter, throughput, dan packet loss belum dapat dilakukan secara presisi. Penelitian selanjutnya perlu melibatkan sensor fisik, proses kalibrasi, pengujian

pada lingkungan akuakultur nyata, serta evaluasi beberapa skema QoS MQTT agar analisis kinerja sistem dapat dilakukan dengan lebih komprehensif.

REFERENCES

[1] C.-H. Chen, Y.-C. Wu, J.-X. Zhang, and Y.-H. Chen, "IoT-Based Fish Farm Water Quality Monitoring System," *Sensors*, vol. 22, no. 17, p. 6700, 2022, doi: 10.3390/s22176700.
[2] N. A. M. Jais, A. F. Abdullah, M. S. Mohd Kassim, M. M. Abd Karim, A. M., and N. A. Muhadi, "Improved Accuracy In IoT-Based Water Quality Monitoring For Aquaculture Tanks Using Low-Cost Sensors: Asian Seabass Fish Farming," *Heliyon*, vol. 10, no. 8, p. e29022, 2024, doi: 10.1016/j.heliyon.2024.e29022.
[3] J. Huan, H. Li, F. Wu, and W. Cao, "Design Of Water Quality Monitoring System For Aquaculture Ponds Based On NB-IoT," *Aquac. Eng.*, vol. 90, p. 102088, 2020, doi: 10.1016/j.aquaeng.2020.102088.
[4] R. Eso, H. T. Mokui, A. Arman, L. Safiuddin, and H. Husein, "Water Quality Monitoring System Based On The Internet Of Things (IoT) For Vannamei Shrimp Farming," *ComTech Comput. Math. Eng. Appl.*, vol. 15, no. 1, pp. 53–63, 2024, doi: 10.21512/comtech.v15i1.10657.
[5] Sugiarto, I. Nugraha, T. M. Fahrudin, A. Rizqina, and K. Agvenia, "IoT-Based Water Quality Monitoring System To Enhance Sustainability And Business Performance In Koi Fish Cultivation," *J. Adv. Inf. Ind. Technol.*, vol. 7, no. 2, pp. 193–206, 2025, doi: 10.52435/jaiit.v7i2.730.
[6] A. Subowo, "Design Of An IoT-Based Aquaculture Monitoring System For Low-Cost Water Quality Monitoring," *Energy J. Ilm. Ilmu-Ilmu Tek.*, vol. 15, no. 2, 2025, doi: 10.51747/energy.v15i2.15203.
[7] A. F. Choiri, "IoT-Based Water Quality Monitoring System For Fish Ponds Using Fuzzy Inference Method," *J. Teknol. Inf. dan Terap.*, vol. 11, no. 2, 2024, doi: 10.25047/jtit.v11i2.5794.
[8] R. P. Shete, A. M. Bongale, and D. Dharrao, "IoT-Enabled Effective Real-Time Water Quality Monitoring Method For Aquaculture," *MethodsX*, vol. 13, p. 102906, 2024, doi: 10.1016/j.mex.2024.102906.
[9] R. N. Andiyansyah, O. Setyawati, and F. H. Partiansyah, "IoT-Based Water Quality Monitoring System For Koi Fish Quarantine," *J. EECCIS*, vol. 18, no. 3, pp. 92–99, 2024, doi: 10.21776/jeeccis.v18i3.1751.
[10] D. Rosandi, Junaidi, D. K. Apriyanto, and A. Surtono, "Design Of Water Quality Monitoring System For Koi Fish Farming Using NodeMCU ESP32 And Blynk Application Based On Internet Of Things," *J. List. Instrumentasi, dan Elektron. Terap.*, vol. 4, no. 1, 2023, doi: 10.22146/juliet.v4i1.83131.
[11] M. Flores-Iwasaki, G. A. Guadalupe, M. Pachas-Caycho, S. Chapa-Gonza, R. C. Mori-Zabarburu, and J. C. Guerrero-Abad, "Internet Of Things (IoT) Sensors For Water Quality Monitoring In Aquaculture Systems: A Systematic Review And Bibliometric Analysis," *AgriEngineering*, vol. 7, no. 3, p. 78, 2025, doi: 10.3390/agriengineering7030078.
[12] S. Kanwal and others, "An Optimal Internet Of Things-Driven Intelligent Decision-Making System For Water Quality Monitoring," *Sensors*, vol. 24, no. 23, p. 7842, 2024, doi: 10.3390/s24237842.
[13] H. M. Forhad et al., "IoT-Based Real-Time Water Quality Monitoring System in Water Treatment Plants (WTPs)," Heliyon, vol. 10, no. 23, p. e40746, 2024, doi: 10.1016/j.heliyon.2024.e40746.
[14] H. Fakhrurroja, E. T. Nuryatno, A. Munandar, M. Fahmi, and N. A. Mahardiono, "Water Quality Assessment Monitoring System Using Fuzzy Logic and the Internet of Things," Journal of Mechatronics, Electrical Power, and Vehicular Technology, vol. 14, no. 2, pp. 198-207, 2023, doi: 10.14203/j.mev.2023.v14.198-207.
[15] C. N. Insani et al., "IoT-Enabled Real-Time Monitoring and Tsukamoto Fuzzy Logic for Water Quality Classification," Jurnal Teknik Informatika, vol. 6, no. 2, 2025, doi: 10.20884/1.jutif.2025.6.2.5249.
[16] J. J. H. Rojas et al., "Sensors and IoT for Water Quality Monitoring: A Systematic Review," Journal of Robotics and Control, vol. 6, no. 3, 2025, doi: 10.18196/jrc.v6i3.28457.
[17] M. Zen, "Design and Simulation of an IoT-Based Water Quality Monitoring System for Aquaculture Applications Using Python and MQTT," International Conference on Innovation in Engineering, 2025. URL: https://proceeding.pancabudi.ac.id/index.php/ICIE/article/view/700.
[18] R. Baena-Navarro et al., "Intelligent Prediction and Continuous Monitoring of Water Quality Using IoT," Water, vol. 17, no. 1, p. 82, 2025, doi: 10.3390/w17010082.
[19] A. Zuhaer, A. Khandoker, N. Enayet, P. K. P. Partha, and M. A. Awal, "Sustainable Aquaculture: An IoT-Integrated System for Real-Time Water Quality Monitoring Featuring Advanced DO and Ammonia Sensors," Aquacultural Engineering, vol. 109, p. 102620, 2025, doi: 10.1016/j.aquaeng.2025.102620.
[20] OASIS, "MQTT Version 5.0 OASIS Standard," OASIS Standard, 2019. URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.
[21] Wokwi, "Wokwi Documentation: ESP32 Simulation and Online Electronics Simulator," 2026. URL: https://docs.wokwi.com/.