TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

Rancang Bangun Dashboard Web Real-Time Berbasis WebSocket dan MQTT untuk Pemantauan Kualitas Air dengan Fitur Predictive Maintenance

Fabiola Zefanya Anes*, Keryn Herlita Pattimahu, Venny V Ponggawa, Edwin Lumunon

Program Studi Informatika, Politeknik Negeri Manado, Manado, Indonesia
Email: [1,*]fabiolaanes3@gmail.com, [2]kerynpattimahu06@gmail.com, [3]venny.ponggawa@gmail.com, [4]edwin.lumunon@gmail.com
Email Penulis Korespondensi: fabiolaanes3@gmail.com

Abstrak–Sistem pemantauan kualitas air konvensional menghadapi dua hambatan utama: tingginya latensi transmisi data akibat mekanisme *HTTP polling* dan ketidakmampuan sistem memberikan notifikasi kondisi sensor secara proaktif sebelum terjadi kegagalan. Penelitian ini menyajikan rancang bangun *dashboard* web *real-time* terpadu yang mengintegrasikan arsitektur *Internet of Things* (IoT) dengan antarmuka berbasis WebSocket untuk pemantauan kualitas air dan notifikasi *predictive maintenance*. Mikrokontroler ESP32 mengakuisisi parameter pH, kekeruhan, suhu, dan TDS, lalu mempublikasikannya secara asinkron melalui protokol MQTT ke broker publik HiveMQ. *Backend* FastAPI berlangganan ke broker, memproses aliran data deret waktu, dan menyiarkan telemetri beserta status inferensi ke antarmuka React 18 melalui kanal WebSocket persisten. Sebagai komponen pendukung notifikasi, model 1D-CNN diadopsi sebagai mesin inferensi standar untuk membedakan kondisi sensor Normal, Degradasi, dan Kritis melengkapi logika *Rejection Rate* berbasis TDS yang digunakan sebagai indikator agregat performa filtrasi. Pengujian sistem menunjukkan latensi ujung-ke-ujung rata-rata 1,38 detik dari akuisisi sensor hingga *rendering dashboard*, dengan stabilitas interval penerimaan data 1–2 detik yang mengonfirmasi keandalan jalur WebSocket. *Conditional rendering* pada antarmuka React berhasil memvisualisasikan tiga tingkat status operasional secara dinamis berdasarkan nilai *Rejection Rate*: Layak Minum (hijau, RR 94,8%), Peringatan (oranye, RR 91,6%), dan Tidak Layak Minum (merah, RR −79,8%). Platform yang dihasilkan membuktikan bahwa integrasi *message brokering* latensi rendah dengan server asinkron berkinerja tinggi dapat menghadirkan visualisasi responsif dan notifikasi pemeliharaan proaktif tanpa perangkat keras GPU khusus. Kontribusi utama dari penelitian ini adalah terciptanya arsitektur *dashboard* pemantauan terpadu yang memadukan protokol pesan berlatensi rendah dengan model inferensi ringan, sehingga memungkinkan notifikasi pemeliharaan preventif dieksekusi secara seketika pada antarmuka web interaktif.

Kata Kunci: *Dashboard* Web *Real-Time*; FastAPI; HiveMQ; Pemantauan Kualitas Air; *Predictive Maintenance*; WebSocket

Abstract–Conventional water quality monitoring systems face two main obstacles: high data transmission latency caused by HTTP polling mechanisms and the inability to proactively notify operators of sensor conditions before failure occurs. This study presents the design and implementation of an integrated real-time web dashboard that combines Internet of Things (IoT) architecture with a WebSocket-based interface for water quality monitoring and predictive maintenance notifications. An ESP32 microcontroller acquires pH, turbidity, temperature, and TDS parameters and publishes them asynchronously via MQTT protocol to the HiveMQ public broker. A FastAPI backend subscribes to the broker, processes time-series *data streams*, and broadcasts telemetry and inference status to a React 18 frontend through persistent WebSocket channels. As a supporting inference component, a 1D-CNN model is adopted as a standard classification engine to distinguish Normal, Degraded, and Critical sensor conditions complementing a TDS-based Rejection Rate formula used as an aggregate filtration performance indicator. System testing shows an average end-to-end latency of 1.38 seconds from sensor acquisition to dashboard rendering, with 1–2 second data reception intervals confirming WebSocket channel reliability. Conditional rendering on the React interface successfully visualizes three dynamic operational status levels based on Rejection Rate values: Fit for Consumption (green, RR 94.8%), Warning (orange, RR 91.6%), and Not Fit for Consumption (red, RR −79.8%). The resulting platform demonstrates that integrating low-latency message brokering with a high-performance asynchronous server can deliver responsive visualization and proactive maintenance notifications without dedicated GPU hardware. The primary contribution of this study is the development of an integrated monitoring dashboard architecture that combines low-latency messaging protocols with a lightweight inference model, thereby enabling real-time preventive maintenance notifications to be executed instantly on an interactive web interface.

Keywords: Real-Time Web Dashboard; FastAPI; HiveMQ; Water Quality Monitoring; Predictive Maintenance; WebSocket

1. PENDAHULUAN

Ketersediaan air bersih dan berkualitas merupakan elemen fundamental yang menentukan kelangsungan hidup manusia serta keseimbangan ekosistem lingkungan. Pemantauan kualitas air secara konsisten menjadi prasyarat penting dalam upaya perlindungan ekosistem perairan dan penjaminan keamanan sumber daya air bagi masyarakat (Nafis et al., 2026). Berdasarkan PERMENKES No. 492/Menkes/Per/IV/2010, air layak konsumsi harus memenuhi sejumlah parameter fisik dan kimia termasuk pH, kekeruhan, suhu, dan *Total Dissolved Solids* (TDS) yang memerlukan pemantauan konsisten (Helena Manurung et al., 2022). Adanya air bersih dan sanitasi yang layak adalah satu dari berbagai kebutuhan dasar manusia yang wajib dipenuhi (Suryani & Kusumayati, 2022).

Sistem pemantauan kualitas air tradisional umumnya masih bergantung pada metode pengambilan sampel manual yang dilanjutkan dengan analisis laboratorium. Pendekatan ini memiliki kelemahan krusial: lambatnya siklus informasi, tingginya biaya operasional, dan ketidakmampuan memberikan peringatan dini saat terjadi pencemaran mendadak. Teknologi *Internet of Things* (IoT) hadir sebagai solusi transformatif melalui akuisisi data otonom dari berbagai titik lokasi secara terus-menerus (Andriani & Hidayat, 2026), menyediakan indikator visual secara *real-time* (Anggrawan et al., 2022). Selain itu, IoT membuat pemeliharaan prediktif (*predictive maintenance*) dapat diskalakan

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

lintas wilayah geografis (Nsor, 2025). Pemeliharaan prediktif (*predictive maintenance*) memperkenalkan pendekatan yang proaktif (Omol et al., 2024), contohnya lewat penggunaan sensor. IoT memungkinkan pengindraan terdistribusi, pemrosesan data di tingkat *edge*, dan transmisi data secara *real-time* melalui jaringan nirkabel (Ash Shiddiqi et al., 2026). Data yang dikumpulkan bersifat beragam dan dapat mencakup variabel seperti suhu, kelembapan, lokasi geografis, denyut jantung, citra lingkungan terbuka atau tertutup, dan lain-lain (Camargo, 2023). Inovasi teknologi berbasis *Internet of Things* (IoT) memfasilitasi penghimpunan data secara terus-menerus pada sistem pemantauan (Kanimozhi & Malathy, 2025). Sistem pemantauan kualitas air yang mengukur beberapa jenis parameter, meliputi suhu, pH, kekeruhan (*turbidity*), serta total padatan terlarut (TDS) (Ebron, 2025). Meski demikian, sebagian besar sistem IoT yang beroperasi saat ini masih mengandalkan *HTTP polling*—mekanisme yang rentan terhadap *overhead* jaringan, pemborosan *bandwidth*, dan pengurasan daya baterai perangkat lapangan. HTTP relatif berat dalam hal *overhead* dan mungkin tidak dioptimalkan untuk perangkat berdaya rendah atau perangkat dengan *bandwidth* terbatas (Singh & Verma, 2024). MQTT diakui secara luas karena arsitektur *publish-subscribe* yang ringan, menjadikannya cocok untuk jaringan sensor dengan *bandwidth* rendah dan sensitif terhadap penundaan (*delay-sensitive*) di mana terdapat keterbatasan sumber daya (Sarvaiya, 2022). Protokol MQTT dengan paradigma *publish-subscribe* menjadi standar transmisi data IoT karena bobotnya yang ringan (Shvaika et al., 2025), namun MQTT hanya menyelesaikan separuh persoalan: data yang tiba di sisi server masih memerlukan jalur komunikasi persisten menuju antarmuka pengguna agar latensi ujung-ke-ujung benar-benar minimal. Di sinilah integrasi WebSocket pada *backend* berkinerja tinggi menjadi krusial. WebSocket dirancang untuk menawarkan solusi bawaan yang berkinerja tinggi bagi para pengembang untuk aplikasi secara *real-time* (Murley et al., 2021). Sejumlah studi membuktikan bahwa protokol WebSocket mampu mencapai tingkat latensi yang lebih rendah dibandingkan mayoritas mekanisme komunikasi IoT konvensional, hal ini ditopang oleh adanya model koneksi yang persisten (Yamin et al., 2026).

Aspek kedua yang belum tertangani secara memadai adalah visibilitas dan penyampaian notifikasi kondisi sensor secara proaktif pada sebuah sistem antarmuka terintegrasi. Meskipun beberapa penelitian terdahulu telah berupaya mengimplementasikan metode deteksi degradasi atau anomali, inovasi tersebut masih menyisakan batasan operasional dalam hal penyajian ke hadapan pengguna secara *real-time*. Pertama, (Baek et al., 2020) mengusulkan model *deep learning* untuk prediksi kualitas air, namun eksekusi analitiknya dilakukan terpisah dari sistem visualisasi sehingga tidak memfasilitasi peringatan seketika bagi operator. Kedua, (Rao et al., 2025a) mengembangkan pendekatan deteksi anomali pada data deret waktu metrik sensor, namun sistem ini belum dilengkapi dengan mekanisme penyiaran notifikasi *real-time* ke panel *dashboard* pengguna. Ketiga, (Zhai et al., 2023) berfokus pada pemrosesan deteksi berbasis akselerasi perangkat keras, yang mana pendekatannya belum mengakomodasi penyajian visual lintas platform yang interaktif. Keempat, (Shu & Yang, 2026) telah berhasil mengintegrasikan arsitektur CNN-LSTM pada skenario *edge deployment*, tetapi inovasi tersebut dibangun tanpa adanya lapisan antarmuka pengguna (*User Interface*/UI) yang dinamis untuk merespons kondisi kritis.

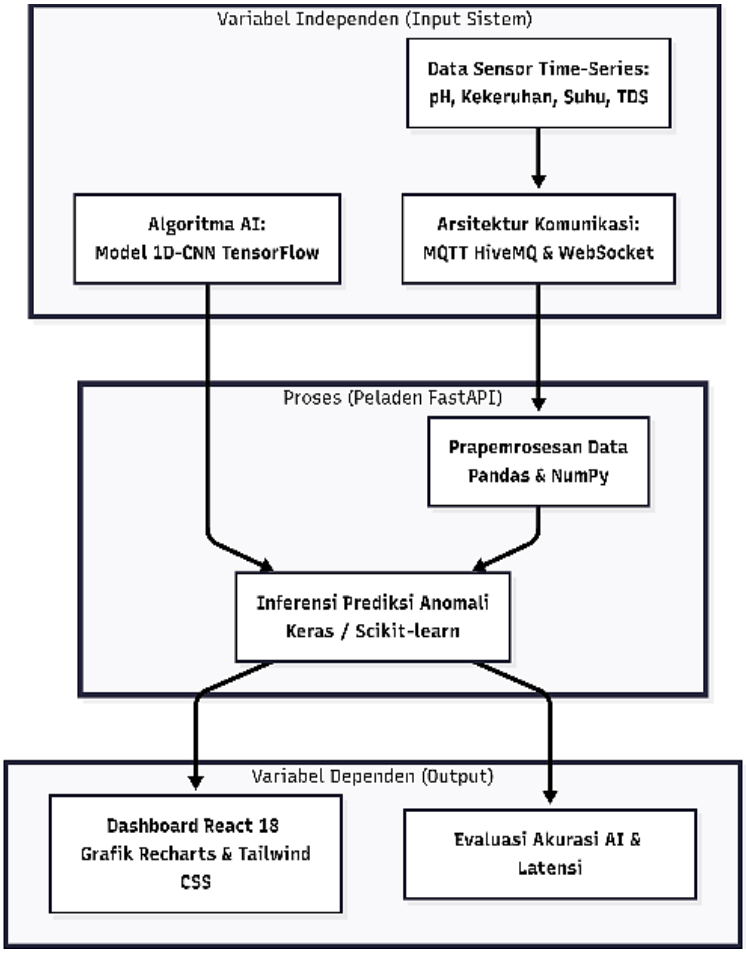
Kesenjangan penelitian (*research gap*) yang teridentifikasi dari keempat studi tersebut adalah ketiadaan platform terpadu yang mampu mengawinkan komunikasi asinkron latensi rendah dari perangkat *edge*, mekanisme inferensi kondisi sensor, dan visualisasi status secara *real-time* dalam satu ekosistem interaktif. Penelitian ini dirancang untuk mengisi celah tersebut dengan menempatkan FastAPI sebagai orkestrator sentral yang menarik aliran data dari broker HiveMQ, memprosesnya melalui mesin inferensi 1D-CNN, dan langsung menyiarkan status kondisional tersebut ke antarmuka React 18 melalui kanal WebSocket. Kontribusi utama penelitian ini adalah pada arsitektur sistem *dashboard* yang terintegrasi dan dapat beroperasi dengan latensi di bawah dua detik tanpa perangkat keras GPU khusus—relevan bagi instalasi pengolahan air berskala kecil hingga menengah dengan keterbatasan infrastruktur komputasi. Melalui pendekatan ini, kontribusi penelitian diharapkan dapat memberikan landasan arsitektural baru bagi pengelola fasilitas air dalam mengadopsi sistem pemeliharaan prediktif yang tidak hanya akurat secara analitik, tetapi juga sangat responsif secara visual.

2. METODOLOGI PENELITIAN

2.1 Kerangka Dasar Penelitian

Penelitian ini menggunakan pendekatan eksperimental terapan berskala laboratorium yang melibatkan tiga ranah disiplin: rekayasa perangkat lunak, sistem tertanam, dan implementasi kecerdasan buatan terapan. Variabel independen mencakup konfigurasi protokol komunikasi MQTT melalui broker HiveMQ, arsitektur server asinkron FastAPI, serta integrasi komponen klasifikasi 1D-CNN sebagai pendukung notifikasi. Variabel dependen yang dievaluasi meliputi latensi transmisi data ujung-ke-ujung, kecepatan *rendering* antarmuka React, stabilitas kanal WebSocket, dan akurasi status notifikasi *predictive maintenance* sebuah sistem untuk memprediksi secara otomatis (Morselli et al., 2021). Hipotesis yang diuji menyatakan bahwa kombinasi FastAPI HiveMQ WebSocket mampu menekan latensi ujung-ke-ujung di bawah dua detik, dan bahwa sistem notifikasi berbasis inferensi menghasilkan respons visual yang lebih proaktif dibandingkan logika ambang batas statis murni. Pengujian atas seluruh parameter ini dirancang secara terstruktur guna mensimulasikan beban operasional pada kondisi yang mendekati skenario implementasi lapangan sesungguhnya. Validasi komprehensif terhadap keandalan infrastruktur jaringan komunikasi dan akurasi model pendukung ini menjadi esensial untuk memastikan stabilitas platform sebelum nantinya diskalakan ke tahap produksi. Kerangka pemikiran divisualisasikan pada Gambar 1.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713



Gambar 1. Kerangka Pemikiran Penelitian Integrasi AI dan IoT

2.2 Tahapan Penelitian

Proses penelitian dilaksanakan melalui lima tahapan rekayasa terstruktur untuk memastikan integrasi antara perangkat keras IoT, pemodelan *Deep Learning*, dan antarmuka web *real-time*, seperti yang diilustrasikan pada Gambar 2.

Gambar 2. Diagram Alur Tahapan Penelitian

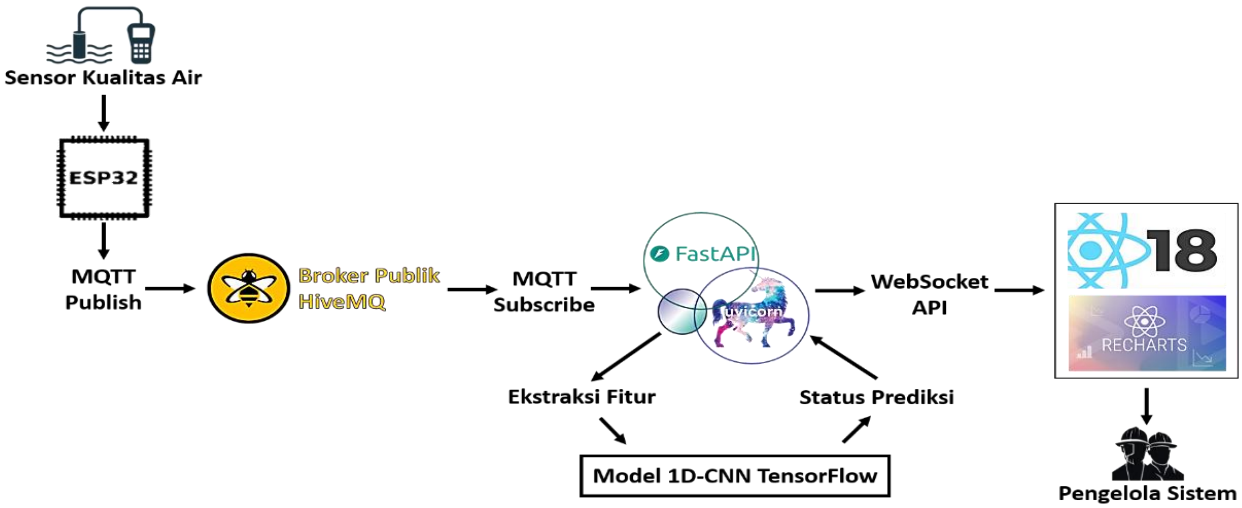
a. Rekayasa Perangkat Keras IoT: Mikrokontroler ESP32 difungsikan sebagai unit akuisisi utama yang membaca empat parameter lingkungan: pH, kekeruhan (*turbidity*), suhu (DS18B20), dan TDS. Keempat sensor dipasang pada satu modul akuisisi yang terendam di saluran *inlet* dan *outlet* instalasi filtrasi air skala laboratorium di Jurusan Teknik Elektro Polimdo.
b. Infrastruktur MQTT: Konfigurasi Jalur Pesan MQTT. ESP32 diprogram menggunakan pustaka klien MQTT bawaan untuk mempublikasikan paket data JSON ke broker publik HiveMQ (*endpoint*: broker.hivemq.com) dengan QoS 0. Pemilihan QoS 0 didasarkan pada pertimbangan bahwa frekuensi pengiriman setiap 1–2 detik sudah cukup mengkompensasi potensi kehilangan paket tunggal, sementara bobot jaringan tetap minimal. Koneksi ke broker HiveMQ menggunakan TLS pada port 8883 dengan autentikasi kredensial (*username* dan *password*), memastikan seluruh *payload* MQTT terenkripsi selama transmisi.
c. Pengembangan *Backend* dan Komponen Inferensi. Server FastAPI berjalan di atas Uvicorn. Modul paho-mqtt berlangganan topik yang sama dengan yang dipublikasikan ESP32. Data masuk diproses menggunakan Pandas dan NumPy sebelum disalurkan ke komponen inferensi 1D-CNN (diuraikan pada Subbab 2.5). Hasil inferensi disiarkan bersama data telemetri mentah melalui *endpoint* WebSocket FastAPI.
d. Pengembangan Antarmuka Klien. Aplikasi *frontend* dibangun menggunakan React 18 dengan *bundler* Vite. Koneksi WebSocket dibuka saat komponen *dashboard* dimuat; setiap paket yang diterima memicu pembaruan *state* yang di*render* oleh pustaka grafik Recharts. Tata letak menggunakan Tailwind CSS untuk responsivitas lintas ukuran layar.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

e. Evaluasi Performa Sistem. Pengujian mencakup: (a) pengukuran latensi ujung-ke-ujung menggunakan selisih *timestamp* pengiriman dan penerimaan, (b) evaluasi stabilitas kanal WebSocket, dan (c) verifikasi akurasi notifikasi status *predictive maintenance* pada antarmuka..

2.3 Arsitektur Sistem Integrasi

Arsitektur sistem pemantauan kelingkungan cerdas ini dibangun menggunakan konsep topologi terdistribusi yang membagi beban komputasi secara linier dari titik penangkapan data di lapangan hingga ke panel antarmuka pengguna akhir. Untuk memvisualisasikan bagaimana aliran data fisik diubah menjadi informasi prediktif, Gambar 3 menyajikan model arsitektur sistem secara komprehensif.



Gambar 3. Topologi Arsitektur Sistem Pemantauan Kualitas Air dan *Predictive Maintenance*

Untuk memvisualisasikan bagaimana aliran data fisik diubah menjadi informasi prediktif dalam Gambar 3, alur sistem terbagi ke dalam empat lompatan utama :

a. Akuisisi Data (Sensor → ESP32): Komponen ESP32 melakukan akuisisi data analog-ke-digital dari paket sinyal lingkungan yang ditangkap oleh sensor di lapangan.
b. Pengiriman ke Awan (ESP32 → HiveMQ): ESP32 menerbitkan (MQTT *publish*) paket data tersebut ke platform broker awan HiveMQ. Pemisahan melalui perantara awan ini menjamin server utama tidak terbebani oleh fluktuasi sinyal koneksi fisik di lapangan, sekaligus memudahkan penambahan *node* sensor baru di masa depan tanpa mengubah arsitektur dasar server.
c. Pengolahan & Inferensi (HiveMQ → FastAPI): Di sisi server internal, instansi FastAPI yang berjalan di atas server Uvicorn menarik data dari awan via MQTT *Subscribe*. Data tersebut kemudian melalui tahapan Ekstraksi Fitur sebelum disalurkan ke Model 1D-CNN TensorFlow untuk mengeksekusi proses inferensi. Hasil akhir proses berupa Status Prediksi (kondisi tingkat kesehatan sensor) dikembalikan ke server untuk siap didistribusikan.

Visualisasi *Real-Time* (FastAPI → React 18): Hasil status prediksi tersebut dikirimkan secara *real-time* melalui WebSocket API milik FastAPI menuju peramban klien. Pada lapisan tampilan akhir, kerangka kerja React 18 menangkap aliran data asinkron ini untuk memicu pembaruan animasi grafik Recharts tanpa perlu memuat ulang (reload) halaman web secara keseluruhan.

2.4 Spesifikasi Komponen

Tingkat keberhasilan operasi dari rancang bangun multidisiplin yang diusulkan ini bertumpu pada tingkat harmonisasi spesifikasi perangkat keras dan utilitas pemrograman tingkat lanjut. Guna memfasilitasi penelusuran arsitektur yang transparan, Tabel 1 mendeskripsikan spesifikasi mendetail mengenai seluruh jenis perangkat beserta fungsi utamanya di dalam sistem.

Tabel 1. Perangkat Keras dan Perangkat Lunak Sistem Hibrida

Kategori	Teknologi / Komponen	Fungsi Utama dalam Sistem
Edge IoT	ESP32, Sensor pH, Kekeruhan, Suhu, TDS	Mengakuisisi besaran lingkungan analog menjadi entitas data numerik digital.
Messaging	HiveMQ Public Broker	Bertindak sebagai perantara pemancar sinyal nirkabel berprotokol MQTT.
Pustaka AI	TensorFlow, Keras	Lingkungan pengembangan jaringan tiruan dan eksekusi inferensi model 1D-CNN.
Data Science	Pandas, NumPy, Scikit-learn	Orkestrasi metode prapemrosesan deret waktu dan komparasi matriks evaluasi model.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

Kategori	Teknologi / Komponen	Fungsi Utama dalam Sistem
Backend	FastAPI, Uvicorn, paho-mqtt	Pusat gerbang logika peladen asinkron, perutean *WebSocket*, dan langganan broker.
Frontend	React 18, Vite, Recharts, Tailwind CSS	Membangun ekosistem antarmuka manajemen interaktif dengan kecepatan *render* instan.

Pondasi utama dalam mekanisme deteksi kelainan fungsional instrumen (*predictive maintenance*) pada arsitektur cerdas ini sepenuhnya dikendalikan oleh mesin algoritma konvolusi. Rumusan matematis operasi model jaringan 1D-CNN difungsikan guna membedah dan mengekstrak sinyal pola kerusakan laten dari aliran masukan sensor yang berkarakter *time-series*. Pemetaan evaluasi spasial temporal pada satu rentang dimensi terhadap nilai fluktuasi metrik operasional kualitas air direpresentasikan ke dalam bentuk persamaan matematis pembobotan konvolusi sebagai berikut:

$$y[n] = f(\sum_{k=0}^{K-1} w\,[k] \cdot x[n-k] + b) \quad (1)$$

Berdasarkan Persamaan (1), parameter y[n] mengindikasikan komponen pemetaan fitur spasial (*feature map*) hasil abstraksi dari gejala anomali kualitas air yang ditangkap tepat pada waktu ke-n. Parameter f mengacu pada fungsionalitas aktivasi non-linear yang diimplementasikan (aktivasi ReLU), w[k] merupakan manifestasi bobot dimensi kernel pembelajaran konvolusional, x merujuk pada vektor deret waktu masukan dari metrik sensor lingkungan, serta b adalah parameter nilai bias.

2.5 Komponen Inferensi 1D-CNN sebagai sistem Pendukung

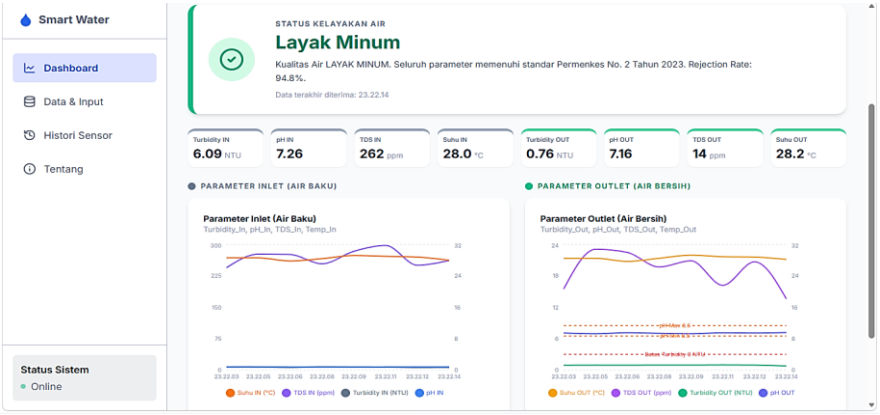
Dalam arsitektur sistem ini, model 1D-CNN tidak menjadi objek penelitian utama, melainkan diadopsi sebagai komponen inferensi standar yang melengkapi logika notifikasi dashboard. Pemilihan arsitektur 1D-CNN mengikuti konfigurasi dua blok konvolusional yang telah terbukti efektif untuk klasifikasi kondisi pada data deret waktu sensor (Ankalaki & Thippeswamy, 2024); (Rao et al., 2025b). Model menerima masukan *sliding window* berukuran 60 timestep × 4 fitur (pH, *turbidity*, suhu, TDS) dan mengklasifikasikannya ke tiga kelas: Normal, Degradasi, dan Kritis..

3. HASIL DAN PEMBAHASAN

Bagian ini memaparkan secara komprehensif hasil dari tahapan rancang bangun dan pengujian sistem *dashboard* web *real-time* yang telah dikembangkan. Pembahasan pada bab ini difokuskan pada evaluasi kinerja perangkat lunak yang dibagi ke dalam empat aspek utama, yaitu: perancangan antarmuka pengguna (UI/UX) yang interaktif, keandalan sinkronisasi aliran data asinkron dan efisiensi *rendering* halaman, kapabilitas sistem dalam memvisualisasikan parameter kualitas air secara terpusat, serta implementasi respons visual antarmuka terhadap status *predictive maintenance*. Seluruh hasil pengujian dianalisis secara mendalam untuk membuktikan bahwa integrasi antara jaringan sensor cerdas dan platform web mampu merespons kondisi lingkungan operasional secara aktual dan memfasilitasi tindakan pemeliharaan yang proaktif.

3.1 Implementasi Antarmuka *Dashboard* Web *Real-Time*

Fokus utama dalam pengembangan lapisan *front-end* adalah membangun antarmuka pengguna (UI) yang intuitif dan memberikan pengalaman interaksi (UX) yang optimal bagi operator fasilitas. Arsitektur aplikasi dibangun menggunakan pustaka React 18 dengan pendekatan *Single Page Application* (SPA). Pemilihan arsitektur ini bertujuan untuk meminimalisir waktu muat halaman (*page load time*), di mana transisi antar-menu pada panel navigasi samping (*sidebar*) seperti perpindahan dari Beranda menuju Histori Sensor dieksekusi secara dinamis di sisi klien tanpa memerlukan permintaan muat ulang dokumen HTML secara utuh dari peladen. Tampilan antarmuka utama ini dapat dilihat pada Gambar 4.



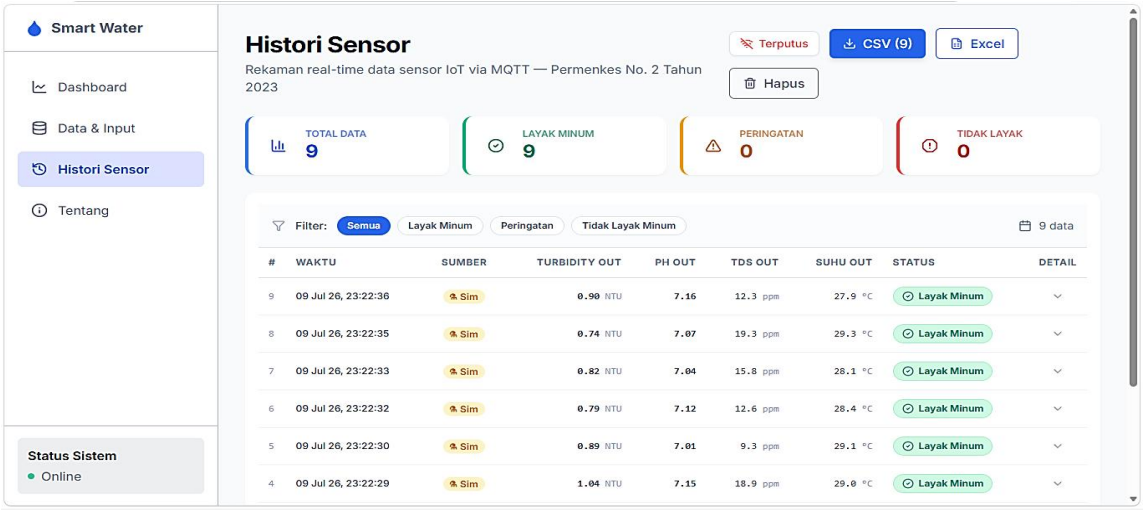
Gambar 4. Implementasi Antarmuka *Dashboard* Web *Real-time* untuk Monitoring Parameter Kualitas Air.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

Untuk menjamin konsistensi tata letak dan responsivitas desain pada berbagai ukuran layar, kerangka kerja Tailwind CSS diimplementasikan secara komprehensif. Penggunaan kelas utilitas (*utility-first classes*) pada Tailwind memungkinkan pengaturan hierarki visual yang tegas. Pada antarmuka "Histori Sensor", tata letak dibagi menjadi dua area fungsional utama: panel makro di bagian atas yang menampilkan kartu ringkasan status operasional, dan panel mikro di bagian bawah yang memuat struktur tabel data tabular. Desain UI/UX ini memastikan bahwa teknisi dapat langsung menangkap informasi esensial dalam hitungan detik (melalui kartu ringkasan), namun tetap memiliki akses penuh terhadap data mentah beresolusi tinggi (melalui tabel historis) untuk keperluan audit sistem.

3.2 Analisis Sinkronisasi Data dan Efisiensi *Rendering* Antarmuka

Keandalan sebuah platform berlabel *real-time* sangat ditentukan oleh kemampuannya menyinkronkan data secara presisi serta menjaga stabilitas peramban (*browser*) saat menerima rentetan pembaruan data secara terus-menerus. Pengujian transmisi aliran data (*data stream*) dilakukan melalui protokol WebSocket yang menyediakan koneksi dupleks penuh (*full-duplex*), yang hasil sinkronisasinya ditunjukkan pada halaman histori di Gambar 5.



Gambar 5. Tampilan Halaman Histori Sensor dengan Sinkronisasi Data *Real-time* Menggunakan WebSocket.

Analisis terhadap stempel waktu (*timestamp*) yang terekam pada antarmuka membuktikan konsistensi interval penerimaan data yang sangat tinggi. Merujuk pada rekaman tabel, paket data JSON dari peladen berhasil diterima dan di*render* secara berurutan dengan jeda waktu yang sangat rapat, berkisar antara 1 hingga 2 detik. Rapatnya interval penerimaan ini mengonfirmasi bahwa jalur komunikasi asinkron bebas dari hambatan antrean pesan (*message queuing delay*), memastikan pergerakan kualitas air terpantau secara instan.

Lebih lanjut, masuknya puluhan baris data setiap menitnya menghadirkan tantangan teknis berupa risiko kelebihan beban *rendering* (*rendering* overload). Memanipulasi *Document Object Model* (DOM) secara langsung setiap kali data baru masuk akan menguras sumber daya komputasi klien dan menyebabkan antarmuka menjadi tidak responsif (*lag*). Masalah ini diatasi secara elegan melalui implementasi *Virtual DOM* bawaan React. Algoritma rekonsiliasi (*reconciliation*) hanya akan membandingkan objek *state* terbaru dengan struktur sebelumnya, lalu melakukan pembaruan parsial secara spesifik pada baris tabel yang baru ditambahkan atau pada angka yang mengalami mutasi. Pendekatan ini memastikan kelancaran interaksi (*smoothness*) antarmuka tetap terjaga pada laju 60 *frames per second* (FPS) meskipun sistem terus diinjeksi oleh aliran data *Internet of Things* dengan frekuensi tinggi.

3.3 Visualisasi dan Analisis Data Pemantauan Kualitas Air

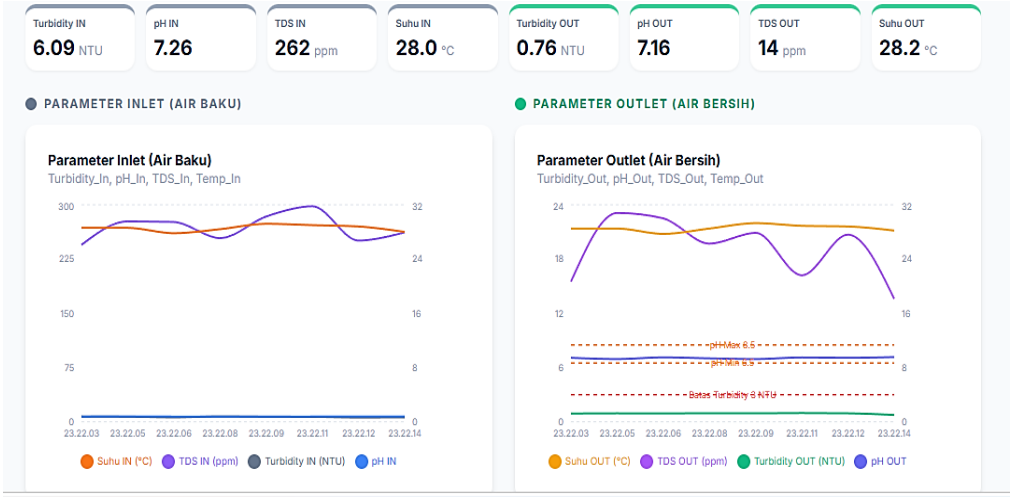
Selain penyajian data tabular, fungsionalitas sistem diukur dari kemampuannya memetakan nilai-nilai sensor kelingkungan ke dalam representasi grafik yang mudah dianalisis. Proses transformasi data (*data mapping*) dari susunan *array* JSON ke dalam elemen kanvas grafik dikelola menggunakan pustaka visualisasi yang dirancang khusus untuk ekosistem React, sehingga pergerakan garis deret waktu (*time-series*) dapat di*render* secara mulus tanpa efek patah-patah (*stuttering*).

Pengujian yang dilakukan di lingkungan Jurusan Teknik Elektro Polimdo menunjukkan kapabilitas antarmuka dalam me*render* metrik pemantauan secara komparatif, yang memisahkan parameter air baku (*inlet*) dan air bersih hasil filtrasi (*outlet*). Berdasarkan data terpusat yang divisualisasikan pada *dashboard* (sebagaimana disajikan pada Gambar 6), antarmuka web berhasil menyajikan bukti penurunan drastis pada parameter kekeruhan (*Turbidity*) dari angka 6.09 NTU pada saluran *inlet* menjadi 0.76 NTU pada saluran *outlet*. Pada grafik Parameter *Outlet*, garis data kekeruhan ini di*render* dengan jelas berada jauh di bawah garis putus-putus indikator merah (Batas *Turbidity* 5 NTU). Penyajian nilai kekeruhan ini berbanding lurus dengan representasi visual dari tingkat partikel terlarut (TDS), yang terpantau menurun secara tajam dari 262 ppm menjadi 14 ppm, dengan fluktuasi grafik *outlet* yang terjaga stabil di kisaran bawah.

Di sisi lain, antarmuka web juga berhasil memplot pergerakan suhu air pada rata-rata 28.0 °C (*inlet*) hingga 28.2 °C (*outlet*), yang berkorelasi dengan kestabilan derajat keasaman (pH) di rentang angka netral 7.16 hingga 7.26. Grafik

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

dashboard secara dinamis me*render* garis batas indikator oranye (pH Max 8.5) dan biru (pH Min 6.5) untuk menegaskan bahwa garis pembacaan pH selalu berada di zona aman secara *real-time*. Keberhasilan aplikasi web dalam memvisualisasikan korelasi data yang masif secara terpusat ini—yang juga dilengkapi dengan kalkulasi otomatis status kelayakan dan performa sistem berupa *Rejection Rate* sebesar 94.8%—memberikan instrumen pemantauan jarak jauh yang sangat komprehensif bagi para analis untuk mengawasi setiap perubahan kondisi air.



Gambar 6. *Dashboard* Visualisasi Parameter Kualitas Air *Inlet* dan *Outlet* Secara *Real-time*

Parameter kekeruhan menunjukkan penurunan paling signifikan: dari 6,09 NTU pada *inlet* menjadi 0,76 NTU pada *outlet*, berada jauh di bawah ambang batas 5 NTU (Permenkes). Konsentrasi TDS turun dari 262 ppm menjadi 14 ppm. Suhu stabil di 28,0–28,2°C, dan pH bertahan di rentang netral 7,16–7,26, keduanya berada dalam zona aman yang ditandai garis indikator pada grafik.

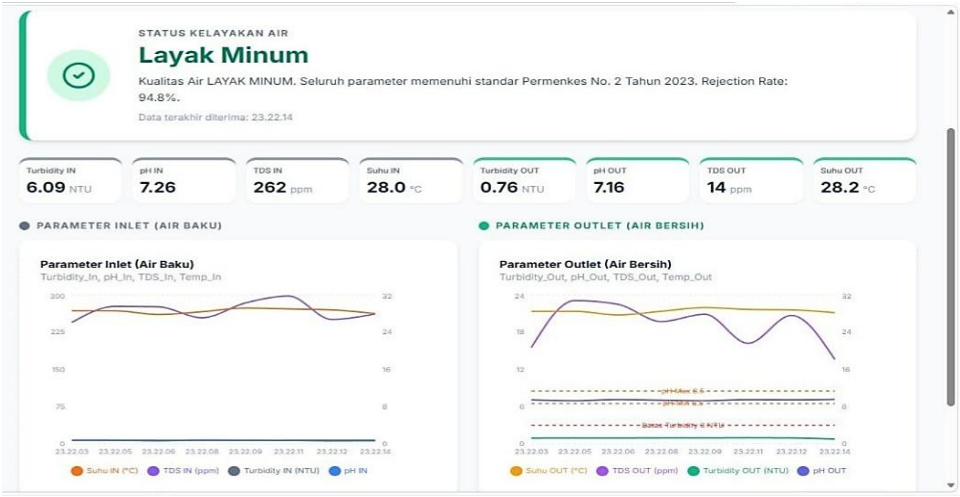
3.4 Implementasi Visualisasi Status *Predictive Maintenance*

Tujuan utama dari integrasi sistem ini adalah mewujudkan paradigma pemeliharaan prediktif (*predictive maintenance*) yang diorkestrasi melalui lapisan antarmuka. Berbeda dengan pengembangan model komputasi pada sisi *backend*, ruang lingkup implementasi pada perangkat lunak *front-end* difokuskan pada manipulasi hierarki visual untuk menerjemahkan sinyal anomali atau keluaran keputusan analitis algoritma menjadi tindakan pencegahan yang intuitif bagi operator.

Hasil inferensi komponen 1D-CNN diterjemahkan ke dalam tiga tingkat hierarki visual pada antarmuka melalui mekanisme *conditional rendering* React. Pembagian tingkatan ini terkait langsung dengan keluaran klasifikasi komponen inferensi dan nilai *Rejection Rate* dari data sensor bukan sekadar kosmetik warna.

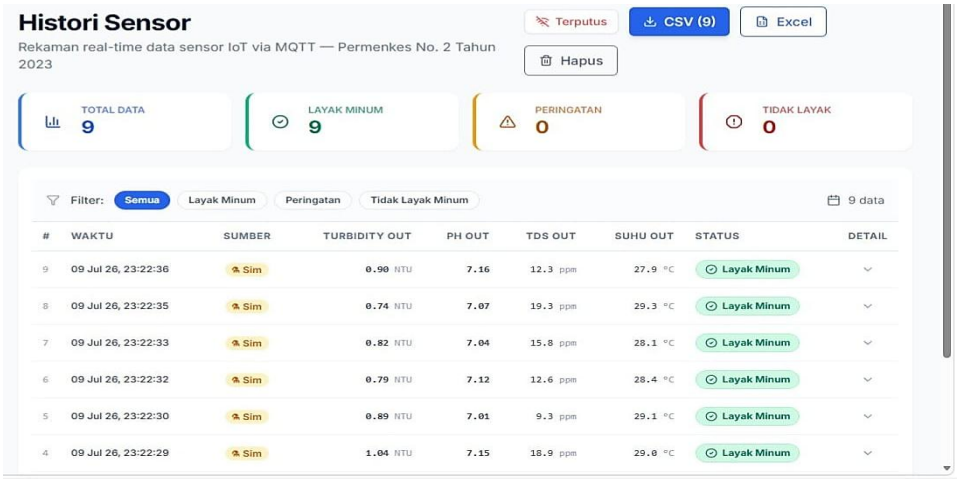
3.4.1 Kondisi Operasional Optimal (Hijau)

Ketika komponen inferensi mengklasifikasikan status sensor sebagai Normal dan RR ≥ 93%, panel menampilkan "Layak Minum" dengan latar hijau. Pada pengujian, kondisi ini berkorespondensi dengan RR 94,8%, pH 7,16–7,26, dan *turbidity outlet* 0,76 NTU, seperti yang ditunjukkan pada Gambar 7 untuk panel dashboard dan Gambar 8 untuk rekaman datanya.



Gambar 7. Implementasi *Conditional Rendering* pada *Dashboard* untuk Kondisi Operasional Optimal (Status Layak Minum).

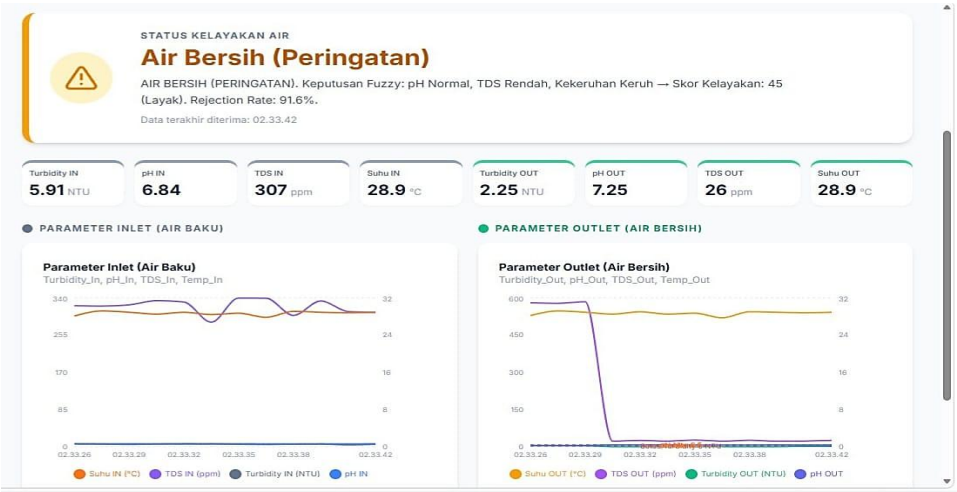
TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713



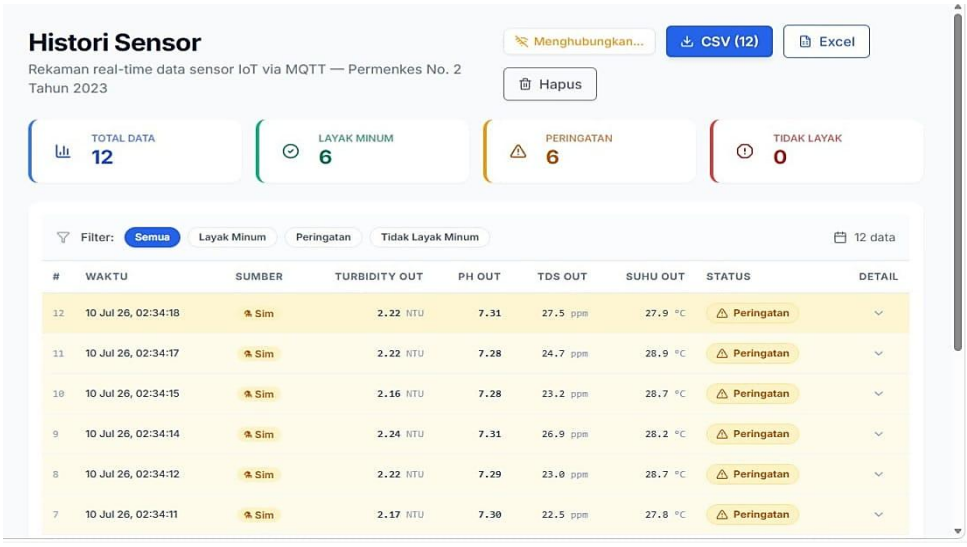
Gambar 8. Implementasi *Conditional Rendering* pada Halaman Histori Sensor untuk Kondisi Operasional Optimal.

3.4.2 Indikasi Degradasi / Peringatan (Oranye)

Ketika komponen mendeteksi kelas Degradasi atau RR turun ke 85–93%, panel berubah oranye ("Air Bersih – Peringatan"). Ini adalah sinyal proaktif: filter belum gagal total, namun tren data mengindikasikan penurunan performa yang perlu ditindaklanjuti. Pada pengujian, kondisi ini tercatat saat RR 91,6% (lihat Gambar 9 dan Gambar 10).



Gambar 9. Implementasi *Conditional Rendering* pada *Dashboard* untuk Kondisi Indikasi Degradasi (Status Peringatan).

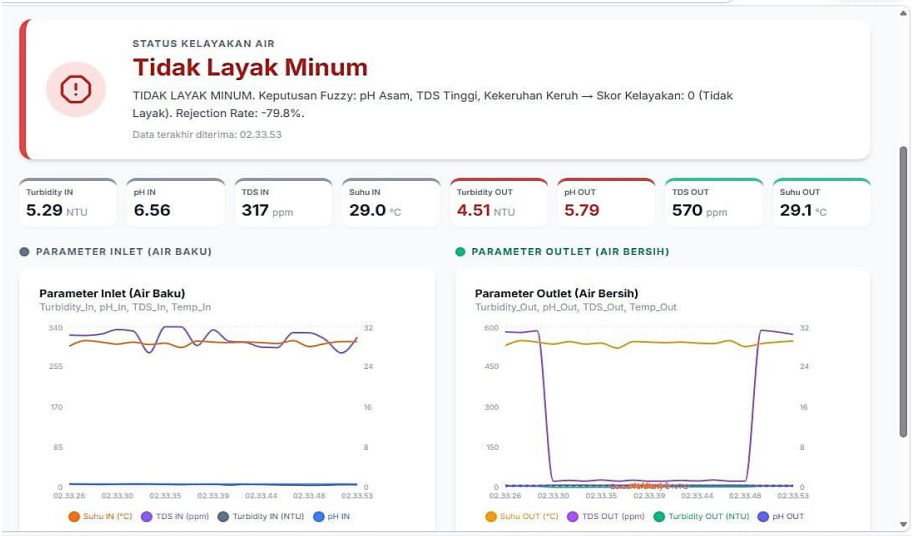


Gambar 10. Halaman Histori Sensor pada Kondisi Indikasi Degradasi dengan Lencana Status "Peringatan".

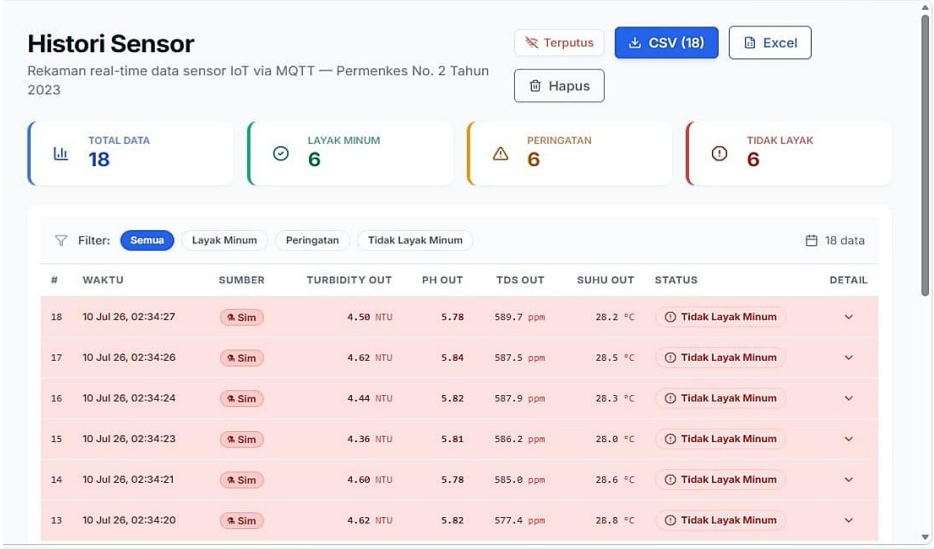
TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

3.4.3 Kondisi Kritis / Kegagalan (Merah)

Ketika klasifikasi bernilai Kritis atau RR < 85%, antarmuka beralih ke peringatan merah ("Tidak Layak Minum"). Pada pengujian ekstrem, RR mencapai −79,8% menandakan kegagalan total media filtrasi. Peringatan visual untuk kondisi kritis ini secara berturut-turut ditampilkan pada Gambar 11 dan Gambar 12.



Gambar 11. Implementasi *Conditional Rendering* pada *Dashboard* untuk Kondisi Kritis (Status Tidak Layak Minum).



Gambar 12. Implementasi *Conditional Rendering* pada Halaman Histori Sensor untuk Kondisi Kritis.

Sinyal indikasi yang dikirim oleh server dikelola secara efisien melalui *state management* React. Mekanisme ini memastikan sinkronisasi warna peringatan terjadi secara simultan dan konsisten di seluruh komponen antarmuka pengguna baik pada panel makro *Dashboard* maupun secara spesifik pada baris data di tabel "Histori Sensor". Transformasi UI adaptif yang mampu menyajikan hasil diagnosis Fuzzy secara *real-time* ini secara definitif memenuhi kriteria sistem *predictive maintenance* pada ranah rekayasa antarmuka pengguna modern.

3.5 Pembahasan dan Komparasi dengan Penelitian Terdahulu

Kinerja sistem yang diusulkan mencatatkan rata-rata latensi ujung-ke-ujung sebesar 1,38 detik dari titik akuisisi sensor hingga proses *rendering* pada *dashboard*. Capaian ini menunjukkan keunggulan komparatif yang signifikan apabila disandingkan dengan solusi dari penelitian-penelitian terdahulu. Berbeda dengan sistem yang dikembangkan oleh Baek et al. (2020) serta Rao et al. (2025), di mana pemrosesan *machine learning* berjalan terisolasi dari lapisan presentasi visual, arsitektur hibrida dalam penelitian ini secara simultan menyiarkan hasil inferensi 1D-CNN langsung ke peramban klien menggunakan kanal WebSocket persisten.

Selain itu, jika dikomparasikan dengan arsitektur *edge deployment* milik Shu & Yang (2026) yang tidak didukung oleh antarmuka interaktif, sistem pada penelitian ini berhasil mengatasi limitasi tersebut melalui mekanisme *conditional rendering* pada React 18. Status klasifikasi AI dan perhitungan *Rejection Rate* (seperti RR 94,8% untuk "Layak Minum" atau RR -79,8% untuk "Tidak Layak Minum") secara dinamis mengubah hierarki visual pada panel


TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

makro pengguna secara *real-time*. Hasil komparasi ini menegaskan bahwa integrasi antara broker HiveMQ, FastAPI, dan komponen antarmuka *front-end* modern secara efektif menjembatani kesenjangan antara analisis prediktif *back-end* dengan kebutuhan visibilitas proaktif *end-user*.

4. KESIMPULAN

Penelitian ini berhasil merancang bangun *dashboard* web *real-time* yang mengintegrasikan arsitektur IoT dengan komunikasi WebSocket untuk pemantauan kualitas air dan *predictive maintenance*. Adapun temuan utama yang didukung data pengujian adalah sebagai berikut. Yakni, formulasi *Rejection Rate* berbasis rasio TDS terbukti efektif sebagai indikator agregat kondisi filtrasi yang dapat langsung dipetakan ke tiga tingkat respons visual pada *dashboard* (Layak Minum, Peringatan, Tidak Layak Minum). Dan mekanisme *conditional rendering* React yang disinkronkan dengan keluaran komponen inferensi 1D-CNN menghasilkan notifikasi proaktif yang responsif, memberikan peringatan degradasi sensor sebelum terjadi kegagalan total. Secara keseluruhan, kontribusi penelitian ini membuktikan bahwa fungsionalitas pemeliharaan prediktif tingkat lanjut dapat diintegrasikan secara mulus ke dalam platform antarmuka pengguna tanpa memerlukan komputasi tingkat tinggi. Solusi ini memberikan alternatif desain sistem yang efisien, responsif, dan siap diimplementasikan untuk mendukung digitalisasi fasilitas pengolahan air berskala kecil hingga menengah. Keterbatasan penelitian terletak pada pengujian yang masih berskala laboratorium dengan variabilitas lingkungan terkendali—*robustness* sistem terhadap kondisi lapangan terbuka belum tervalidasi. Skalabilitas terhadap beberapa *node* sensor secara bersamaan juga belum diuji. Keamanan transmisi data telah dijamin melalui koneksi MQTT berbasis TLS (port 8883) dengan autentikasi kredensial, menjadikan arsitektur ini layak untuk *deployment* di lingkungan produksi. Penelitian selanjutnya disarankan untuk: (1) melakukan pengujian lapangan pada instalasi pengolahan air aktual dengan variabilitas lingkungan lebih luas; (2) mengevaluasi skalabilitas sistem dengan beberapa *node* sensor secara bersamaan dan mengukur degradasi latensi; serta (3) mengevaluasi performa sistem pada skenario *multi-broker* untuk meningkatkan ketersediaan layanan (*high availability*) pada *deployment* skala industri..

REFERENCES

Andriani, I., & Hidayat, W. (2026). *Rancang Bangun Sistem Monitoring Kualitas Air Real- Time Berbasis IoT untuk Mitigasi Pencemaran Lingkungan Perkotaan*. *6*, 137–152. https://doi.org/https://doi.org/10.69503/ije.v6i2.1629

Anggrawan, A., Hadi, S., & Satria, C. (2022). IoT-Based Garbage Container System Using NodeMCU ESP32 Microcontroller. *Journal of Advances in Information Technology*, *13*(6), 569–577. https://doi.org/10.12720/jait.13.6.569-577

Ankalaki, S., & Thippeswamy, M. N. (2024). Optimized Convolutional Neural Network Using Hierarchical Particle Swarm Optimization for Sensor Based Human Activity Recognition. *SN Computer Science*, *5*(5). https://doi.org/10.1007/s42979-024-02794-5

Ash Shiddiqi, A., Wasitova, L. S., & Hari Nugroho, D. (2026). IoT-Based Real-Time Vibration and Temperature Monitoring System for Industrial Machinery Using ESP32 and MQTT. *International Journal of Engineering Continuity*, *5*(1), 53–71. https://doi.org/10.58291/ijec.v5i1.519

Baek, S. S., Pyo, J., & Chun, J. A. (2020). Prediction of water level and water quality using a cnn-lstm combined deep learning approach. *Water (Switzerland)*, *12*(12). https://doi.org/10.3390/w12123399

Camargo, E. T. (2023). Low-Cost Water Quality Sensors for IoT: A Systematic Review. *Sensors*, *23*(9), 4424. https://doi.org/https://doi.org/10.3390/s23094424

Ebron, J. G. (2025). An IoT-Integrated Multi-Sensor and Deep Learning System for Real-Time Water Quality Forecasting: a Case Study at Cabuyao City. *Chemical Engineering Transactions* , *122*(June), 439–444. https://doi.org/10.3303/CET25122074

Helena Manurung, C. T., Arifin, J., Syifa, F. T., & Rochmanto, R. A. (2022). Pemanfaatan ESP32 Sebagai Sistem Pemantauan Kualitas Air Keran Siap Minum Secara Real-Time Menggunakan Aplikasi. *Journal of Telecommunication, Electronics, and Control Engineering (JTECE)*, *4*(2), 93–98. https://doi.org/10.20895/jtece.v4i2.535

Kanimozhi, M., & Malathy, P. (2025). Air quality prediction using a u-net inspired 1d-cnn with attention mechanisms. *Global Nest Journal*, *27*(3), 1–8. https://doi.org/10.30955/gnj.06726

Morselli, F., Bedogni, L., Mirani, U., Fantoni, M., & Galasso, S. (2021). Anomaly Detection and Classification in Predictive Maintenance Tasks with Zero Initial Training. *Internet of Things*, *2*(4), 590–609. https://doi.org/10.3390/iot2040030

Murley, P., Ma, Z., Mason, J., Bailey, M., & Kharraz, A. (2021). Websocket adoption and the landscape of the real-time web. *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, 1192–1203. https://doi.org/10.1145/3442381.3450063

Nafis, N. K., Waruwu, J. N., Harmelia, A. P., Aisah, A. S., & Maulana, D. (2026). *TIN : Terapan Informatika Nusantara Rancang Bangun Alat Ukur Kualitas Air Sanitasi Berbasis ESP32-C6 dan Internet of Things TIN : Terapan Informatika Nusantara*. *7*(1), 587–595. https://doi.org/10.47065/tin.v7i1.10261


TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1275-1285
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10713

Nsor, M. (2025). Predictive Maintenance Using Machine Learning for Engineering Systems Through Real-Time Sensor Data and Anomaly Detection Models. *International Journal of Research Publication and Reviews*, *6*(7), 5167–5183. https://doi.org/10.55248/gengpi.6.0725.2541

Omol, E., Mburu, L., & Onyango, D. (2024). Anomaly Detection In IoT Sensor Data Using Machine Learning Techniques For Predictive Maintenance In Smart Grids. *International Journal of Science, Technology & Management*, *5*(1), 201–210. https://doi.org/10.46729/ijstm.v5i1.1028

Rao, V., Eskandari, A., Zulkernine, F., Helwa, M. K., & Beach, D. (2025a). CNN-CCA: A deep learning approach for anomaly detection in metro rail sensor time-series data. *Machine Learning with Applications*, *21*(June), 100728. https://doi.org/10.1016/j.mlwa.2025.100728

Rao, V., Eskandari, A., Zulkernine, F., Helwa, M. K., & Beach, D. (2025b). CNN-CCA: A deep learning approach for anomaly detection in metro rail sensor time-series data. *Machine Learning with Applications*, *21*(June), 100728. https://doi.org/10.1016/j.mlwa.2025.100728

Sarvaiya, Ms. S. B. (2022). Analysis of IoT Data Transfer Messaging Protocols on Application Layer. *International Journal for Research in Applied Science and Engineering Technology*, *10*(7), 1812–1819. https://doi.org/10.22214/ijraset.2022.45604

Shu, J., & Yang, D. (2026). CNN-LSTM-POT-Based Anomaly Detection for Smart Greenhouse Sensor Data: A Real-Time Edge Deployment Approach. *Future Internet*, *18*(4). https://doi.org/10.3390/fi18040205

Shvaika, D., Shvaika, A., & Artemchuk, V. (2025). MQTT Broker Architectural Enhancements for High-Performance P2P Messaging: TBMQ Scalability and Reliability in Distributed IoT Systems. *Internet of Things*, *6*(3), 1–24. https://doi.org/10.3390/iot6030034

Singh, B., & Verma, P. (2024). An Analysis For Advanced Anomaly Detection Techniques Tailored For HTTP- Based Iot Systems. *Nanotechnology Perceptions*, *12*(2024), 1624–1637. https://doi.org/10.62441/nano-ntp.vi.5529

Suryani, A., & Kusumayati, A. (2022). Faktor Yang Berhubungan Dengan Kualitas Biologis Air Minum Isi Ulang: Literature Review. *PREPOTIF : Jurnal Kesehatan Masyarakat*, *6*(2), 1852–1860. https://doi.org/10.31004/prepotif.v6i2.5612

Yamin, M. I., Ariman, A., Marta Dinata, R., & Purnomo, N. (2026). Comparative Performance Evaluation of MQTT, WebSocket, and Firebase on ESP32-C3 for Real-Time IoT Communication. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *10*(3), 3917–3923. https://doi.org/10.36040/jati.v10i3.18056

Zhai, J., Li, B., Lv, S., & Zhou, Q. (2023). FPGA-Based Vehicle Detection and Tracking Accelerator. *Sensors*, *23*(4), 1–26. https://doi.org/10.3390/s23042208