

Prototipe Alat Deteksi Banjir Berbasis ESP32, TinyML, dan Notifikasi Telegram

Laily Rahmad Fauzi^{1*}, Imam Suharjo¹

¹*Prodi Informatika Universitas Mercu Buana Yogyakarta, Sleman, Indonesia*

lailyrahmadfauzi@gmail.com*

| Received: 29/12/2025 | Revised: 28/01/2026 | Accepted: 03/02/2026 |

Copyright©2026 by authors. Authors agree that this article remains permanently open access under the terms of the Creative Commons

Abstrak

Banjir merupakan salah satu bencana hidrometeorologi yang paling sering terjadi di Indonesia dan menimbulkan dampak signifikan terhadap keselamatan, aktivitas ekonomi, serta infrastruktur masyarakat. Untuk mendukung peningkatan kesiapsiagaan, dibutuhkan sistem pemantauan tinggi muka air yang mampu bekerja secara real-time, akurat, dan mudah diimplementasikan di lapangan. Penelitian ini bertujuan mengembangkan prototipe sistem pemantauan dan klasifikasi ketinggian air berbasis Internet of Things (IoT) menggunakan mikrokontroler ESP32, sensor ultrasonik HC-SR04, serta model Tiny Machine Learning (TinyML) yang melakukan inferensi secara lokal pada perangkat (on-device inference). Penentuan kondisi air dilakukan berdasarkan kriteria dan threshold ketinggian air yang ditetapkan pada tahap analisis, yang selanjutnya digunakan sebagai dasar pelabelan data dan pembentukan model TinyML. Sistem dilengkapi indikator LED (hijau, kuning, merah), tampilan OLED I2C 0,96 inci sebagai monitoring lokal, serta notifikasi otomatis melalui aplikasi Telegram sebagai peringatan dini jarak jauh. Metode penelitian menggunakan pendekatan Research and Development (R&D) dengan model ADDIE yang meliputi tahap analisis kebutuhan, perancangan sistem, pengembangan perangkat keras dan perangkat lunak, implementasi prototipe, serta evaluasi kinerja sistem. Hasil pengujian menunjukkan bahwa sistem mampu mengukur ketinggian air secara stabil dengan error maksimum sebesar 0,2 cm. Nilai Mean Absolute Error (MAE) sebesar 0,025 cm diperoleh dari 8 kali percobaan pada rentang pengujian 0–22,5 cm. Sistem berhasil mengklasifikasikan kondisi air ke dalam kategori “Aman”, “Waspada” dan “Banjir” menggunakan model TinyML dengan kesesuaian hasil terhadap skenario uji sebesar 100% pada sepuluh kondisi pengujian. Selain itu, mekanisme notifikasi Telegram bekerja sesuai rancangan, di mana pesan peringatan terkirim pada seluruh transisi status peringatan yang diuji (lima dari lima skenario) dan tidak terkirim pada kondisi aman yang stabil. Temuan ini menunjukkan bahwa integrasi IoT dan TinyML berpotensi mendukung sistem pemantauan dan peringatan dini banjir yang responsif dan efisien, serta sesuai diterapkan sebagai prototipe pada skala kecil hingga menengah seperti lingkungan permukiman rawan genangan dengan perubahan ketinggian air yang terukur.

Kata kunci: IoT, ESP32, TinyML, ketinggian air, peringatan banjir

Abstract

Floods are among the most frequent hydrometeorological disasters in Indonesia and cause significant impacts on public safety, economic activities, and community infrastructure. To support improved preparedness, a water level monitoring system that operates in real time, provides accurate measurements, and is easy to implement in the field is required. This study aims to develop a prototype of an Internet of Things (IoT)-based water level monitoring and classification system using an ESP32 microcontroller, an HC-SR04 ultrasonic sensor, and a Tiny Machine Learning (TinyML) model that performs on-device inference. Water condition determination is based on predefined criteria and water level thresholds established during the analysis stage, which are subsequently used for data labeling and TinyML model development. The system is equipped with LED indicators (green, yellow, red), a 0.96-inch I2C OLED display for local monitoring, and an automatic notification mechanism via the Telegram application as a remote early warning system. The research adopts a Research and Development (R&D) approach using the ADDIE model, which includes needs analysis, system design, hardware and software development, prototype implementation, and system performance evaluation. Experimental results show that the system is able to measure water levels stably with a maximum error of 0.2 cm. A Mean Absolute Error (MAE) of 0.025 cm was obtained from eight experimental trials within a test range of 0–22.5 cm. The system successfully classified water conditions into “Safe,” “Alert,” and “Flood” categories using the TinyML model, achieving 100% agreement with test scenarios across ten test conditions. Furthermore, the Telegram notification mechanism operated as designed, with warning messages sent during all tested warning status transitions (five out of five scenarios) and not sent under stable safe conditions. These findings indicate that the integration of IoT and TinyML has strong potential to support responsive and efficient flood monitoring and early warning systems, suitable for small- to medium-scale prototype deployment in residential areas prone to inundation with measurable water level variations.

Keywords: IoT, ESP32, TinyML, water level, flood warning

Pendahuluan

Banjir merupakan salah satu bencana hidrometeorologi yang sering terjadi di Indonesia dan memengaruhi berbagai aspek kehidupan masyarakat, mulai dari gangguan aktivitas sosial hingga kerusakan infrastruktur. Di banyak wilayah, pemantauan ketinggian air masih dilakukan secara manual atau semi-manual sehingga kurang responsif dan berpotensi menimbulkan keterlambatan dalam penyampaian informasi (Permana & Irawati, 2025). Pemantauan ketinggian air secara konvensional umumnya dilakukan menggunakan alat ukur manual seperti penggaris atau patok ukur. Metode ini memiliki keterbatasan karena bersifat tidak kontinu dan sangat bergantung pada kehadiran manusia di lokasi pengamatan (Rienandie & Pramudita, 2025).

Kemajuan teknologi Internet of Things (IoT) kemudian membuka peluang pengembangan sistem monitoring berbasis sensor ultrasonik dan mikrokontroler, di mana perangkat seperti ESP32 terbukti mampu melakukan pengukuran jarak air secara otomatis dan real-time (Hasibuan et al., 2025).

Meskipun perangkat IoT telah banyak digunakan, sebagian besar penelitian masih mengandalkan metode klasifikasi berbasis ambang batas sederhana yang diterapkan secara langsung pada hasil pembacaan sensor. Pendekatan ini cenderung kurang stabil ketika digunakan tanpa pengolahan lanjutan terhadap pola data, terutama pada kondisi pembacaan sensor yang berfluktuasi. Untuk mengatasi keterbatasan tersebut, *Tiny Machine Learning* (TinyML) hadir sebagai teknologi yang memungkinkan penerapan model pembelajaran mesin berukuran kecil secara langsung pada perangkat berdaya rendah seperti mikrokontroler, sehingga proses inferensi dapat dilakukan secara lokal tanpa memerlukan sumber daya komputasi yang besar (Banbury et al., 2021). Penelitian (Hamzah et al., 2024) serta (Samanta et al., 2025) menunjukkan bahwa model pembelajaran mesin berukuran kecil dapat dijalankan pada mikrokontroler dan meningkatkan interpretasi data sensor tanpa bergantung pada server eksternal. Di sisi lain, penggunaan notifikasi berbasis Telegram telah terbukti efektif sebagai sarana penyampaian peringatan dini secara cepat pada berbagai sistem IoT (Setiawan et al., 2024). Dalam penelitian ini, ambang batas ketinggian air tetap dimanfaatkan sebagai dasar penentuan kelas kondisi dan pelabelan dataset, namun proses klasifikasi dilakukan menggunakan model TinyML yang mempelajari pola data hasil pengukuran sensor sehingga sistem menjadi lebih adaptif terhadap variasi pembacaan.

Permasalahan yang menjadi fokus penelitian ini adalah ketiadaan sistem pemantauan ketinggian air yang mampu menggabungkan IoT, pengolahan data cerdas melalui TinyML, dan mekanisme notifikasi otomatis secara terpadu. Sistem ideal harus mampu mengklasifikasikan kondisi air secara adaptif ke dalam kategori aman, waspada, dan banjir, sehingga masyarakat dapat menerima informasi lebih dini dan akurat. Selain itu, penelitian ini juga mempertimbangkan kebutuhan indikator lokal seperti LED dan tampilan OLED untuk memudahkan pengguna dalam memantau kondisi secara langsung.

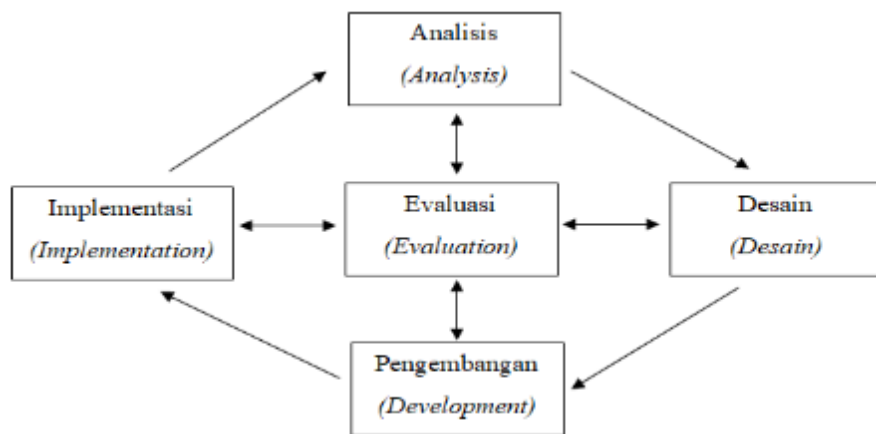
Penelitian ini bertujuan merancang dan mengimplementasikan prototipe pemantauan ketinggian air berbasis ESP32 dan sensor HC-SR04 yang dipadukan dengan model TinyML untuk klasifikasi kondisi air secara real-time. Sistem dilengkapi indikator LED serta tampilan OLED sebagai penunjang visual, dan didukung fitur pengiriman notifikasi otomatis melalui Telegram ketika kondisi air memasuki kategori peringatan. Manfaat praktis dari penelitian ini adalah menyediakan solusi monitoring yang murah, sederhana, dan dapat direplikasi, sementara manfaat akademisnya mencakup kontribusi terhadap pengembangan sistem mitigasi bencana berbasis IoT dan kecerdasan tertanam.

Metodologi Penelitian

Penelitian ini dilaksanakan di lingkungan rumah dengan memanfaatkan ruang kerja sebagai laboratorium mini dan aquarium sebagai media simulasi perubahan ketinggian air. Kondisi tersebut memungkinkan pengaturan eksperimen secara terkontrol, termasuk pergerakan permukaan air, posisi sensor, serta stabilitas perangkat. Penelitian berlangsung selama beberapa minggu, mencakup proses perancangan, pelatihan model TinyML, integrasi perangkat keras, hingga pengujian fungsional sistem.

Jenis penelitian yang digunakan adalah metode Research and Development (R&D) dengan model pengembangan ADDIE (Analysis, Design, Development, Implementation, Evaluation). Metode R&D dipilih karena berorientasi pada proses perancangan, pengembangan, dan evaluasi suatu produk teknologi secara sistematis hingga dihasilkan prototipe yang layak digunakan dan diuji pada skala terbatas (Zakariah et al., 2020). Model ADDIE dipakai karena memiliki tahapan yang terstruktur dan fleksibel, sehingga sesuai untuk pengembangan sistem berbasis perangkat keras dan perangkat lunak secara terintegrasi.

Pendekatan yang digunakan memadukan analisis konseptual, eksperimen perangkat keras, serta evaluasi performa perangkat dan model. Sumber data terdiri dari data primer berupa hasil pembacaan jarak oleh sensor HC-SR04, serta data sekunder berupa literatur teknis dan referensi ilmiah terkait IoT, sensor ultrasonik, ESP32, TinyML, dan sistem deteksi ketinggian air. Alur metode penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur Metode Penelitian

Analysis (Analysis)

Analisis kebutuhan dilakukan untuk mengidentifikasi komponen yang diperlukan dalam membangun sistem pemantauan ketinggian air berbasis ESP32, sensor ultrasonik, TinyML, dan notifikasi Telegram. Analisis ini mencakup kebutuhan perangkat keras dan perangkat lunak, serta penentuan kriteria kondisi air yang digunakan sebagai dasar pelabelan data dan pembentukan model klasifikasi. Seluruh analisis dilakukan pada lingkungan uji terkontrol menggunakan media simulasi berupa aquarium.

Kebutuhan Perangkat Keras (Hardware)

Tabel 1 menunjukkan komponen perangkat keras yang digunakan dalam penelitian ini beserta fungsinya.

Tabel 1. Kebutuhan Komponen Perangkat Keras

No	Komponen	Fungsi
1	Mikrokontroler ESP32	Unit pemroses utama yang membaca data sensor, menjalankan model TinyML di perangkat (on-device inference), mengontrol LED/LCD, serta mengirimkan notifikasi melalui koneksi Wi-Fi.
2	Sensor ultrasonik HC-SR04	Mengukur jarak permukaan air secara non-kontak; nilai jarak ini diolah menjadi level ketinggian air untuk klasifikasi TinyML.
3	PCB lubang dan kabel	Sebagai media perakitan rangkaian elektronik agar sambungan stabil, rapi, dan meminimalkan noise atau koneksi longgar selama pengujian.
4	LED indikator	Memberikan sinyal visual cepat mengenai kondisi level air tanpa harus melihat LCD atau Telegram. LED hijau = aman, kuning = waspada, merah = banjir.
5	LCD OLED 0.96 I2C	Menampilkan data lokal seperti jarak terukur, level air hasil klasifikasi TinyML, status Wi-Fi, atau pesan proses saat sistem berjalan. Membantu pemantauan langsung tanpa membuka smartphone.
6	Box Akrilik	Melindungi seluruh rangkaian dari air, debu, dan sentuhan yang tidak diinginkan. Selain itu memperbaiki estetika alat dan menjaga komponen tetap kokoh selama pengujian.
7	Catu daya	Menyediakan suplai listrik stabil untuk ESP32, sensor, dan OLED. Menghindari reset otomatis akibat drop tegangan.
8	Wadah simulasi (Aquarium)	Digunakan sebagai media uji untuk mensimulasikan kondisi kenaikan air secara bertahap. Nilai air asli menjadi dasar label dan validasi TinyML.
9	Penggaris / alat ukur manual	Menjadi acuan pembandingan untuk validasi data sensor ultrasonik dan evaluasi akurasi model TinyML.
10	Laptop/komputer	Digunakan untuk pemrograman ESP32, training TinyML, pembuatan dataset, kalibrasi sensor, dan analisis performa model.
11	Smartphone	Menerima notifikasi peringatan melalui Telegram Bot dan memantau kondisi level air secara real-time dari jarak jauh.

Kebutuhan Perangkat Lunak (Software)

Tabel 2 menunjukkan perangkat lunak dan pustaka yang digunakan untuk mendukung pengembangan, pelatihan TinyML, dan implementasi firmware pada ESP32.

Tabel 2. Kebutuhan Perangkat Lunak

No	Komponen	Fungsi
1	Arduino IDE	Digunakan untuk menulis, mengompilasi, dan mengunggah program ke mikrokontroler ESP32.
2	ESP32 Board Package (espressif)	Menyediakan toolchain, compiler, bootloader, dan konfigurasi pin yang diperlukan agar Arduino IDE dapat memprogram ESP32 dengan benar.
3	Library EloquentTinyML / TensorFlow Lite Micro (tflm_esp32)	Menangani pemuatan model TinyML, inference, alokasi tensor arena, dan operasi fully-connected/softmax sesuai model klasifikasi level air.
4	Library WiFi.h	Digunakan ESP32 untuk koneksi ke jaringan Wi-Fi agar dapat mengirim data dan notifikasi.
5	UniversalTelegramBot + ArduinoJson	Mengirimkan notifikasi status level air (aman/waspada/banjir) secara real-time ke Telegram Bot.
6	Library Ultrasonic	Mengelola pemicuan TRIG, pembacaan ECHO, debouncing, dan perhitungan jarak berbasis waktu tempuh gelombang ultrasonik.
7	Library Adafruit SSD1306 & Adafruit GFX	Mengelola tampilan OLED 0.96 inci I2C untuk menampilkan jarak terukur, status Wi-Fi, level air hasil inference.
8	Model TinyML (file header)	Model klasifikasi tiga kelas hasil pelatihan yang diintegrasikan langsung ke firmware ESP32.
9	Serial Monitor & Serial Plotter	Digunakan untuk debugging: monitoring heap memory, log koneksi Wi-Fi, nilai sensor mentah, dan output inference TinyML.
10	Dataset Training (CSV / JSON)	Berisi data jarak air dan label kelas (aman/waspada/banjir) yang dipakai untuk melatih model TinyML sebelum dikonversi ke TFLite.
11	Python + TensorFlow (untuk training)	Digunakan untuk preprocessing data, melatih model ANN sederhana, melakukan konversi TFLite Micro, lalu mengekspor model ke header C untuk digunakan ESP32.

Penentuan Kriteria dan Threshold Ketinggian Air

Penentuan kriteria dan threshold ketinggian air dilakukan sebagai bagian dari tahap analisis untuk mendefinisikan kondisi sistem dan mendukung proses pelabelan dataset. Threshold ditetapkan berdasarkan tinggi maksimum media simulasi (aquarium) dan disesuaikan dengan skenario pengujian yang merepresentasikan kondisi aman hingga banjir pada skala laboratorium. Kriteria ini digunakan untuk mengelompokkan kondisi air ke dalam tiga kategori, yaitu aman, waspada, dan banjir.

Tabel 1. Kriteria dan Threshold Ketinggian Air

No	Kategori	Rentang Ketinggian Air	Deskripsi Kondisi
1	Aman	0 – < 10 cm	Kondisi air rendah
2	Waspada	10 – < 20 cm	Ketinggian air meningkat
3	Banjir	\geq 20 cm	Potensi banjir

Nilai threshold ini digunakan sebagai acuan awal dalam proses pelabelan dataset, di mana setiap data ketinggian air hasil pembacaan sensor ultrasonik HC-SR04 diberi label sesuai dengan kategori yang telah ditentukan. Dataset berlabel tersebut selanjutnya digunakan untuk melatih model TinyML agar mampu mempelajari pola hubungan antara ketinggian air dan kelas kondisi air.

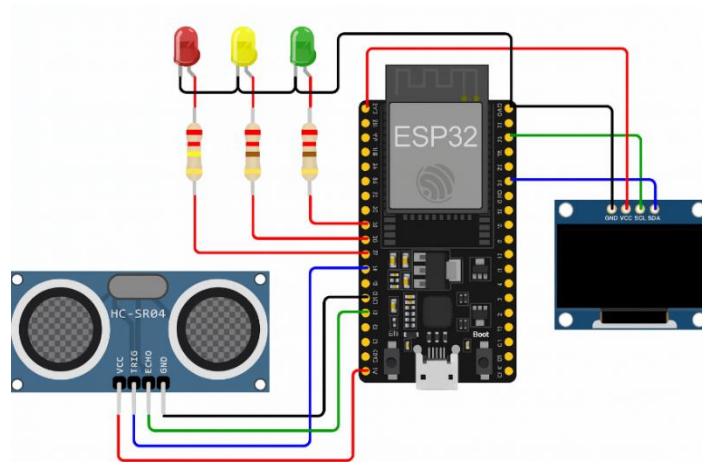
Pada tahap implementasi, threshold tidak digunakan sebagai logika keputusan langsung dalam sistem. Keputusan klasifikasi sepenuhnya dihasilkan oleh model TinyML yang melakukan inferensi secara lokal pada perangkat ESP32 (on-device inference). Dengan pendekatan ini, sistem tidak hanya bergantung pada ambang batas statis, tetapi memanfaatkan kemampuan pembelajaran mesin untuk melakukan klasifikasi kondisi air secara adaptif terhadap variasi data sensor.

Design (Perancangan)

Tahap perancangan bertujuan menghasilkan gambaran struktural perangkat keras dan alur kerja perangkat lunak sebelum prototipe direalisasikan. Perancangan dilakukan berdasarkan hasil analisis kebutuhan, karakteristik komponen, serta fungsi utama sistem dalam mendeteksi dan mengklasifikasikan kondisi air.

Perancangan Perangkat Keras

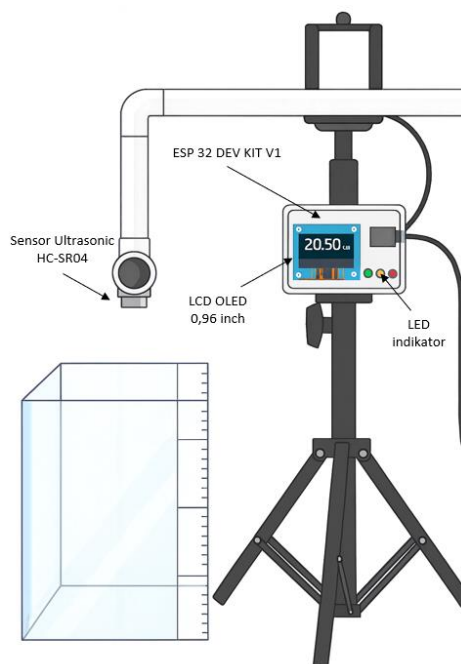
Perangkat keras dirancang dengan mengintegrasikan beberapa komponen utama, yaitu ESP32 DevKit V1 sebagai pengendali sistem, sensor ultrasonik HC-SR04 sebagai pendeteksi jarak permukaan air, layar OLED 0.96 inci sebagai media tampilan, dan tiga LED indikator sebagai penanda visual kondisi air. Untuk mendukung pemantauan secara langsung, indikator lokal berupa *Light Emitting Diode* (LED) dan layar OLED digunakan guna menampilkan kondisi sistem di lokasi pemasangan perangkat (Kurniawan & Budjianto, 2025). Struktur hubungan antar komponen digambarkan dalam rancangan koneksi perangkat keras pada Gambar 2.



Gambar 2. Rancangan Koneksi Perangkat Keras

Sensor HC-SR04 diposisikan sebagai komponen input utama yang mengirimkan sinyal jarak melalui pin TRIG dan ECHO. OLED terhubung menggunakan antarmuka I2C (SDA dan SCL), sementara LED hijau, kuning, dan merah ditempatkan pada pin output digital untuk memberikan indikasi status “aman–waspada–banjir”. Susunan perangkat keras dirancang agar memastikan stabilitas suplai daya, keamanan koneksi pin, serta kemudahan observasi hasil visual.

Rancangan fisik alat ditunjukkan pada Gambar 3, yang menampilkan tata letak komponen dalam wadah pelindung. Sensor ditempatkan menghadap permukaan air untuk memaksimalkan akurasi pembacaan, sedangkan modul ESP32 dan OLED disusun pada area yang terlindungi namun tetap mudah diakses. LED indikator ditempatkan pada sisi depan perangkat agar perubahan status dapat diamati secara cepat. Keseluruhan desain bertujuan menghasilkan struktur perangkat keras yang stabil, ringkas, dan optimal untuk proses pengujian dalam lingkungan simulasi.



Gambar 3. Rancangan Fisik Alat Pendeteksi Banjir

Perancangan Perangkat Lunak

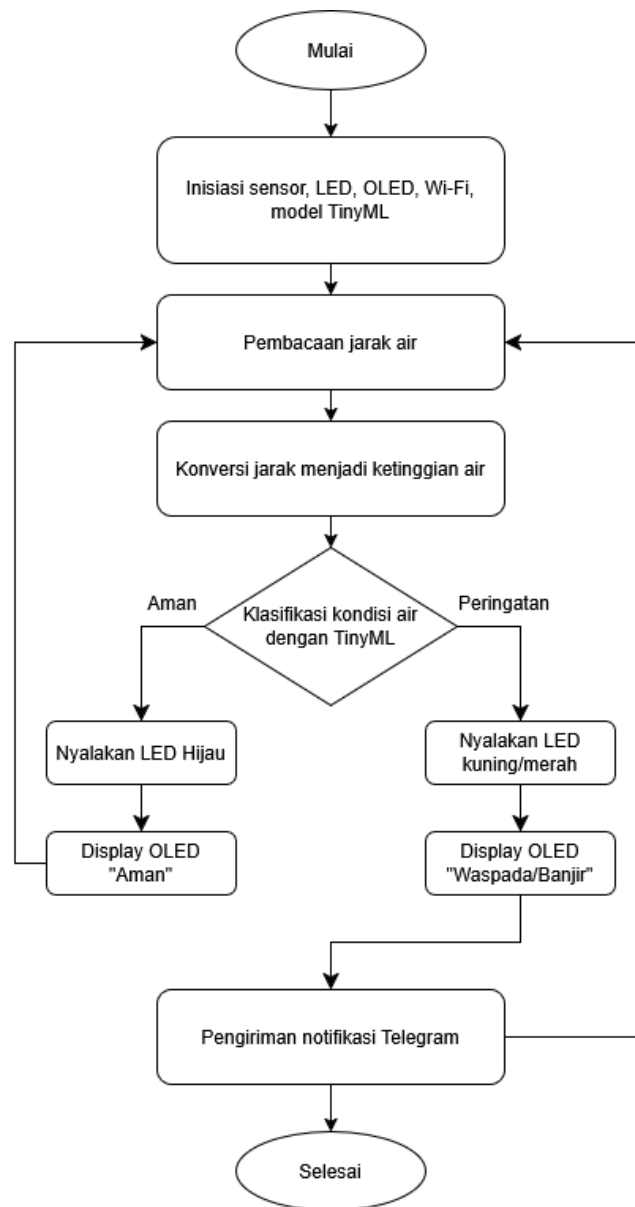
Perancangan perangkat lunak dilakukan untuk menetapkan struktur logika, alur proses, dan komponen fungsional yang diperlukan agar sistem deteksi ketinggian air dapat berjalan sesuai dengan tujuan penelitian. Tahap ini menghasilkan rancangan modular yang mencakup proses akuisisi data, pemrosesan, klasifikasi TinyML, visualisasi status, dan mekanisme notifikasi. Rancangan ini disusun sebelum masuk ke tahap implementasi sehingga setiap komponen memiliki fungsi yang jelas dan saling terintegrasi.

Secara umum, perancangan perangkat lunak dibagi menjadi empat modul utama:

1. Modul pembacaan sensor HC-SR04
2. Modul konversi dan pemrosesan data
3. Modul inferensi TinyML
4. Modul keluaran (LED, OLED, Telegram).

Setiap modul dirancang agar dapat berjalan secara berurutan dalam loop utama ESP32. Modul inferensi TinyML berperan sebagai komponen utama dalam menentukan kategori kondisi air. Klasifikasi aman, waspada, dan banjir tidak ditentukan oleh ambang batas statis pada tahap operasional, melainkan dihasilkan melalui proses inferensi model pembelajaran mesin yang telah dilatih sebelumnya menggunakan dataset berlabel.

Modul keluaran dirancang untuk menyajikan hasil klasifikasi kondisi air baik secara lokal maupun jarak jauh. Indikator lokal berupa *Light Emitting Diode* (LED) dan layar OLED digunakan untuk menampilkan status sistem secara langsung di lokasi pemasangan perangkat. Keberadaan indikator lokal ini berfungsi sebagai sarana informasi cadangan yang tetap dapat diakses ketika koneksi jaringan mengalami gangguan (Oktarian et al., 2023). Selain itu, sistem ini juga dilengkapi dengan mekanisme notifikasi otomatis melalui Telegram sebagai media penyampaian peringatan dini kepada pengguna ketika kondisi air memasuki kategori peringatan, sebagaimana umum diterapkan pada sistem monitoring berbasis IoT (Lengkong et al., 2025).



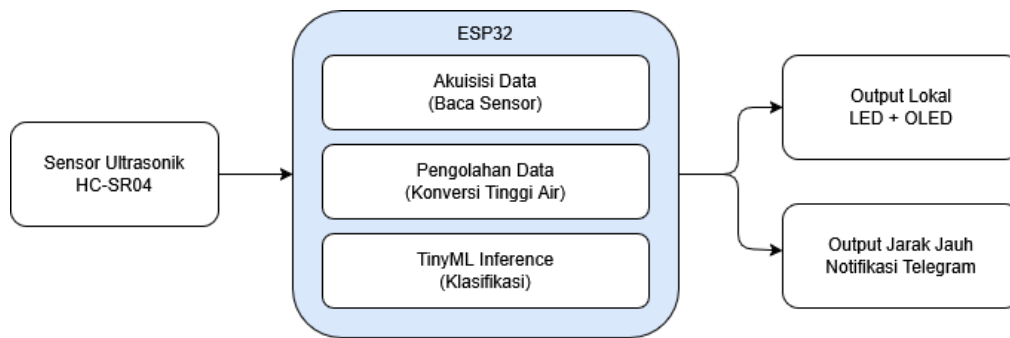
Gambar 4. Flowchart Perancangan Perangkat Lunak

Flowchart menggambarkan bahwa perangkat lunak dimulai dari inisialisasi komponen, dilanjutkan dengan pembacaan jarak dari sensor ultrasonik, konversi menjadi tinggi air, pengiriman data ke model TinyML untuk inferensi klasifikasi, dan akhirnya penyajian keluaran melalui LED, OLED, serta notifikasi Telegram.

Untuk mendukung konsistensi informasi, ditetapkan tiga struktur keluaran:

1. Status numeric (angka tinggi air dalam cm),
2. Status simbolik (Aman, Waspada, Banjir),
3. Status visual (warna LED dan teks OLED).

Seluruh keluaran tersebut merepresentasikan hasil klasifikasi yang dihasilkan oleh model TinyML secara on-device dan tidak berasal dari keputusan berbasis ambang batas langsung.



Gambar 5. Diagram Alur Modul Sistem

Setiap modul dirancang memanfaatkan library yang ringan dan kompatibel dengan kapasitas ESP32, termasuk library ultrasonik untuk pembacaan jarak, library TinyML/EloquentTinyML untuk inferensi model, serta library I2C untuk OLED. Notifikasi Telegram menggunakan API bot yang bekerja dengan protokol HTTPS, sehingga modul keluaran dirancang dengan buffer data kecil agar tidak membebani memori.

Dengan rancangan ini, perangkat lunak dapat diimplementasikan secara bertahap dan diuji modul per modul sebelum diuji sebagai sistem terpadu pada tahap implementasi.

Development (Pengembangan)

Tahap pengembangan berfokus pada proses realisasi rancangan perangkat keras dan perangkat lunak menjadi sebuah prototipe yang berfungsi sesuai tujuan penelitian. Seluruh aktivitas mengikuti struktur rancangan yang telah ditetapkan pada tahap sebelumnya, sehingga setiap komponen yang dikembangkan memiliki peran dan fungsi yang jelas dalam sistem.

Pengembangan Perangkat Keras

Pengembangan perangkat keras dilakukan dengan merakit komponen utama sesuai blok diagram sistem. Modul ESP32 DevKit V1 dihubungkan dengan sensor HC-SR04, OLED 0.96 inci, serta tiga LED indikator menggunakan konfigurasi pin yang telah dirancang pada tahap design. Setelah rangkaian selesai, dilakukan verifikasi sambungan untuk memastikan suplai daya dan jalur sinyal berfungsi dengan stabil. Sensor ultrasonik diposisikan tegak lurus terhadap permukaan air untuk memperoleh pembacaan jarak yang konsisten dan meminimalkan kesalahan pengukuran akibat sudut pantul gelombang ultrasonik, sebagaimana direkomendasikan pada sistem pemantauan berbasis sensor ultrasonik (Fauziyah et al., 2024). Seluruh rangkaian kemudian dipasang pada wadah uji berbasis aquarium dengan posisi tetap agar kondisi lingkungan eksperimen dapat direplikasi.

Pengembangan Perangkat Lunak

Pengembangan perangkat lunak mencakup perwujudan seluruh modul yang telah dirancang, mulai dari pembacaan sensor hingga klasifikasi kondisi air. Modul pembacaan sensor dikembangkan menggunakan library ultrasonik untuk memperoleh jarak permukaan air, kemudian jarak tersebut dikonversi menjadi tinggi air sesuai konfigurasi ketinggian pemasangan sensor. Data sampel dari berbagai variasi ketinggian air dikumpulkan untuk melatih model TinyML menggunakan arsitektur Dense–ReLU–Dense–Softmax. Model yang telah dilatih dikonversi menjadi format TensorFlow Lite Micro dan diintegrasikan dalam firmware ESP32 sebagai berkas header. Data ketinggian air yang diperoleh sebelumnya diberi label berdasarkan

kriteria ambang batas yang telah ditetapkan pada tahap analisis. Pelabelan ini memungkinkan model TinyML dilatih menggunakan pendekatan pembelajaran terawasi (*supervised learning*), sehingga model mampu mempelajari hubungan antara nilai ketinggian air dan kelas kondisi yang sesuai.

Setelah modul inferensi selesai, dikembangkan modul keluaran berupa pengendalian LED, tampilan OLED melalui antarmuka I2C, serta fungsi notifikasi Telegram menggunakan bot API. Setiap modul diuji secara bertahap untuk memastikan kompatibilitas dan efisiensi memori, mengingat keterbatasan kapasitas ESP32. Seluruh modul kemudian digabungkan dalam loop utama sehingga sistem dapat membaca, memproses, mengklasifikasikan, dan mengirim keluaran secara real-time.

Tahap pengembangan ini menghasilkan prototipe lengkap yang siap diuji pada tahap implementasi dan evaluasi.

Implementation (Implementasi)

Tahap implementasi merupakan proses penerapan hasil perancangan ke dalam bentuk prototipe yang berfungsi secara menyeluruh. Implementasi dilakukan dalam dua bagian utama, yaitu implementasi perangkat keras dan implementasi perangkat lunak. Tahap ini bertujuan untuk memastikan bahwa seluruh komponen sistem, mulai dari pembacaan sensor, proses klasifikasi berbasis TinyML, hingga penyampaian notifikasi melalui Telegram, dapat bekerja sesuai dengan fungsi dan alur sistem yang telah dirancang (Mahardhika et al., 2025).

Implementasi Perangkat Keras

Implementasi perangkat keras dilakukan dengan merakit seluruh komponen utama sistem, yaitu ESP32 DevKit V1, sensor ultrasonik HC-SR04, OLED 0.96 inci, dan tiga LED indikator (hijau, kuning, merah). Proses implementasi meliputi pemasangan komponen, pengujian konektivitas, dan verifikasi tegangan operasional.

Tahapan implementasi perangkat keras adalah sebagai berikut:

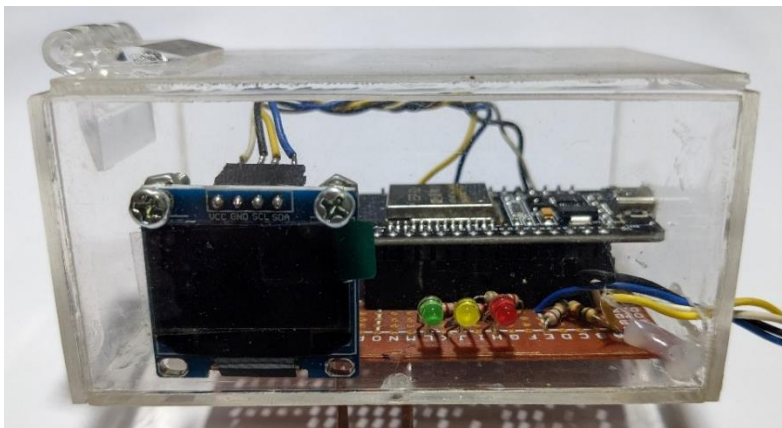
1. Penyambungan Sensor HC-SR04 ke ESP32
 - a. Pin TRIG dihubungkan ke pin digital output ESP32.
 - b. Pin ECHO dihubungkan ke pin input ESP32.
 - c. Pin VCC dan GND disambungkan ke sumber daya ESP32.
Posisi sensor diarahkan tegak lurus ke permukaan air dalam aquarium, dengan jarak tetap terhadap titik referensi ketinggian air.
2. Pemasangan LED Indikator
 - a. LED hijau dipasang pada pin output untuk kondisi aman.
 - b. LED kuning dipasang untuk kondisi waspada.
 - c. LED merah dipasang untuk kondisi banjir.
Seluruh LED menggunakan resistor seri guna menjaga arus tetap aman.
3. Integrasi OLED 0.96 I2C
 - a. Pin SDA dan SCL dihubungkan ke jalur I2C ESP32.

- b. OLED digunakan sebagai tampilan lokal untuk menunjukkan jarak, ketinggian air, dan status klasifikasi TinyML.

4. Penempatan Fisik Komponen

- a. Sensor diletakkan pada bagian atas aquarium secara tetap agar pembacaan ultrasonik konsisten.
- b. ESP32 dan OLED ditempatkan di area yang terlindungi dan mudah diamati.
- c. Seluruh komponen dirakit secara rapi menggunakanudukan atau rangka sederhana agar stabil saat pengujian.

Hasil implementasi perangkat keras ditampilkan pada Gambar 6, sementara foto hasil perakitan ditampilkan pada Gambar 7.



Gambar 6. Hasil Implementasi Perangkat Keras



Gambar 7. Hasil Perakitan Alat Pendeteksi Banjir

Implementasi Perangkat Lunak

Implementasi perangkat lunak dilakukan dengan menggabungkan seluruh modul program yang telah dirancang pada tahap desain. Proses ini meliputi instalasi library, pemrograman logika sistem, integrasi model TinyML, serta pengujian fungsi notifikasi Telegram. Implementasi software berjalan sepenuhnya pada mikrokontroler ESP32. Implementasi perangkat lunak ini mengacu pada kriteria dan ambang batas ketinggian air yang telah ditetapkan pada tahap analisis, yang sebelumnya digunakan sebagai dasar pelabelan dataset dan pelatihan model TinyML.

Tahapan implementasi perangkat lunak terdiri dari:

1. Inisialisasi Komponen dan Library
 - a. Library *Ultrasonic*, *Adafruit SSD1306*, *EloquentTinyML/TFLM*, *WiFi*, dan *UniversalTelegramBot* dipasang.
 - b. ESP32 dikonfigurasi untuk koneksi Wi-Fi menggunakan SSID dan password yang telah ditentukan.
2. Pemrograman Pembacaan Sensor
 - a. ESP32 mengirimkan sinyal TRIG dan membaca durasi ECHO dari HC-SR04.
 - b. Jarak dihitung menggunakan rumus:
$$d = \frac{v.t}{2}$$
 - c. Ketinggian air dihitung dari selisih antara jarak sensor dan jarak referensi.
3. Integrasi Model TinyML
 - a. Model klasifikasi tiga kelas (aman–waspada–banjir) dimuat dalam bentuk header .h hasil konversi TensorFlow Lite Micro.
 - b. Input berupa tinggi air dikirimkan ke model untuk proses inferensi langsung pada ESP32, di mana keputusan klasifikasi ditentukan oleh hasil pembelajaran model, bukan oleh penerapan ambang batas statis secara langsung.
4. Pengendalian Output Lokal
 - a. Hasil klasifikasi digunakan untuk menyalakan LED sesuai kondisi.
 - b. OLED menampilkan teks status lengkap (aman/waspada/banjir) serta nilai tinggi air.
5. Implementasi Notifikasi Telegram
 - a. Sistem memicu pengiriman pesan otomatis ketika kondisi waspada atau banjir.
 - b. Pesan berisi kategori kondisi air, tinggi air, dan timestamp.
6. Loop Sistem dan Monitoring Berkelanjutan
 - a. Seluruh proses berjalan dalam loop utama yang mengulang pembacaan sensor, klasifikasi, dan keluaran dengan interval waktu tertentu yang telah disesuaikan dengan kebutuhan monitoring dan stabilitas pembacaan sensor.

- b. Sistem diuji dengan variasi ketinggian air untuk memastikan performa stabil.

Evaluation (Evaluasi)

Tahap evaluasi dilakukan untuk menilai kinerja keseluruhan prototipe setelah implementasi perangkat keras dan perangkat lunak terselesaikan. Evaluasi diarahkan pada tiga aspek utama, yaitu akurasi pembacaan sensor, keandalan klasifikasi TinyML, dan konsistensi keluaran indikator (OLED, LED, dan notifikasi Telegram). Seluruh pengujian dilakukan di lingkungan simulasi menggunakan aquarium dengan variasi ketinggian air yang direplikasi secara sistematis.

Evaluasi perangkat keras difokuskan pada kestabilan pembacaan HC-SR04, respons LED terhadap perubahan kategori kondisi air, serta keterbacaan tampilan OLED. Setiap variasi ketinggian air dicatat dan dibandingkan dengan hasil pengukuran menggunakan alat ukur manual sebagai data referensi untuk memperoleh selisih pembacaan dan menilai tingkat akurasi sensor.

Evaluasi perangkat lunak diarahkan pada performa model TinyML dalam mengklasifikasikan kondisi “aman–waspada–banjir”. Proses inferensi dievaluasi menggunakan data hasil pembacaan sensor secara langsung pada perangkat (on-device inference), dengan hasil klasifikasi dibandingkan terhadap kondisi ketinggian air aktual yang telah ditentukan berdasarkan skenario pengujian dan pengukuran referensi. Selain itu, keandalan sistem notifikasi Telegram diuji melalui uji pemicu (trigger test) untuk memastikan pesan peringatan terkirim secara tepat ketika kondisi air memasuki kategori waspada atau banjir (Wikantama & Puspitasari, 2023). Aspek ketepatan isi pesan dan konsistensi pengiriman turut menjadi bagian dari evaluasi, mengacu pada praktik implementasi notifikasi berbasis Telegram pada sistem IoT (Ovid et al., 2025).

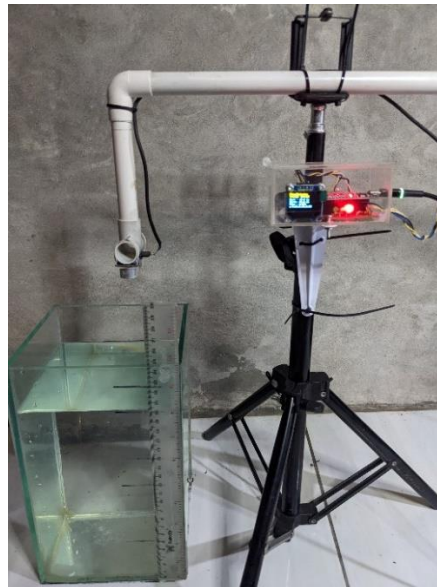
Hasil evaluasi digunakan untuk menilai kesesuaian sistem dengan tujuan penelitian, yaitu menghasilkan prototipe pendeteksi ketinggian air yang akurat, responsif, dan dapat memberikan peringatan dini. Temuan dari tahap evaluasi juga menjadi dasar rekomendasi pengembangan lanjutan, seperti peningkatan kestabilan pembacaan sensor, optimasi ukuran dan struktur model TinyML, serta penyempurnaan proses pelabelan data agar sistem semakin adaptif terhadap variasi kondisi lingkungan.

Hasil dan Pembahasan

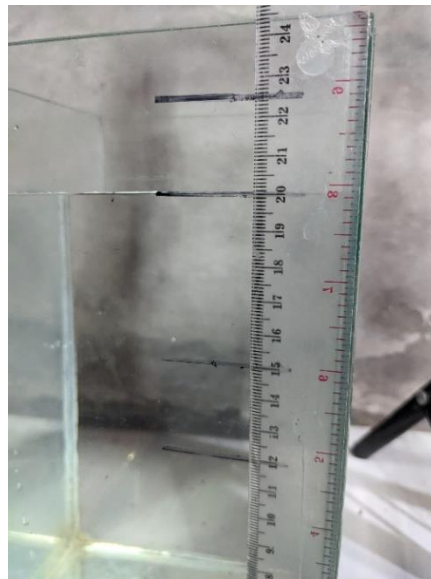
Bagian ini menyajikan hasil pengujian prototipe sistem deteksi ketinggian air berbasis ESP32 yang telah diimplementasikan. Data ditampilkan dalam bentuk tabel dan gambar untuk memperlihatkan performa sensor, akurasi klasifikasi TinyML, serta fungsi indikator LED, OLED, dan notifikasi Telegram. Pembahasan difokuskan pada interpretasi hasil untuk menjawab rumusan masalah dan tujuan penelitian.

Hasil Uji Fungsional Sistem (LED + OLED + Telegram + TinyML)

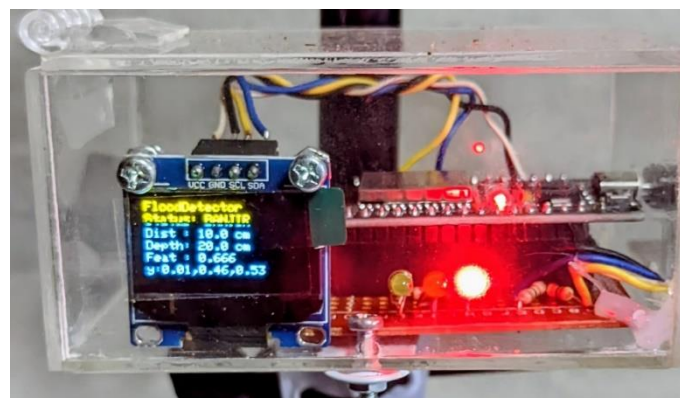
Uji fungsional dilakukan untuk memverifikasi bahwa sistem mampu menjalankan fungsi utama secara terpadu, yaitu: (1) mengukur jarak permukaan air, (2) menghitung ketinggian air (*depth*), (3) mengklasifikasikan status menggunakan TinyML, serta (4) menampilkan status melalui LED dan OLED, dan (5) mengirim notifikasi Telegram sesuai aturan peringatan.



Gambar 8. Sampel Uji Fungsional Alat



Gambar 9. Sampel Ketinggian Air dengan Penggaris



Gambar 10. Status OLED dan LED Banjir



Gambar 11. Notifikasi Telegram

Tabel 4. Uji Fungsional Alat (output LED + OLED + Telegram + TinyML)

No	Kondisi Uji (Setpoint)	Dist Target (cm)	Depth Ekspektasi (cm)	Status Ekspektasi	LED Ekspektasi	OLED Ekspektasi (Status)	Telegram Terkirim?	Hasil Model (vote)	Sesuai?
1	Aquarium kosong (air 0)	30,0	0,0	AMAN	Hijau	“AMAN”	Ya (awal aktif)	AMAN	YA
2	Air rendah	26,0	4,0	AMAN	Hijau	“AMAN”	Tidak	AMAN	YA
3	Air mendekati batas A/W	21,0	9,0	AMAN	Hijau	“AMAN”	Tidak	AMAN	YA
4	Tepat batas A/W	20,0	10,0	WASPADA	Kuning	“WASPADA”	Ya	WASPADA	YA
5	Waspada stabil	15,0	15,0	WASPADA	Kuning	“WASPADA”	Ya	WASPADA	YA
6	Mendekati batas W/B	11,0	19,0	WASPADA	Kuning	“WASPADA”	Ya	WASPADA	YA
7	Tepat batas W/B	10,0	20,0	BANJIR	Merah	“BANJIR”	Ya	BANJIR	YA
8	Banjir stabil	8,0	22,0	BANJIR	Merah	“BANJIR”	Ya	BANJIR	YA
9	Kembali turun (banjir→waspada)	12,0	18,0	WASPADA	Kuning	“WASPADA”	Ya	WASPADA	YA
10	Kembali turun (waspada→aman)	24,0	6,0	AMAN	Hijau	“AMAN”	Tidak	AMAN	YA

Berdasarkan Tabel 4, pengujian dilakukan pada 10 skenario setpoint yang mencakup kondisi AMAN, WASPADA, BANJIR, serta dua kondisi penurunan (BANJIR→WASPADA dan WASPADA→AMAN). Hasil menunjukkan:

1. Kesesuaian status antara Status Ekspektasi dan Hasil Model (vote) tercapai pada seluruh skenario (10/10 = 100%).
2. LED indikator konsisten terhadap status: hijau untuk AMAN, kuning untuk WASPADA, merah untuk BANJIR pada seluruh skenario.
3. OLED menampilkan teks status yang sesuai (AMAN/WASPADA/BANJIR) pada seluruh skenario.
4. Notifikasi Telegram mengikuti logika sistem: tidak mengirim pada kondisi AMAN (kecuali notifikasi awal aktif pada saat sistem menyala), dan mengirim pada kondisi

peringatan (WASPADA/BANJIR). Ini terlihat dari skenario 4–9 yang menghasilkan pengiriman notifikasi.

Hasil pengujian ini menegaskan bahwa integrasi sensor–konversi–inferensi TinyML–output lokal–Telegram berjalan stabil pada skenario uji yang mencakup kondisi batas kelas (A/W dan W/B).

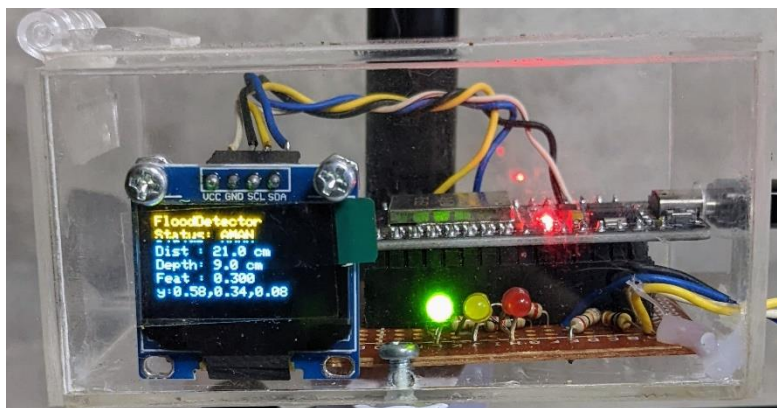
Hasil Uji Akurasi Pembacaan Sensor (Banding Penggaris)

Uji akurasi dilakukan untuk menilai ketepatan pembacaan sensor ultrasonik HC-SR04 terhadap acuan manual (penggaris). Perhitungan ketinggian air menggunakan formula pada sistem:

$$Depth = H_{ref} - Dist$$

Tabel 5. Uji akurasi pembacaan sensor (banding penggaris)

No	Tinggi Air Aktual (cm)	Dist Terukur (cm)	Depth Terhitung (cm) = 30 - dist	Error (cm) = Aktual - Terhitung	Status Ekspektasi	Status Tampil (OLED)	Sesuai
1	0,0	30,0	0,0	0,0	AMAN	AMAN	YA
2	5,0	25,0	5,0	0,0	AMAN	AMAN	YA
3	9,0	21,0	9,0	0,0	AMAN	AMAN	YA
4	10,0	19,8	10,2	0,2	WASPADA	WASPADA	YA
5	15,0	15,0	15,0	0,0	WASPADA	WASPADA	YA
6	19,0	11,0	19,0	0,0	WASPADA	WASPADA	YA
7	20,0	10,0	20,0	0,0	BANJIR	BANJIR	YA
8	22,5	7,5	22,5	0,0	BANJIR	BANJIR	YA
Rata-rata error				0,025			



Gambar 12. Status OLED dan LED Aman

Berdasarkan Tabel 5, digunakan $H_{ref} = 30$ cm sehingga $Depth = 30 - Dist$.

Hasil uji menunjukkan:

1. Dari 8 titik uji, 7 titik memiliki error 0,0 cm, dan 1 titik memiliki error 0,2 cm (pada aktual 10,0 cm; terhitung 10,2 cm).
2. Ringkasan metrik error (berdasarkan Tabel 5):
 - a. Total error absolut diperoleh dengan menjumlahkan seluruh selisih absolut antara tinggi air aktual dan hasil perhitungan sistem pada delapan titik uji, yaitu $0,0 + 0,0 + 0,0 + 0,2 + 0,0 + 0,0 + 0,0 + 0,0 = 0,2$ cm
 - b. MAE (Mean Absolute Error) dihitung dari total error absolut tersebut dibagi jumlah percobaan, sehingga
$$MAE = \frac{0,2}{8} = 0,025 \text{ cm}$$
 - c. Error maksimum yang diperoleh sebesar 0,2 cm
 - d. RMSE (Root Mean Square Error) dihitung sebagai
$$RMSE = \sqrt{0,2^2/8} \approx 0,071 \text{ cm}$$
 - e. Kolom “Status tampil (OLED)” menunjukkan status yang tampil selalu sesuai dengan status ekspektasi pada semua titik uji ($8/8 = 100\%$).

Hasil ini mengindikasikan bahwa pada rentang pengujian 0–22,5 cm, pembacaan sensor dan konversi ke depth cukup akurat untuk kebutuhan klasifikasi status. Nilai error maksimum sebesar 0,2 cm masih berada dalam toleransi sistem dan tidak menyebabkan kesalahan klasifikasi pada titik uji tersebut.

Hasil Uji Respons Notifikasi Telegram (Berbasis Transisi Status)

Uji Telegram dilakukan untuk memverifikasi bahwa sistem mengirim pesan ketika terjadi transisi menuju kondisi peringatan dan tidak mengirim pesan ketika kondisi kembali aman atau stabil aman.

Tabel 6. Uji respons notifikasi Telegram (berbasis perubahan status)

No	Transisi Status	Dist Awal → Akhir (cm)	Ekspektasi Telegram	Pesan Masuk? (Ya/Tidak)	Waktu Terima	Isi Pesan Sesuai (Status, Dist, Depth)?
1	AMAN → WASPADA	24 → 19.5	Harus Kirim	YA	23.20	YA
2	WASPADA → BANJIR	11 → 9.5	Harus Kirim	YA	23.05	YA
3	BANJIR → WASPADA	9.0 → 12	Harus Kirim	YA	23.04	YA
4	WASPADA → AMAN	19 → 24	Tidak Kirim	TIDAK	-	-
5	AMAN stabil	26 → 26	Tidak Kirim	TIDAK	-	-

Berdasarkan Tabel 6:

1. Sistem mengirim notifikasi pada transisi peringatan yang diharapkan:
 - a. AMAN → WASPADA: pesan masuk YA
 - b. WASPADA → BANJIR: pesan masuk YA
 - c. BANJIR → WASPADA: pesan masuk YA
2. Sistem tidak mengirim pada kondisi yang diharapkan tidak mengirim:
 - a. WASPADA → AMAN: pesan masuk TIDAK
 - b. AMAN stabil: pesan masuk TIDAK
3. Kesesuaian perilaku notifikasi terhadap ekspektasi adalah 5/5 (100%), dan isi pesan dinyatakan sesuai (memuat status dan parameter yang relevan).

Pembahasan

Hasil pengujian menunjukkan bahwa seluruh fungsi utama prototipe berjalan sesuai rancangan. Pada uji fungsional (Tabel 4), sistem berhasil menampilkan status yang konsisten antara Status Ekspektasi dan Hasil Model (*vote*) pada seluruh skenario uji (10 dari 10 skenario, 100%). Output lokal juga berjalan sesuai rancangan, di mana LED hijau menandakan kondisi AMAN, LED kuning menandakan WASPADA, dan LED merah menandakan BANJIR, serta OLED menampilkan teks status yang sesuai pada setiap kondisi. Notifikasi Telegram terkirim pada kondisi peringatan (WASPADA dan BANJIR), sedangkan pada kondisi AMAN tidak dilakukan pengiriman, kecuali pada saat awal sistem aktif.

Pada uji akurasi pembacaan sensor (Tabel 5), hasil menunjukkan error absolut pembacaan ketinggian air sangat kecil, dengan error maksimum sebesar 0,2 cm pada salah satu titik uji dan error 0,0 cm pada titik lainnya. Secara rata-rata, MAE berdasarkan 8 titik uji adalah 0,025 cm, sehingga pembacaan sensor dinilai memadai sebagai sumber data untuk estimasi ketinggian air dan penentuan kelas kondisi. Hasil ini sejalan dengan beberapa penelitian sebelumnya yang melaporkan bahwa sensor ultrasonik berbasis mikrokontroler mampu memberikan akurasi yang

cukup baik untuk pemantauan ketinggian air pada skala lingkungan terkontrol, khususnya ketika digunakan sebagai dasar sistem peringatan dini berbasis IoT.

Uji respons notifikasi Telegram (Tabel 6) membuktikan bahwa mekanisme peringatan berbasis perubahan status berjalan sesuai desain. Sistem mengirim pesan pada transisi AMAN→WASPADA, WASPADA→BANJIR, dan BANJIR→WASPADA, serta tidak mengirim pesan pada kondisi WASPADA→AMAN dan AMAN stabil. Pola ini konsisten dengan temuan studi-studi IoT sebelumnya yang menekankan pentingnya mekanisme notifikasi berbasis peristiwa (*event-based notification*) untuk menghindari pengiriman pesan berlebih dan meningkatkan efektivitas sistem peringatan dini. Namun, pengujian ini masih terbatas pada verifikasi keberhasilan pengiriman dan kesesuaian isi pesan; pengukuran delay/latensi belum dihitung karena data yang tersedia berupa waktu terima saja.

Secara keseluruhan, hasil pengujian mendukung bahwa integrasi ESP32, HC-SR04, TinyML, LED, OLED, dan Telegram mampu membentuk sistem monitoring ketinggian air real-time yang bekerja konsisten pada lingkungan simulasi aquarium. Dari sisi implikasi praktis, prototipe ini paling sesuai diterapkan sebagai sistem pemantauan dan peringatan dini banjir pada skala kecil hingga menengah, seperti saluran drainase lingkungan, kolam retensi, atau area permukiman rawan genangan dengan perubahan ketinggian air yang relatif terukur. Sistem ini relevan bagi masyarakat, pengelola lingkungan, atau pengelola fasilitas lokal yang membutuhkan solusi pemantauan ketinggian air yang sederhana, berbiaya relatif rendah, dan mampu melakukan klasifikasi kondisi air secara lokal pada perangkat.

Batasan Penelitian

Penelitian ini memiliki beberapa batasan yang perlu diperhatikan dalam menginterpretasikan hasil yang diperoleh. Pengujian sistem dilakukan pada lingkungan simulasi menggunakan aquarium, sehingga kondisi pengujian belum sepenuhnya merepresentasikan dinamika lingkungan nyata seperti hujan, arus air, turbulensi, dan gangguan eksternal lainnya yang umum terjadi pada sungai atau saluran drainase. Selain itu, sistem dirancang menggunakan satu sensor ultrasonik HC-SR04, sehingga aspek redundansi dan toleransi kesalahan pengukuran antar sensor belum dievaluasi. Evaluasi performa notifikasi Telegram pada penelitian ini juga masih terbatas pada keberhasilan pengiriman pesan dan kesesuaian isi notifikasi, tanpa pengukuran latensi pengiriman secara kuantitatif. Batasan-batasan tersebut menunjukkan bahwa hasil penelitian ini merepresentasikan kinerja sistem pada skala dan kondisi terbatas, serta membuka peluang pengembangan dan pengujian lanjutan pada lingkungan yang lebih kompleks.

Kesimpulan

Penelitian ini berhasil menghasilkan prototipe sistem pendeteksi ketinggian air berbasis ESP32, sensor ultrasonik HC-SR04, indikator LED, OLED I2C 0,96 inci, dan model TinyML yang mampu melakukan pemantauan serta klasifikasi kondisi air secara real-time ke dalam kategori AMAN, WASPADA, dan BANJIR.

Berdasarkan uji fungsional, sistem menunjukkan kesesuaian keluaran LED, OLED, dan hasil klasifikasi model terhadap status yang diharapkan pada 10 skenario uji. Pada uji akurasi sensor, perhitungan ketinggian air menunjukkan error maksimum 0,2 cm dan MAE 0,025 cm pada 8 titik uji, sehingga pembacaan sensor dinilai stabil dan layak digunakan sebagai data input sistem klasifikasi berbasis TinyML. Pengujian notifikasi Telegram menunjukkan bahwa pesan

peringatan terkirim pada transisi status peringatan dan tidak terkirim pada kondisi aman sesuai rancangan.

Secara keseluruhan, hasil penelitian menunjukkan bahwa integrasi IoT dan TinyML dengan inferensi langsung pada perangkat (on-device inference) berpotensi mendukung sistem pemantauan dan peringatan dini banjir pada skala kecil hingga menengah, seperti lingkungan permukiman, saluran drainase, atau kolam retensi, serta relevan digunakan oleh masyarakat dan pengelola lingkungan yang membutuhkan solusi monitoring ketinggian air yang sederhana dan responsif. Pengembangan selanjutnya dapat diarahkan pada peningkatan kualitas evaluasi model TinyML, analisis kinerja sistem notifikasi, serta pengujian prototipe pada lingkungan nyata dengan kondisi permukaan air yang lebih dinamis.

Daftar Pustaka

- Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., Huang, X., Hurtado, R., Kanter, D., Lokhmotov, A., Patterson, D., Pau, D., Seo, J., Sieracki, J., Thakker, U., Verhelst, M., & Yadav, P. (2021). *Benchmarking TinyML Systems: Challenges and Direction*. <http://arxiv.org/abs/2003.04821>
- Fauziyah, A., Saleh, I., & Usman, I. (2024). Pengembangan prototipe sistem monitoring level ketinggian air menggunakan sensor ultrasonik berbasis internet of things. *Jurnal Aplikasi Fisika*, 20(02), 1–8. <https://doi.org/10.62749/jaf.v20i02.p1-8>
- Hamzah, S. A., Dalimin, M. N., Som, M. M., Zainal, M. S., Ramli, K. N., Ikhsan, M. H. S., Yusop, A., Ishak, M., Rahman, R. A., Sidek, A., Utomo, W. M., Jamil, A. A., Talib, R., Yusoff, N. A., & Mustapa, M. S. (2024). Flood Level Detection System using Ultrasonic Sensor and ESP32 Camera: Preliminary Results. *Journal of Advanced Research in Applied Mechanics*, 119(1), 162–173. <https://doi.org/10.37934/aram.119.1.162173>
- Hasibuan, A., Zahiri, M., Jannah, M., Roid, F., Almunadiansyah, R., Abta, A., & Nnarth, I. M. A. (2025). Design of flood warning prototype using ESP32 module-based ultrasonic sensors. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, 14(1), 126. <https://doi.org/10.11591/ijres.v14.i1.pp126-135>
- Kurniawan, M. D., & Budjianto, A. (2025). Implementasi Sistem Programable Timer Menggunakan Esp32-C3 Supermini Dengan Tampilan Oled Dan Indikator Buzzer. *Jurnal Mahasiswa Teknik Informatika*, 9(6), 9950–9956.
- Lengkong, A. G., Salaki, D. T., & Alfonsius, E. (2025). *Implementation of An Internet of Things (IoT)-Based Water Level Early Warning System with Telegram Notification*. 6(2), 196–207. <https://doi.org/10.33365/jatika.v6i2.349>
- Mahardhika, P., Sari, Z., & Akbi, D. R. (2025). Rancang Bangun Prototype Sistem Monitoring Ketinggian Air Sungai Berbasis Mikrokontroler ESP32 dan Telegram Sebagai Upaya Deteksi Banjir Secara Dini. *Jurnal Repositor*, 7(4 SE-Sistem dan keamanan jaringan). <https://doi.org/10.22219/repositor.v7i4.40639>
- Oktarian, A. R., Tasmi, Antony, F., & Verano, D. A. (2023). RANCANG SISTEM MONITORING KUALITAS DAN KETINGGIAN AIR BERBASIS INTERNET OF THINGS (IOT). *JINI GLOBAL*, 1(2), 75–83. <https://doi.org/10.36982/jinig.v1i2.3649>

- Ovid, A., Yandani, E., & Asril. (2025). Prototype Sistem Pemantauan Tinggi Muka Air Berbasis Esp32 Untuk Sistem Pringatan Dini Banjir Melalui Telegram. *JEKIN (Jurnal Teknik Informatika)*, 5(3), 1081–1089. <https://doi.org/10.58794/jekin.v5i3.1634>
- Permana, I., & Irawati, R. (2025). Pengembangan Early Warning System Untuk Mitigasi Bencana Banjir Rob Di Indonesia. *Journal of Information Technology*, 5(2 SE-Articles), 300–307. <https://doi.org/10.46229/jifotech.v5i2.1039>
- Rienandie, N. F., & Pramudita, R. (2025). DESIGN OF AN INTERNET OF THINGS-BASED WATER LEVEL MONITORING SYSTEM. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 11(2), 265–272.
- Samanta, R., Saha, B., & Ghosh, S. K. (2025). A Low-Power Low-cost System for Disaster Locations Detection using ESP32 CAM and TinyML. *2025 17th International Conference on COMMunication Systems and NETWORKS (COMSNETS)*, 907–910. <https://doi.org/10.1109/COMSNETS63942.2025.10885754>
- Setiawan, I. P. E., Desnanjaya, I. G. M. N., Supartha, K. D. G., Ariana, A. A. G. B., & Putra, I. D. P. G. W. (2024). Implementation of Telegram Notification System for Motorbike Accidents Based on Internet Of Things. *Jurnal Galaksi*, 1(1), 1–11. <https://doi.org/10.70103/galaksi.v1i1.1>
- Wikantama, P. T., & Puspitasari, R. (2023). Perancangan Perangkat Pengukur Ketinggian Banjir dengan ESP32 dan Telegram Berbasis IoT. *Elektriase: Jurnal Sains Dan Teknologi Elektro*, 13(02), 107–114. <https://doi.org/10.47709/elektriase.v13i02.3108>
- Zakariah, M. A., Alfriani, V., & Zakariah, K. M. (2020). Metodologi Penelitian Kualitatif, Kuantitatif, Mix Method. *Research and Development. In Madani Media. Yayasan Pondok Pesantren Al Mawaddah Warrahmah Kolaka.*