SMART AQUARIUM BERBASIS *INTERNET OF THINGS* DAN *MACHINE LEARNING* SEBAGAI PREDIKSI PERGANTIAN AIR

SKRIPSI

DISUSUN OLEH

DUWI PRAYETNO

2109020168

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

2026

SMART AQUARIUM BERBASIS *INTERNET OF THINGS* DAN *MACHINE LEARNING* SEBAGAI PREDIKSI PERGANTIAN AIR

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

Duwi Prayetno
NPM. 2109020168

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
2026

LEMBAR PENGESAHAN

Judul Skripsi	:	*SMART AQUARIUM* BERBASIS *INTERNET OF THINGS* DAN *MACHINE LEARNING* SEBAGAI PREDIKSI PERGANTIAN AIR
Nama Mahasiswa	:	Duwi Prayetno
NPM	:	2109020168
Program Studi	:	Teknologi Informasi

Menyetujui

Komisi Pembimbing

(Mulkan Azhari, S.Kom, M.Kom)

NIDN. 0108129402

Ketua Program Studi	**Dekan**
(Fatma Sari Hutagalung, S.Kom, M.Kom)	**(Dr. Al-Khowarizmi, S.Kom., M.Kom.)**
NIDN. 0117019301	**NIDN. 0127099201**

PERNYATAAN ORISINALITAS

SMART AQUARIUM BERBASIS *INTERNET OF THINGS* DAN *MACHINE LEARNING* SEBAGAI PREDIKSI PERGANTIAN AIR

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Agustus 2026

Yang membuat pernyataan



Duwi Prayetno

NPM. 2109020168

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Duwi Prayetno
NPM	: 2109020168
Program Studi	: Teknologi Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (Non-Exclusive Royalty free Right) atas penelitian skripsi saya yang berjudul:

***SMART AQUARIUM* BERBASIS *INTERNET OF THINGS* DAN *MACHINE LEARNING* SEBAGAI PREDIKSI PERGANTIAN AIR**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Agustus 2026
Yang membuat pernyataan



Duwi Prayetno
NPM. 2109020168

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Duwi Prayetno
Tempat dan Tanggal Lahir	: Medan, 13 November 2002
Alamat Rumah	: Jl. Gaperta No.247 Medan
Telepon/Faks/HP	: 081265157606
E-mail	: dwiprayetno15.dp@gmail.com

DATA PENDIDIKAN

SD	: SD KARTIKA 1-1	TAMAT: 2015
SMP	: SMP NEGERI 16	TAMAT: 2018
SMA	: SMA SWASTA PANCABUDI	TAMAT: 2021

KATA PENGANTAR



Puji dan syukur penulis panjatkan kehadirat Allah SWT atas segala rahmat, nikmat, dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul **"*Smart Aquarium* Berbasis *Internet of Things* dan *Machine Learning* Sebagai Prediksi Pergantian Air"** dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara.

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Akrim, M.Pd., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Assoc. Prof. Dr. Al - Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Dr. Firahmi Rizky, S.Kom.,M.Kom Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
4. Bapak Mhd. Basri, S.Si, M.Kom Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
5. Ibu Fatma Sari Hutagalung, S.Kom,M.Kom Ketua Program Studi Teknologi Informasi
6. Bapak Okvi Nugroho, S.Kom., M.Kom Sekretaris Program Studi Teknologi Informasi.
7. Pembimbing Bapak Mulkan Azhari, S.Kom.,M.Kom.
8. Teristimewa kepada kedua orang tua dan keluarga penulis yang senantiasa memberikan doa, dukungan, motivasi, serta bantuan baik secara moril maupun materil selama proses perkuliahan hingga penyelesaian skripsi ini.

9. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan keterbatasan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi penyempurnaan skripsi ini. Semoga skripsi ini dapat memberikan manfaat bagi penulis maupun bagi pembaca serta dapat menjadi referensi dan bahan pengembangan bagi penelitian selanjutnya.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan, dukungan, dan doa selama proses penyusunan skripsi ini. Semoga segala kebaikan yang telah diberikan mendapatkan balasan yang terbaik dari Allah SWT.

SMART AQUARIUM BERBASIS *INTERNET OF THINGS* DAN *MACHINE LEARNING* SEBAGAI PREDIKSI PERGANTIAN AIR

ABSTRAK

Kualitas air merupakan faktor penting dalam pemeliharaan ikan hias karena secara langsung memengaruhi kesehatan ikan dan stabilitas ekosistem akuarium. Parameter seperti pH, suhu, dan kekeruhan harus dipantau secara teratur untuk menjaga kondisi air yang sesuai. Namun, pemantauan kualitas air dan penjadwalan penggantian air sering dilakukan secara manual sehingga dapat menyebabkan keterlambatan respons ketika kualitas air memburuk. Penelitian ini bertujuan untuk mengembangkan sistem *Smart Aquarium* berbasis *Internet of Things* (IoT) dan *Machine Learning* untuk pemantauan kualitas air secara *real-time* dan prediksi penggantian air. Sistem ini menggunakan sensor pH, DS18B20 untuk mengukur suhu, dan sensor kekeruhan yang terhubung dengan mikrokontroler ESP32 untuk mengumpulkan data kualitas air secara terus-menerus. Data sensor ditransmisikan ke server berbasis Flask dan disimpan dalam basis data MySQL untuk analisis lebih lanjut. Algoritma *Decision Tree* digunakan untuk mengklasifikasikan kondisi air ke dalam kategori ideal dan tidak ideal, sedangkan *Gradient Boosting* digunakan untuk memprediksi sisa hari sebelum penggantian air diperlukan. Hasil eksperimen menunjukkan bahwa model *Decision Tree* mencapai akurasi sebesar 100%, sedangkan model *Gradient Boosting* memperoleh nilai *Root Mean Squared Error* (RMSE) sebesar 0,44. Sistem yang dikembangkan mampu memantau kualitas air dan memberikan prediksi penggantian air, memberikan informasi mengenai kondisi air, serta membantu pengguna dalam menentukan waktu optimal untuk penggantian air secara lebih efektif dan efisien.

Kata Kunci:*Internet of Things*,*Smart Aquarium,Gradient Boosting, Water Quality Monitoring,Decision Tree*

Smart Aquarium Based on Internet of Things and Machine Learning for Water Change Prediction

ABSTRACT

Water quality is a critical factor in ornamental fish maintenance because it directly affects fish health and aquarium ecosystem stability. Parameters such as pH, temperature, and turbidity must be monitored regularly to maintain suitable water conditions.However, water quality monitoring and water replacement scheduling are often performed manually, which may lead to delayed responses when water quality deteriorates. This study aims to develop an Internet of Things (IoT) and Machine Learning-based Smart Aquarium system for real-time water quality monitoring and water replacement prediction. The system utilizes pH, DS18B20 temperature, and turbidity sensors connected to an ESP32 microcontroller to continuously collect water quality data. Sensor data are transmitted to a Flask-based server and stored in a MySQL database for further analysis. A Decision Tree algorithm is employed to classify water conditions into ideal and non-ideal categories, while a Gradient Boosting Regressor is used to predict the remaining days before water replacement is required. Experimental results show that the Decision Tree model achieved an accuracy of 100%, while the Gradient Boosting Regressor obtained a Root Mean Squared Error (RMSE) of 0.44. The developed system is capable of monitoring water quality and providing water replacement predictions, providing information about water conditions, and assisting users in determining the optimal time for water replacement more effectively and efficiently.

Keywords:Internet of Things,Smart Aquarium,Gradient Boosting, Water Quality Monitoring,Decision Tree

DAFTAR ISI

LEMBAR PENGESAHAN ..... **Error! Bookmark not defined.**
PERNYATAAN ORISINALITAS ..... **Error! Bookmark not defined.**
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS ..... **Error! Bookmark not defined.**
RIWAYAT HIDUP i
KATA PENGANTAR v
ABSTRAK vii
ABSTRACT viii
DAFTAR ISI ix
DAFTAR TABEL xi
DAFTAR GAMBAR xii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 5
1.5 Manfaat Penelitian 5
BAB II LANDASAN TEORI 6
2.1 Aquarium 6
2.2 Komponen *Hardware* yang Digunakan 8
2.3 *Software* 15
2.4 *Internet of Thing* (IoT) 16
2.5 Database 17
2.6 *Machine Learning* 17
2.7 *Flask* 19
2.8 Penelitian Terdahulu 20
BAB III METODOLOGI PENELITIAN 21
3.1 Jenis Penelitian 21
3.2 Waktu dan Tempat Penelitian 21
3.3 Alur Penelitian 23
3.3.1 Perancangan Sistem IoT 23

3.3.2 Pengumpulan dan Pengolahan Data 25
3.3.3 Perancangan dan Pemodelan Sistem Informasi 27
3.3.4 Analisis Sistem 35
3.4 Metode Analisis Masalah : Diagram *Fishbone* 36
3.4.1 Manusia 37
3.4.2 Metode 37
3.4.3 Lingkungan 38
3.4.4 Peralatan 38
BAB IV HASIL DAN PEMBAHASAN 39
4.1 Implementasi Sistem *Smart Aquarium* 39
4.1.1 Arsitektur Sistem 39
4.1.2 Implementasi *Backend Flask* 40
4.1.3 Implementasi *Database* 41
4.2 Pengujian Alat 41
4.2.1 Pengujian Sensor pH 42
4.2.2 Pengujian Sensor Kekeruhan 47
4.2.3 Pengujian Sensor Suhu 51
4.2.4 Pengujian Keseluruhan Alat 56
4.3 Pengujian Prediksi Hari Penggantian Air 58
4.3.1 Metode Prediksi 58
4.3.2 Hasil Prediksi 58
4.4 Visualisasi Monitoring Menggunakan *Node-RED* 59
4.4.1 Implementasi *Dashboard* 59
4.4.2 Hasil Visualisasi 59
4.5 Analisis Sistem 59
4.5.1 Analisis Sistem 59
4.5.2 Keterbatasan Sistem 60
4.5.3 Pembahasan Hasil 61
4.6 Ringkasan BAB 61
BAB V KESIMPULAN & SARAN 62
5.1 Kesimpulan 62
5.2 Saran 63
DAFTAR PUSTAKA 65

DAFTAR TABEL

HALAMAN

Tabel 3. 1 Tahapan dan Waktu Penelitian 22

Tabel 3. 2 aquarium_data 34

Tabel 4. 1 Pengujian Sensor pH dengan pH 4.01 42

Tabel 4. 2 Pengujian Sensor pH dengan pH *Buffer* 6.8 44

Tabel 4. 3 Pengujian sensor pH dengan pH *Buffer* 9.1 46

Tabel 4. 4 . Pengujian Sensor Kekeruhan Dengan Cahaya 48

Tabel 4. 5 Pengujian Sensor Kekeruhan Tanpa Cahaya 50

Tabel 4. 6 Pengujian Sensor Suhu Dengan Suhu Ruangan 51

Tabel 4. 7 Pengujian Sensor Suhu Dengan Es 53

Tabel 4. 8 Pengujian Sensor Suhu Dengan Api 55

Tabel 4. 9 Pengujian Keseluruhan Alat 57

DAFTAR GAMBAR

HALAMAN

Gambar 2. 1 Pinout Diagram ESP32 9

Gambar 2. 2 Konverter Digital ke Analog Dasar 11

Gambar 2. 3 Skema Sensor Suhu DS18B20 14

Gambar 2. 4 Sensor *Turbidity* 15

Gambar 3. 1 Blok Diagram Alat 24

Gambar 3. 2 *Smart Aquarium Prototype* 25

Gambar 3. 3 *Flowchart* Sistem 28

Gambar 3. 4 *Use Case* Diagram 30

Gambar 3. 5 *Wireframe* 35

Gambar 3. 6 Diagram *Fishbone* 36

Gambar 4. 1 Pengujian sensor pH menggunakan pH *Buffer* 4.01 42

Gambar 4. 2 Pengujian sensor pH Meenggunakan pH *Buffer* 6.8 44

Gambar 4. 3 Pengujian Sensor pH Menggunakan pH *Buffer* 9.18 46

Gambar 4. 4 Pengujian Sensor Kekeruhan Dengan Memberikan Cahaya 48

Gambar 4. 5 Pengujian Sensor Kekeruhan Dengan Menutup Sensor 49

Gambar 4. 6 Pengujian Sensor Suhu Dengan Suhu Ruangan 51

Gambar 4. 7 Pengujian Sensor Suhu Menggunakan Es 52

Gambar 4. 8 Pengujian Sensor Suhu Menggunaka Api 54

Gambar 4. 9 Pengujian Keseluruhan ALat 56

BAB I

PENDAHULUAN

1.1 Latar Belakang

Akuarium merupakan salah satu media yang banyak digunakan untuk memelihara ikan hias dan berbagai organisme air dalam lingkungan buatan. Dalam pemeliharaannya, kualitas air menjadi salah satu faktor yang perlu diperhatikan karena kondisi air dapat berubah seiring dengan waktu dan aktivitas organisme di dalam akuarium. Air yang jarang diganti atau tidak dipantau secara berkala dapat mengalami penurunan kualitas sehingga kondisi lingkungan di dalam akuarium menjadi kurang sesuai bagi ikan dan organisme air yang dipelihara. Oleh karena itu, pergantian air perlu dilakukan dengan mempertimbangkan kondisi kualitas air, bukan hanya berdasarkan perkiraan waktu semata.

Pada praktiknya, proses pergantian air pada akuarium masih sering dilakukan secara manual. Pemilik akuarium biasanya melakukan pemeriksaan dengan melihat kondisi air secara langsung atau berdasarkan kebiasaan dan perkiraan waktu tertentu. Cara tersebut memiliki keterbatasan karena perubahan kualitas air tidak selalu dapat diketahui hanya melalui pengamatan visual. Beberapa parameter kualitas air seperti tingkat keasaman (pH), suhu, dan kekeruhan perlu diukur untuk mengetahui kondisi air secara lebih objektif. Apabila pemantauan tidak dilakukan secara berkala, pemilik akuarium dapat terlambat mengetahui adanya perubahan kualitas air dan terlambat melakukan tindakan perawatan.

Salah satu parameter yang perlu diperhatikan dalam pemeliharaan ikan air tawar adalah pH air. Berdasarkan penelitian Wasito et al. (2024), air yang sesuai untuk ikan air tawar memiliki nilai pH pada rentang 6,5–8,5. Selain pH, suhu air juga menjadi parameter penting karena kondisi suhu dapat memengaruhi lingkungan hidup ikan. Taufiq Tamam dkk. (2022) menyebutkan bahwa suhu yang sesuai untuk ikan air tawar berada pada rentang 26–30 °C.

Parameter lainnya adalah tingkat kekeruhan air. Air dengan tingkat kekeruhan yang tinggi dapat menunjukkan adanya partikel tersuspensi yang memengaruhi kondisi dan kejernihan air akuarium. Berdasarkan acuan yang digunakan dalam penelitian ini, batas maksimal kekeruhan air adalah 25 NTU (*Nephelometric Turbidity Unit*) (Prasetiyo et al., 2021). Ketiga parameter tersebut, yaitu pH, suhu, dan kekeruhan, digunakan dalam penelitian ini sebagai parameter untuk memantau kondisi kualitas air akuarium.

Permasalahan tersebut menunjukkan perlunya suatu sistem yang dapat membantu pemilik akuarium dalam melakukan pemantauan kualitas air secara berkala. Perkembangan teknologi *Internet of Things* (IoT) memungkinkan perangkat sensor diintegrasikan dengan mikrokontroler untuk membaca kondisi lingkungan dan mengirimkan data melalui jaringan. Dengan penerapan IoT, data kualitas air dapat dikirimkan ke server dan ditampilkan kepada pengguna sehingga proses pemantauan tidak hanya bergantung pada pemeriksaan manual.

Selain melakukan pemantauan, sistem juga dapat dikembangkan untuk memberikan estimasi waktu penggantian air berdasarkan data kualitas air yang diperoleh dari sensor. Pada penelitian ini, teknologi *Machine Learning* digunakan untuk melakukan prediksi berdasarkan data pH, suhu, dan kekeruhan air.

Algoritma *Gradient Boosting* digunakan untuk menghasilkan prediksi berupa estimasi hari tersisa sebelum dilakukan penggantian air. Dengan demikian, sistem tidak hanya memberikan informasi mengenai kondisi parameter air, tetapi juga memberikan estimasi yang dapat digunakan sebagai salah satu pertimbangan dalam menentukan waktu penggantian air.

Dalam penelitian ini, evaluasi kondisi kualitas air dilakukan menggunakan pendekatan rule-based berdasarkan batas parameter yang telah ditentukan, sedangkan algoritma *Gradient Boosting* digunakan khusus untuk melakukan prediksi hari tersisa sebelum penggantian air. Pemisahan fungsi tersebut dilakukan agar proses evaluasi kondisi air dan proses prediksi memiliki tujuan yang jelas dalam sistem.

Dengan menggabungkan teknologi *Internet of Things* dan *Machine Learning*, sistem *Smart Aquarium* yang dikembangkan diharapkan dapat membantu pemilik akuarium melakukan pemantauan kualitas air secara lebih teratur dan mengurangi ketergantungan terhadap pemeriksaan manual. Sistem ini menghasilkan *prototipe* yang mengintegrasikan sensor pH, suhu, dan kekeruhan dengan mikrokontroler ESP32, *server* berbasis *Flask*, *database* MySQL, serta *dashboard Node-RED* sebagai media pemantauan. Berdasarkan permasalahan tersebut, penelitian ini dilakukan dengan judul "*Smart Aquarium* Berbasis *Internet of Things* dan *Machine Learning* Sebagai Prediksi Pergantian Air.".

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang dan membangun sistem *Smart Aquarium* berbasis *Internet of Things* (IoT) menggunakan mikrokontroler ESP32 yang dapat memantau kualitas air akuarium berdasarkan parameter pH, suhu, dan kekeruhan air secara berkala?
2. Bagaimana mengimplementasikan algoritma *Machine Learning* berbasis regresi, yaitu *Gradient Boosting*, untuk memprediksi jumlah hari tersisa sebelum pergantian air berdasarkan data parameter pH, suhu, dan kekeruhan air?
3. Bagaimana mengintegrasikan sensor pH, sensor suhu, sensor kekeruhan, ESP32, *Internet of Things* (IoT), *server*, basis data, dan model *Gradient Boosting* sehingga dapat menghasilkan sistem monitoring dan prediksi pergantian air pada akuarium?

1.3 Batasan Masalah

Adapun batasan masalah yang di ambil pada penelitian ini adalah:

1. Sensor yang digunakan terbatas pada sensor suhu, sensor pH, dan sensor kekeruhan air. Karena untuk menekan biaya dalam melakukan penelitian ini.
2. Algoritma yang digunakan terbatas pada *Gradient Boosting* untuk prediksi waktu pergantian air.
3. Prediksi yang dilakukan oleh sistem difokuskan pada estimasi waktu optimal pergantian air berdasarkan parameter pH, suhu, dan kekeruhan air, dengan evaluasi kondisi air dilakukan menggunakan pendekatan *rule-based.*

4. ESP32 hanya sebagai pelaksana dari keseluruhan sistem, *Node-red* sebagai visual sistem yang berjalan, *server Flask* sebagai pengolah data, pelaksana *Machine Learning*, dan pemberi Keputusan.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu:

1. Menciptakan solusi praktis bagi penggemar *aquarium* untuk memantau dan menjaga kualitas air secara berkala tanpa perlu memeriksa secara manual.
2. Menggunakan algoritma *Machine Learning* (*Gradient Boosting*) untuk memprediksi waktu terbaik mengganti air berdasarkan data yang dikumpulkan dari sensor (pH, suhu, dan kekeruhan).
3. Memberikan informasi kondisi *aquarium* melalui *platform IoT (Internet of Things)* seperti *Node RED* yang dapat diakses dari perangkat.

1.5 Manfaat Penelitian

1. Menyediakan solusi yang praktis dan efisien bagi penggemar *aquarium* dalam menjaga kesehatan ikan hias melalui prediksi dan data yang mudah dipahami.
2. Menerapkan algoritma *Machine Learning* (*Gradient Boosting*) untuk memprediksi kapan waktu yang tepat untuk mengganti air berdasarkan data sensor suhu, pH, dan kekeruhan air.
3. Mengurangi risiko kesehatan ikan akibat kualitas air yang buruk, sehingga meningkatkan tingkat keberlangsungan hidup ikan hias di dalam *aquarium*.

4. Menggabungkan teknologi *IoT (Internet of Things)* dan *Machine Learning* untuk memudahkan dalam mengetahui kondisi air dan memprediksi pergantian air pada *aquarium*.

BAB II

LANDASAN TEORI

2.1 Aquarium

2.1.1 Pengertian *Aquarium*

Asal mula kata *aquarium* adalah dari Bahasa latin, yakni *aqua* yang berarti air, dan rium yang artinya wadah. Jadi, pengertian *aquarium* adalah " wadah ikan, tanaman dan organisme air untuk dilihat " (Damayanti et al., n.d.).

Aquarium adalah wadah atau tempat yang digunakan untuk menampung ikan, tanaman air, dan hewan air lainnya. Akuarium biasanya terbuat dari kaca atau plastik yang transparan dan diisi dengan air serta dilengkapi dengan filter, sistem pemanas, dan sistem pencahayaan (Emir Garden, 2023).

Berdasarkan dari kedua definisi tersebut peneliti menyimpulkan bahwa *Aquarium* merupakan sebuah tempat tinggal ikan yang terbuat dari kaca dan akrilik atau wadah bagi ikan dan organisme lainnya dalam lingkungan buatan.

2.1.2 Fungsi *Aquarium*

Berdasarkan definisi dari *aquarium* sebagai wadah atau tempat tinggal bagi ikan, selain dari hal tersebut *aquarium* berfungsi sebagai:

a) **Penghias Ruangan**

Fungsi *aquarium* selain sebagai wadah untuk budidaya ikan, *aquarium* juga dapat berfungsi sebagai penghias ruangan dimana keindahannya dapat dinikmati oleh penggemarnya (Meika Puspita Sari, 2019).

Ada sebagian pemilik *aquarium* kurang begitu memiliki waktu dalam melakukan perawatan namun menyukai dari *aquarium* dikarenakan keindahan ekosistem didalamnya dan dapat dilihat pada saat di dalam ruangan. Penempatan *aquarium* yang paling sering sebagai hiasan pada ruangan kerja atau di meja kerja.

b) **Kegiatan Penggemar Ikan Hias**

Penggemar ikan hias untuk menikmati keindahan dari ikan hias yang mereka miliki biasanya ikan hias akan di tempatan di dalam *aquarium* agar penggemar dapat melihat ikan hias mereka dari balik kaca *aquarium* dan lebih leluasa dalam melihat pergerakan ikan.

Penggemar ikan yang mengisi waktu luang di rumah dengan meihat *aquariuam* dipercaya dapat meningkatkan kesehatan dan ketenangan. Jumlah ikan yang banyak di dalam *aquarium* mampu membuat penggemar semakin fokus dalam sebuah perhatian. Studi ini sendiri dilakukan di sebuah wahana *aquarium* besar berkapasitas 550.000 liter (Galih Agus Saputra, 2022).

2.1.3 Jenis *Aquarium*

Fungsi dan kebutuhan dari *aquarium* menjadikan *aquarium* memiliki berbagai jenis, tergantung dari bagaimana ekosistem akan berjalan di dalam *aquarium* itu sendiri. Ekosistem yang terbentuk akan menentukan jenis dari *aquarium* itu sendiri. Salah satu jenis *aquarium,* yaitu:

a. *Aquarium* Air Tawar

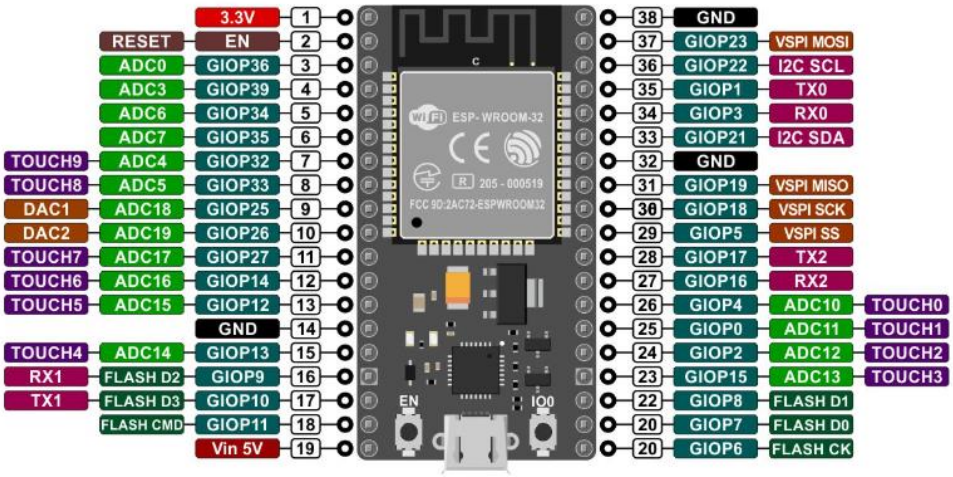
Inti dari pada *aquarium* air tawar adalah ekosistem yang terbentuk dari lingkungan air tawar, seperti rawa, sungai, bahkan danau. Ekosistem tersebut berbagai aspek yang diperlu dipertimbangkan mulai dari tanaman yang digunakan, tempat, arus air, suhu air, dan pH air, tidak kalah penting bakteri yang diperlukan di dalam ekosistem tersebut.

2.2 Komponen *Hardware* yang Digunakan

2.2.1 ESP32

ESP32 merupakan salah satu dari jenis Mikrokontroler, Mikrokontroler merupakan perangkat elekronik yang memiliki fungsi sistem dengan memproses data lalu menjalankan perintah terhadap sensor ataupun alat. ESP 32 adalah mikrokontroler yang dikenalkan oleh *Espressif System* merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dalam *chip* sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things* (Imran & Rasul, 2020).

Dengan hal ini memungkinkan dalam pembuatan suatu alat yang mudah melalui pemograman langsung mnggunakan Arduino IDE, dengan desain dan ukuran yang relatif kecil hal ini memberikan kemudahan dalam membuat atau penempatan sehingga ukuran alat nantinya tidak terlalu besar. ESP32 memiliki Wifi *mode* ganda 2,4 GHz dan koneksi nirkabel BT (*Bluetooth*), dengan SRAM 512 KB dan memori *flash* 4MB yang sudah terdapat di mikrokontroler. Untuk pin pin pada ESP32 dapat di lihat pada gambar.



Gambar 2. 1 Pinout Diagram ESP32

ESP32 ini adalah sistem minimal yang cukup hemat biaya, berdaya rendah dengan Wifi dan *Bluetooth*. ESP32 ini juga memilik banyak pin dengan fungsi dari masing masing yang berbeda, yaitu diantaranya:

a. Pin Daya

Pada ESP32 terdapat 3 umtuk pin daya, yaitu:

a) 3.3v (Pin 1)

Pada pin ini ESP32 mengeluarkan daya 3,3Volt untuk disalurkan ke sensor nantinya.

b) Vin (Pin 19)

Berbeda dengan PIN 1 pada pin ini adalah sumber daya yang masuk ke ESP32, bisa berasal dari USB, atau sumber daya lain sebesar 5 Volt.

c) GND (pin 12,20,37,38)

Pada pin ini adalah *Ground* atau daya negatif, digunakan sebagai referensi tegangan, atau juga arus balik tegangan .

b. GPIO (General Purpose Input/Output)

Pada ESP32 memiliki banyak sekali pin *GPIO* (*General Purpose Input*/*Output*) yang biasa digunakan untuk mengontrol sensor, relay, dan *LED*.

a) GPIO36-GPIO39

Pada GPIO ini hanya bisa digunakan sebagai input dari analog ADC (*Analog-to-Digital Converter*)

b) GPIO0-GPIO35

Pada GPIO ini bisa digunakan sebagai *input* atau *output* digital.

c) GPIO6- GPIO11

Pada GPIO jenis ini digunakan untuk *flash memory*. *Flash memory* adalah jenis memori *non-volatile* yang digunakan untuk menyimpan program dan data pada ESP32. *Non-volatile* berarti tetap tersimpan meskipun daya dimatikan. Seperti layaknya HDD atau SSD pada komputer, namun dengan skala lebih kecil.

c. ADC (Analog-to-Digital Converter)

Pada ESP32 terdapat pin ADC (*Analog-to-Digital Converter*) yang digunakan untuk membaca beberapa sensor analog seperti pada sensor suhu, dan pH. ADC 0 – ADC 19 digunakan untuk membaca nilai analog. Pada ADC 0 – ADC 10 dapat digunakan kapan saja. Namun pada ADC 12 – ADC 19 akan bermasalah jika Wifi aktif, karena akan digunakan oleh sistem Wifi internal ESP32. Total terdapat 18 channel ADC (*Analog-to-Digital Converter*).

d. DAC *(Digital-to-Analog Converter)*

Pada ESP32 memiliki 2 pin DAC (*Digtal-to-Analog Converter*) yang dapat mengubah nilai digital menjadi sinyal analog, DAC1, dan DAC2 pada pin 9 dan 10.



Gambar 2. 2 Konverter Digital ke Analog Dasar

Konverter Digital ke Analog adalah perangkat yang mengubah nilai digital (0 dan 1) menjadi serangkaian tegangan analog kontinu (ElectronicWings, 2024).

Pada ESP32 hanya akan mengerti dengan bahasa atau angka biner kombinasi angka 0 dan 1 dengan kata lain harus mengubah dari angka biner tersebut menjadi gelombang listrik, pada saat ini lah DAC (*Digital to Analog Converter*) bekerja.

e. Touch Sensor Pins

ESP32 memiliki *Touch Sensor Pins*, yang dapat digunakan unuk mendeteksi sentuhan tanpa tombol fisik. Cara kerjanya dengan memanfaatkan sensor kapasitif bawaan pada ESP32 yang bekerja dengan prinsip perubahan kapasitansi. Sensor ini bekerja dengan mendeteksi perubahan kapasitansi objek yang berada di sekitar sensor.

f. UART (Universal Asynchronous Receiver-Transmitter)

UART (*Universal Asynchronous Receiver-Transmitter*) digunakan untuk komunikasi serial, misalnya dengan sensor. UART adalah protokol komunikasi serial asinkron yang digunakan mengirim dan menerima data

antara dua perangkat elektronik. Komunikasi serial asinkron adalah metode pengiriman data bit-per-bit melalui satu jalur data tanpa menggunakan sinyal eksternal .

g. I2C (Inter-Integrated Circuit)

Pada ESP32 terdapat pin yang mendukung I2C (*Inter-Integrated Circuit*) I2C adalah protokol *interface* yang dikembangkan oleh *Philips Semiconductor* dengan konsep dasar komunikasi dua arah antar IC menggunakan dua kabel. Protokol ini mengizinkan desain sistem untuk saling terkoneksi sampai 128 piranti yang berlainan hanya dengan 2 jalur dua arah, satu untuk *serial clock* (SCL), dan satunya untuk serial data (SDA) (Kusumah & Pradana, n.d.).

Dengan kata lain setiap yang terhubung dalam I2C memiliki alamat yang unik dan dapat diakses secara perangkat keras. Dengan 2 jalur dua arah menjadikannya lebih efisien sehingga I2C sering digunakan diberbagai proyek IoT, sensor, LCD, dan perangkat perangkat lainnya. Dikarenakan hanya dua jalur untuk beberapa perangkat.

h. SPI (Serial Peripheral Interface)

SPI adalah standar komunikasi sinkron data serial yang dikenalkan oleh Motorola yang bekerja pada *mode full duplex*. SPI merupakan *high-speed synchronous serial input/output* (I/O) port yang memungkinkan untuk pengaturan lebar data yang akan digeser masuk atau keluar dari *device* dan juga memungkinkan pengaturan pada kecepatan transfer data (Kusumah & Pradana, n.d.).

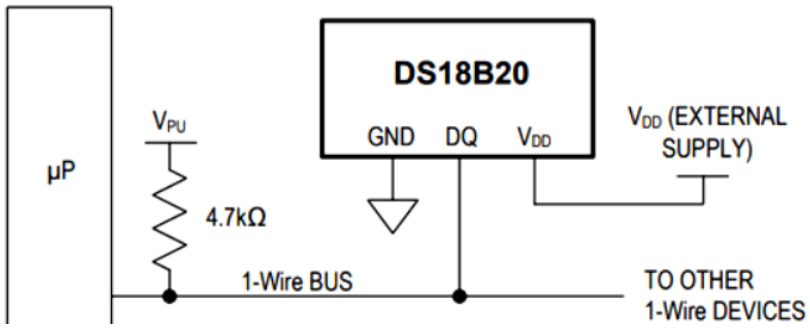
SPI dapat mengirim dan menerima data secara bersamaan (*full duplex*) hal ini memungkinkan untuk bertukar data menjadi lebih cepat. Karena kecapatannya yang tinggi SPI sering digunakan untuk sensor yang keperluannya cepat. Selai itu SPI lebih fleksibel karena dapat mengatur lebar data dan kecepatan transfer sesuai dengan kebutuhan. Kelemahan dari I2C ini memerlukan banyak kabel dalam penerapannya. Dengan kata lain SPI cocok dalam komunikasi yang membutuhkan kecepatan dan dapat mengirim dan menerima data secara bersamaan.

2.2.2 Sensor pH (pH 4502C)

Pada penelitian ini nantinya akan menggunakan sensor pH 4502C.Alat ukur derajat keasaman (pH meter) adalah sebuah alat elektronik yang digunakan untuk mengukur pH (derajat keasaman atau kebasaan) dari suatu cairan. Alat ukur kadar keasaman (pH meter) biasa terdiri dari pengukuran yang terhubung pada sebuah alat elektronik yang mengukur dan menampilkan nilai pH. Prinsip dasar pengukuran pH dengan menggunakan pH meter adalah potensial elektrokimia yang terjadi antara larutan yang terdapat di dalam elektroda gelas yang telah telah diketahui dengan larutan yang terdapat di luar elektroda gelas yang tidak diketahui. Hal ini dikarenakan lapisan tipis dari gelembung kaca akan berinteraksi dengan ion hidrogen yang ukurannya relatif kecil dan aktif Skema elektroda pH meter akan mengukur potensial listrik antara merkuri klorid (HgCl) pada elektroda pembanding dan *potassium chloride* (KCl) yang merupakan larutan di dalam gelas elektroda serta potensial antara larutan dan elektroda perak (*SKRIPSI RAHMAT DONGORAN Terbaru7*, n.d.).

2.2.3 Sensor suhu DS18B20

DS18B20 memiliki tingkat akurasi yang cukup tinggi, yaitu 0,5°C pada rentang suhu -10°C sampai +85°C. Sensor suhu pada umumnya membutuhkan ADC dan beberapa pin port pada mikrokontroler, namun DS18B20 ini tidak membutuhkan ADC agar dapat berkomunikasi dengan mikrokontroler dan hanya membutuhkan 1 *wire* saja (Nurazizah et al., n.d.).



Gambar 2. 3 Skema Sensor Suhu DS18B20

Mikrokontroler pada gambar 2.3 ditandai dengan (µP), DS18B20 merupakan sensor suhu yang memiliki tiga pin, GND (*Ground*) yang terhubung ke pada mikrokontroler, DQ (Data) yang terhubung ke mikrokontroler sebagai jalur komunikasi, VDD (*Power Supply*) yang terhubung langsung ke sumber tegangan *external* yang biasanya 3.3v atau 5v. Resistor disimbolkan dengan Vpu. Resistor 4,7 kΩ digunakan sebagai menjaga kestabilan komunikasi pada jalur data.

2.2.4 Sensor Kekeruhan

Turbidity Sensor (sensor kekeruhan) adalah perangkat elektronik yang dirancang untuk mendeteksi jumlah partikel tersuspensi dalam air. Partikel ini bisa berupa lumpur, tanah liat, mikroorganisme, atau bahan organik lainnya. Kekeruhan diukur dalam unit NTU (*Nephelometric Turbidity Units*), yang

mengindikasikan seberapa banyak cahaya yang terhambur oleh partikel-partikel tersebut dalam air (Admin Alatuji, 2024).



Gambar 2. 4 Sensor Turbidity

Cara kerja dari sensor turbidity ini berdasarkan penghamburan cahaya. Sumber cahaya di hamburkan kedalam air. Sensor lalu mengukur intensitas cahaya yang terhambur ke berbagai arah. Semakin banyak pertikel yang menghambat cahaya maka semakin tinggi tingkat kekeruhan yang terukur.

2.3 Software

2.3.1 Visual Studio Code

Visulal Studio Code atau VS Code merupakan suatu *editor code* yang dikembangkan oleh Microsoft yang berfokuskan pada kecepatan, keringanan dan keterbukaan dalam memberikan pengalaman pengembangan yang tangkas dan responsif. Editor ini mendukung berbagai macam bahasa seperti Java, JavaScript, Python, C# dan lain-lain. Sistem editor ini juga mendukung sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows(Hasya Akhlaqah Harmie, 2024).

2.3.2 XAMPP

XAMPP adalah paket instalasi instan yang diantaranya berisi Apache, PHP dan MySQL yang dapat digunakan untuk membantu instalasi ketiga produk tersebut. XAMPP adalah media lokal atau web server yang dapat digunakan secara *offline* (Website et al., 2023). Pada penelitian ini nantinya yang akan digunakan adalah MySQL sebagai *database* atau tempat menyimpan data dari sensor yang sudah terintegrasi dengan ESP32.

2.4 Internet of Thing (IoT)

Internet of Things (IOT) adalah struktur dimana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer. *Internet of Things* (IOT) merupakan perkembangan keilmuan yang sangat menjanjikan untuk mengoptimalkan kehidupan berdasarkan sensor cerdas dan peralatan pintar yang bekerjasama melalui jaringan internet (Selay et al., 2022).

2.4.1 NodeRED

Node-RED adalah sebuah tool berbasis browser untuk membuat aplikasi *Internet of Things* (IoT) yang mana lingkungan pemrograman visualnya mempermudah penggunanya untuk membuat aplikasi sebagai "*flow*". Bahasa pemrograman sangatlah luas dan meliputi berbagai jenis gaya dan paradigma pemrograman. Bahasa imperatif berorientasi objek saat ini menguasai dunia pemrograman, namun begitu sebetulnya ada alternatif untuk pengembangan atau produksi *software* dan juga untuk membuat prototipe ide

dengan cepat. Node-RED mengambil jalur alternatif tersebut untuk pengembangan *software*. Pertama adalah bahasa pemrograman visual. Daripada membuat aplikasi sebagai barisan kodingan, Node-RED fokus ke program sebagai *flow* (Mulyono et al., 2018).

2.5 Database

Database adalah sebuah sistem yang di buat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. *Database* terdiri dari kumpulan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. *Database* digital di *manage* menggunakan *Database Management System* (DBMS), yang menyimpan isi *database*, mengizinkan pembuatan dan *maintenance* data dan pencarian dan akses yang lain (Ramadhan & Mukhaiyar, 2020).

2.5.1 MySQL

MySQL (*My Structure Query Language*) merupakan sistem manajemen basis data relasional. RDBMS (*Relational Database Management System*) didistribusikan, dikembangkan, dan didukung oleh *Oracle Corporation* (Website et al., 2023). MySQL bekerja dengan cara menyimpan dan mengelola data dalam bentuk tabel yang terdiri dari baris dan kolom.

2.6 Machine Learning

Machine Learning (ML) merupakan bidang studi yang fokus kepada desain dan analisis algoritma sehingga memungkinkan komputer untuk dapat belajar. Menurut Samuel, ML berisi sebuah algoritma yang bersifat *generic* (umum) dimana algoritma tersebut dapat menghasilkan sesuatu yang menarik atau

bermanfaat dari sejumlah data tanpa harus menulis kode yang spesifik. Pada intinya, algoritma yang generik tersebut ketika diberikan sejumlah data maka ia dapat membangun sebuah aturan atau model atau inferensi dari data tersebut. Sebagai contoh sebuah Algoritma untuk mengenali tulisan tangan dapat digunakan untuk mendeteksi email yang berisi *spam* dan bukan *spam* tanpa mengganti kode. Algoritma yang sama ketika diberikan data pelatihan yang berbeda menghasilkan logika klasifikasi yang berbeda (Ibnu Daqiqi Id, 2021).

2.6.1 Regresi *Gradient Boosting*

Gradient boosting merupakan metode pengembangan dari *Recurrent Neural Network* (RNN) yang bertujuan untuk membuat setiap *recurrent* unit bisa dapat menangkap hubungan (dependensi) dalam skala waktu yang berbeda-beda secara adaptif (Putra Nasyuli et al., 2023). Selain itu, *Gradient Boosting* merupakan algoritma regresi yang fleksibel karena mengadopsi sifat-sifat dari *regression tree* yang memiliki interpretabilitas tinggi konsepnya yang sederhana dan efisiensi komputasi dengan pendekatan *boosting* yang dapat meningkatkan akurasi faktor mode (Saputra et al., 2025) .

Untuk menilai dari kinerja Model yang digunaan penelitian ini menggunakan MSE (Mean Squared Error) dan RMSE (Root Mean Squared Error).

1. MSE (*Mean Squared Error*)

   MSE merupakan metrik yang menghitung rataan .dari selisih kuadrat antara nilai prediksi dan nilai aktual (Saputra et al., 2025). Rumus untuk mencarai MSE adalah sebagai berikut:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - F(X_i))^2 \qquad 2.\ 1$$

Penjelasan:

n : merupakan jumlah keseluruhan observasi.

Yi : menunjukkan nilai aktual.

F(Xi) : merupakan nilai prediksi yang dihasilkan oleh model untuk observasi ke-i.

$(Yi - F(Xi))^2$: merupakan perbedaan absolut antara nilai aktual dengan nilai prediksi.

$\sum$: menyatakan jumlah keseluruhan dari setiap perbedaan absolut.

1/n : merupakan kebalikan dari jumlah keseluruhan observasi, yang berfungsi sebagai rata-rata dari perbedaan absolut.

2. *Root Mean Squared Error* (RMSE)

Root Mean Squared Error (RMSE) atau yang disebut salah satu metrik yang dipakai untuk menilai .seberapa akurat model dalam melakukan prediksi Metrik ini diperoleh dengan menghitung .akar kuadrat dari *Mean Squared Error* (MSE). Adapun rumus RMSE adalah sebagai berikut:

$$RMSE = \sqrt{(MSE)} \qquad 2.\ 2$$

Penjelasan:

MSE : merupakan Mean Squared Error.

$\sqrt{\ }$: merupakan fungsi akar kuadrat.

2.7 Flask

Flask adalah *micro-framework Python* yang digunakan untuk pengembangan aplikasi berbasis web. *Flask* mudah dan banyak digunakan untuk

pengembangan *website* berbasis Python(ngodingdata, 2021). *Flask* merupakan *website framework* yang mengunakan bahasa *Python* dan tergolong sebagai jenis *microframework*. *Flask* mempunyai fungsi sebagai kerangka kerja aplikasi dan tampilan suatu *website*, dengan menggunakan *Flask* dan *python* seseorang dapat membuat *website* dengan terstruktur dan dapat mengatur tampilan dengan mudah (Ganda Saputra et al., n.d.) .

2.8 Penelitian Terdahulu

Pada penelitian relevan (Abdurohman et al., 2022) yang berjudul *"A Robust Internet of Things-Based Aquarium Control System Using Decision Tree Regression Algorithm"*pada penelitiannya membahas mengenai suhu pada air. Dengan sistem otomatis untuk menaikkan atau menurunkan suhu air pada *aquarium.* Dengan menggunakan algoritma *Decision Tree Regression* sebagai prediksi suhu pada air. Pada penelitiannya mengandalkan sensor suhu, TDS, sensor oksigen terlarut mengirim datanya ke sever melalui protokol MQTT. Hasil pengujian dari sistem memiliki akurasi prediksi suhu yang tinggi (99,82%). Penelitiannya memiliki kemampuan untuk mengirimkan notifikasi otomatis kepada pengguna saat kondisi air mendekati titik kritis tanpa perlu pemantauan terus menerus.

BAB III

METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Jenis pada penelitian ini adalah kuantitatif dengan metode pendekatan Observasi. Data dikumpulkan melalui sensor yang terintegrasi dengan sistem *Internet of Things* (IoT) dan *Gradient Boosting* sebagai pengolahan data pada prediksi pergantian air *aquarium.*

3.2 Waktu dan Tempat Penelitian

3.2.1 Waktu Penelitian

Tabel 3. 1 Tahapan dan Waktu Penelitian

| No | Keterangan | 2025 | | |

3.2.2 Tempat Penelitian

Tempat penelitian ini dilakukan di tempat tinggal peneliti yang berlokasi di Jl Gaperta, Kecamatan Medan Helvetia, Kelurahan Helvetia Tengah. Dengan tujuan untuk mengamati langsung kondisi dan parameter air akuarium serta melakukan pengujian sistem IoT *Gradient Boosting* sebagai prediksi pergantian air, dalam kondisi lingkungan yang nyata.

3.3 Alur Penelitian

3.3.1 Perancangan Sistem IoT

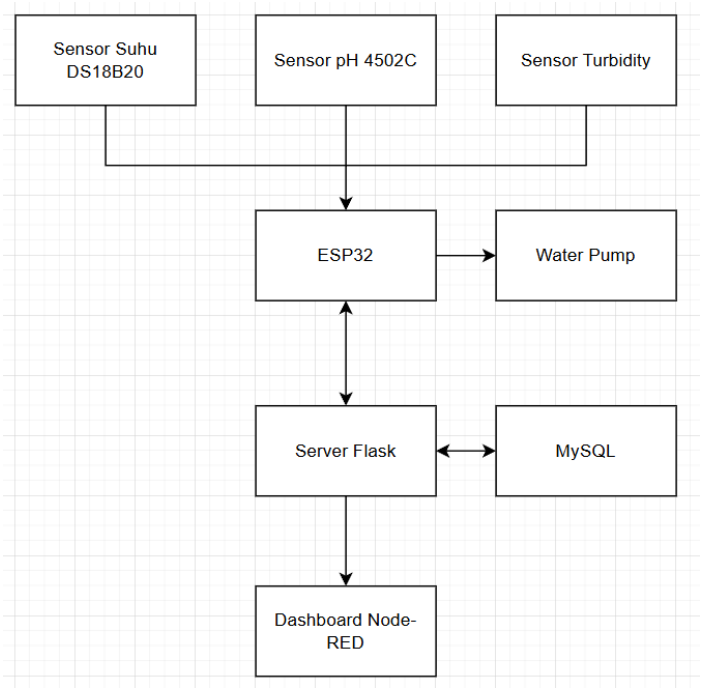
a. Peralatan dan Bahan Penelitian

Pada penelitian ini peneliti membutuhkan alat dan bahan mencakup:

a. ESP32: digunakan sebagai mikrokontroler dan sebagai pengendal dari sistem alat nantinya.
b. Sensor DS18B20: sebagai pendeteksi suhu air pada akuarium.
c. Sensor pH 4502C: sebagai pendeteksi pH pada air akuarium.
d. Sensor *Turbidity*: sebagai pendeteksi kekeruhan pada air akuarium.
e. *Case* alat: sebagai pelindung alat dari percikan air untuk mengurangi kemungkinan terjadinya *korsluiting* pada alat nantinya.
f. *Adaptor* 5V 2A: sebagai penyuplai daya keperangkat.
g. *Water Pump* Celup 3-5V DC: digunakan untuk melakukan proses pergantian air.
h. *Relay Module* 5V 2 *Channel*: sebagai penambah tegangan *Water Pump.*

b. Perancangan Bangun Alat

1) Blok Diagram

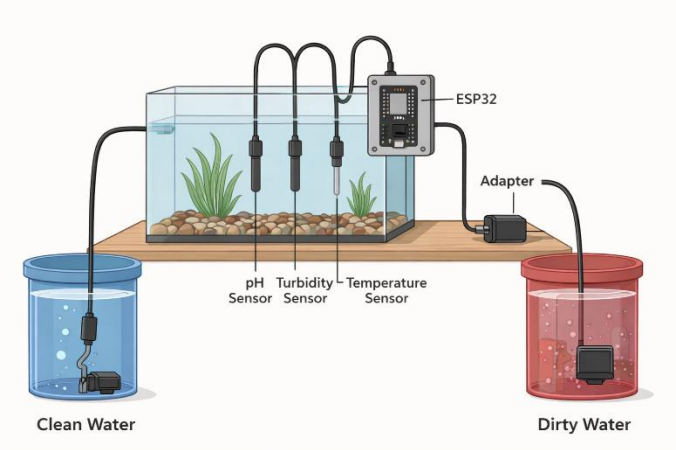


Gambar 3. 1 Blok Diagram Alat

Pada gambar diatas terdapat beberapa blok yang dari setiap bloknya memiliki tujuan untuk mencapai suatu koordinasi dan tujuan yang ingin dicapai, yaitu *input* , proses, dan *output*.

Pada ketiga sensor tersebut yaitu sebagai *input* dari parameter air *aquarium,* diteruskan ke ESP32 lalu dihubungkan dengan *server* dengan menggunakan protokol HTTP dan metode POST untuk mengirim data ke *python server* (*Flask*). Pada *server* nantinya data mentah sensor dari ESP32 akan dimasukkan ke *database* juga *server* melakukan klasifikasi dan prediksi dengan lalu hasilnya akan di masukkan ke *database*. *Node-RED* akan mengakses data ke *database* melalui server *flask* untuk ditampilkan pada *dashboard*, dan server akan mengembalikan status *rule* kembali ke ESP32 untuk menjalankan *Water Pump*

2) Rancangan Alat



Gambar 3. 2 Smart Aquarium Prototype

Pada Gambar 3.2 merupakan rancangan dari Smart *Aquarium*. Pada gambar terdapat ketiga sensor dan *Water Pump* berada di air *aquarium* sebagai pendeteksi dari setiap parameter. Untuk ESP32 terdapat pada luar bagian dari aquarium dan di lindungi cover untuk mengurangi terjadinya arus pendek listrik. Untuk *power* nantinya akan menggunakan *adaptor* 5v sebagai penyuplai daya kepada ESP32 nantinya. Juga terdapat 2 penampung air bersih dan kotor yang sudah dilengkapi *Water pump.*

3.3.2 Pengumpulan dan Pengolahan Data

a. Jenis dan Sumber Data

Pada penelitian ini data yang akan digunakan merupakan data kuantitatif yang diperoleh secara primer atau diperoleh secara langsung dari pembaca sensor pH, suhu, dan kekeruhan air, yang diselaraskan dengan ESP32. Data hasil prediksi waktu untuk pergantian air diperoleh dari pemrosesan algoritma *Gradient Boosting* yang dijalankan pada *server* berbasis *Flask*. Seluruh data disimpan pada database MySQL untuk keperluan visualisasi pada *dashboard Node-RED*.

b. Teknik Pengumpulan Data

Teknik pengumpulan data pada penelitian ini dilakukan melalui integrasi sistem menggunakan ESP32 yang terhubung dengan sensor pH, suhu, dan kekeruhan air. Data dikirim ke *server Flask* menggunakan protokol HTTP POST, data diolah dengan model *Gradient Boosting*, kemudian data disimpan didalam *database* MySQL. Pengumpulan data dilakukan secara terus menerus selama periode pengamatan guna memperoleh data yang cukup untuk proses pelatihan dan pengujian model *Machine Learning.*

c. Analisis Data

Data yang telah dikumpulkan dari sensor akan dianalisis menggunakan algoritma *Gradient Boosting*. Data yang digunakan mencakup parameter-parameter kualitas air seperti pH, suhu, dan kekeruhan. Analisis dilakukan untuk menentukan apakah kondisi air akuarium masih ideal atau tidak ideal. Hasil analisis berupa prediksi hari tersisa untuk melakukan pergantian air akan disimpan dalam *database* dan ditampilkan pada *dashboard Website* menggunakan *Node-RED*.

d. Deskripsi Data Sensor

Data yang digunakan dalam penelitian ini terdiri dari tiga parameter utama yang berfungsi untuk menentukan kualitas air pada akuarium, yaitu:

a. pH Air

Nilai optimal untuk pH air pada ikan hias air tawar adalah 6,5 - 8,5.

b. Suhu Air

Untuk nilai optimal pada suhu air yang sesuai dengan ikan hias air tawar adalah 26-30 derajat *celcius.*

c. Kekeruhan Air

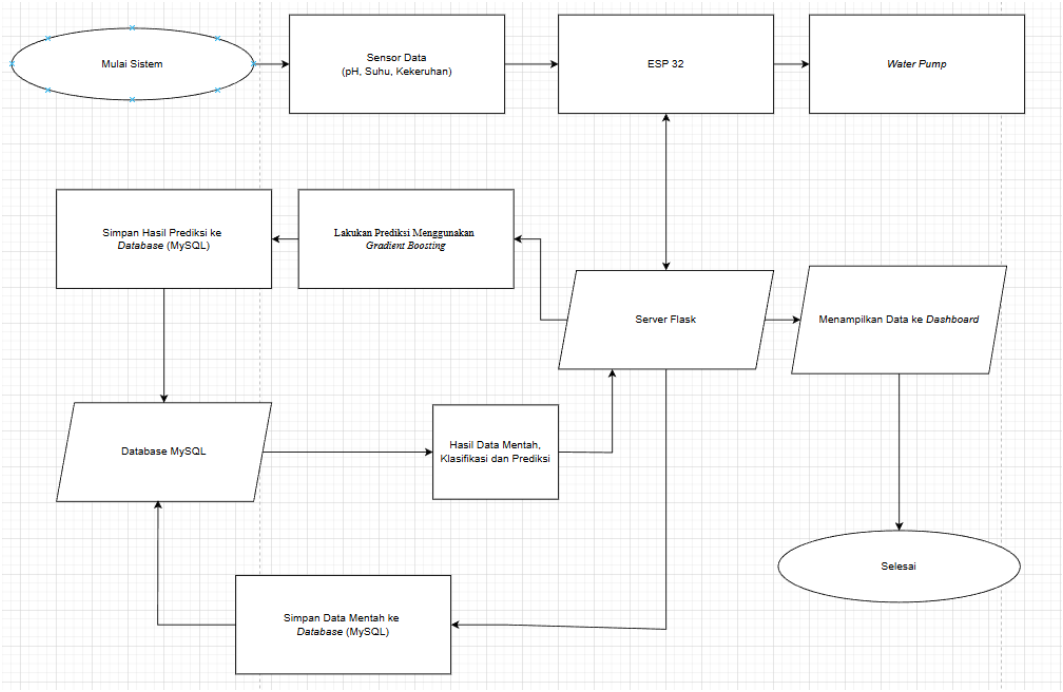
Untuk nilai optimal kekeruhan air pada ikan hias air tawar adalah dibawah 25 NTU (Nephelometric Turbidity Unit).

Setiap parameter tersebut dikumpulkan oleh sensor-sensor yang terhubung ke mikrokontroler ESP32. Data dikirim ke *server Flask* untuk diproses menggunakan algoritma *Gradient Boosting*. *Output* dari model *Gradient Boosting* prediksi hari tersisa untuk melakukan pergantian air. Data mentah dan hasil prediksi juga disimpan di dalam database MySQL dan digunakan untuk visualisasi di *dashboard*.

3.3.3 Perancangan dan Pemodelan Sistem Informasi

a. *Flowchart* Sistem

Flowchart sistem pada penelitian ini menggambarkan alur kerja sistem monitoring kualitas air *Aquarium* berbasis IoT. *Flowchart* ini bertujuan untuk memudahkan proses yang terjadi mulai dari pembacaan data pada sensor hingga prediksi dan penyimpanan hasil ke dalam sistem.



Gambar 3. 3 Flowchart Sistem

Pada Gambar 3.3 terdapat *Flowchart* Sistem dari *Smart Aquarium* ini, berikut penjelasan dari *Flowchart* tersebut:

1) Mulai Sistem

   Pada tahap ini merupakan booting dari perangkat ESP32, pengaktifan server *flask*, serta koneksi ke sensor dan *database*.

2) Sensor Data (pH, Suhu, Kekeruhan)

   Sensor yang terhubung ke ESP32 melakukan pembacaan parameter air yaitu pH, Suhu, Kekeruhan. Data ini akan menjadi input utama dalam proses pengolahan dan analisis kuaitas air.

3) ESP32

ESP32 sebagai pengontrol utama yang mengumpulkan data dari sensor, mengirimkan data data sensor tersebut ke server melalui protokol HTTP (*Metode* POST)

4) Server Flask

Flask disini sebagai server *backend*, setelah menerima data dari ESP32, data sensor diolah lalu data mentah dan hasil prediksi di kirim ke *database*, klasifikasi dan prediksi dilakukan menggunakan model *Machine Learning* dengan bantuan *TensorFlow* untuk mengimplementasikan model di server. Data mentah juga akan dikirim ke *database*.

5) Lakukan Prediksi Menggunakan *Gradient Boosting*

Data yang masuk ke server akan diproses dengan dua model *Machine Learning* yaitu *Gradient Boosting Regressor* untuk memprediksi berapa hari lagi air utuk diganti.

6) Simpan Data Mentah ke *Database* (MySQL)

Data mentah dari ESP32 (tanpa diproses) disimpan ke tabel (*tabel_sensor),* sebagai bahan pemantauan historis.

7) Simpan Hasil Prediksi ke Database (MySQL)

Hasil dari klasifikasi disimpan pada tabel (*tabel_klasifikasi)* dan untuk prediksi disimpan ke tabel (*tabel_prediksi)* pada *database*. Untuk tabel klasifikasi label : ideal / tidak ideal, dan untuk prediksi hari tersisa: x hari lagi.

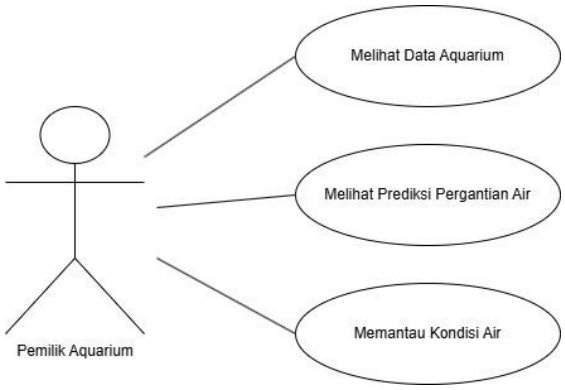
8) Menampilkan Data ke *Dashboard*

Data mentah dan data hasil *Machine Learning* pada database ditampilkan ke *Dashboard Node-RED*, yang ditampilkan adalah : Nilai pH, suhu, kekeruhan, kondisi air (ideal/tidak ideal), dan hari tersisa.

9) Selesai

Proses berakhir hingga data ditampilkan. Sistem akan terus berjalan secara berulang, membaca sensor dan memproses data baru secara berkala.

b. *Use Case* Diagram



Gambar 3. 4 Use Case Diagram

Pada Gambar 3.4 merupakan *Use Case* Diagram yang digunakan untuk menggambarkan interaksi antar pengguna sistem dan fungsi fungsi utama. Diagram ini mempermudah dalam memahami peran pengguna dan fitur yang dapat diakses selama proses pemantauan kualitas air. Pada sistem ini pemilik aquarium terdapat tiga *Use Case* utama yang menjadi bagian dari sistem yaitu:

1) Melihat Data *Aquarium*

Pemilik akuarium dapat melihat data hasil pembacaan sensor secara berkala, yang meliputi parameter penting seperti pH, suhu air, dan

kekeruhan. Data ini ditampilkan dalam bentuk numerik pada *dashboard*, sehingga memudahkan pengguna dalam memahami kondisi air secara berkala.

2) Melihat Prediksi Pergantian Air

Sistem menyediakan melihat kondisi air dan prediksi mengenai pergantian air menggunakan algoritma *Machine Learning*. *Gradient Boosting* utuk memprediksi berapa hari lagi air akan menjadi tidak layak. Prediksi ini membantu pemilik dari *aquarium* dalam menentukan kapan melakukan pergantian air sehingga kualitas air di *aquarium* selalu terjaga dan kematian ikan dapat diminimalkan.

3) Memantau Kondisi Air

Fitur ini memungkinkan pengguna untuk memantau perubahan kualitas air dari hari kehari, serta melihat riwayat data sensor dan hasil prediksi di hari sebelumnya. Jika air terdeteksi dalam kondisi tidak ideal, sistem dapat memberikan peringatan atau notifikasi kepada pengguna melalui *dashboard*.

Use Case Diagram ini menunjukkan bahwa sistem dirancang untuk memberikan kemudahan dan informasi akurat kepada pemilik akuarium, sehingga proses perawatan air dapat dilakukan secara efektif, efisien, dan berbasis data.

c. Dataset

1) Pembuatan Data

Pada penelitian ini dilakukan Perancangan dataset sintetis dengan alasan alat belum jadi atau belum dibuat. Data sintetis ini dibuat

menyerupai aslinya jika data dari alat sudah jadi nantinya. Dataset sintetis dibuat bertujuan untuk membuktikan bahwa model *Machne Learning* bekerja. Ketika alat sudah dibuat atau sudah jadi nantinya data akan diambil sepenuhnya dari alat. Untuk jumlah data sintetis ini berjumlah 2500 data.

d. Pra Pemrosesan Data

a. Penghapusan kolom yang tidak relevan yaitu kolom Tanggal Pergantian Air tidak dibutuhkan untuk model *Machine Learning* karena tidak bersifat nomerik yang membantu dalam melakukan prediksi.
b. Untuk Regresi (*Gradient Boosting*) targetnya adalah hari tersisa. Fitur yang digunakan untuk kedua model tersebut Nilai pH, Nilai Suhu, Nilai Kekeruhan.
c. Dataset digunakan untuk melatih model regresi Gradient Boosting dengan target variabel Hari_Tersisa berdasarkan parameter pH, suhu, dan kekeruhan air.

e. *Training Machine Learning (Gradient Boosting)*

1) Regresi Prediksi Pergantian Air (*Gradient Boosting*)

Disini peneliti memprediksi "HariTersisa" dalam sistem *Smart Aquarium* menggunakan algoritma *Gradient Boosting* berdasarkan tiga fitur yaitu nilai pH, suhu air, kekeruhan air. Alasan peneliti menggunakan model ini adalah untuk mencegah terjadinya *overfitting*. Juga peneliti telah melakukan beberapa *training* pada model model lain yaitu:

- *Random forest*: MSE 0,22, RMSE 0,47
- *Gradient bosting* : MSE 0.20, RMSE 0.44
- *XGBoost*: MSE 0.26, RMSE 0.51
- SVR : MSE 0.24, RMSE 0.49
- *Neural Network* : MSE 0.58, RMSE 0.76
- KNN : MSE 0.56, RMSE 0.75
- *LightGBM* : MSE 0.20, RMSE 0.45
- *ElasticNet* : MSE 0.60, RMSE 0.77

Hasil MSE (*Mean Squared Error*) dan RMSE (*Root Mean Squared Error*) yang paling rendah adalah *Gradient Boosting* sehingga peneliti memilih untuk menggunakan *Gradient Boosting* sebagai prediksi Hari tersisa.

Proses *training Gradient Boosting* dataset akan dibagi menjadi dua bagian *Training Set* (80%) digunakan untuk melatih model *Machne Learning* dan *Testing Set* (20%) digunakan untuk menguji performa model terhadap data baru (yang belum dilatih). Model membaca fitur dari *X_train* adalah nilai pH, nilai suhu, nilai kekeruhan. Model mencocokkan dengan nilai *Y_train* adalah Hari Tersisa. *Gradient Boosting* bekerja dengan membangun banyak *tree kecil* (*Decision Trees*) secara berurutan. Setiap *tree* baru memperbaiki kesalahan prediksi dari *tree* sebelumnya.

2) Evaluasi Model

a) Evaluasi *Gradient Boosting* (Regresi)

Setelah mendapatkan hasil prediksi (*y_pred*) dari data uji (*X_test*), membandingkannya dengan nilai aktual (*y_test*). *Mean Squared Error* (MSE) dan *Root Mean Squared Error* (RMSE) digunakan untuk mengukur seberapa besar kesalahan prediksi antara nilai yang diprediksi dan nilai aktual Hari Tersisa. MSE mengukur rata-rata kuadrat selisih antara prediksi dan nilai aktual, semakin kecil nilai MSE, semakin baik model dalam memprediksi target. Bertujuan untuk menilai seberapa baik model dalam memprediksi Hari Tersisa dengan akurasi regresi yang tinggi. RMSE memudahkan interpretasi karena nilainya langsung merepresentasikan rata-rata kesalahan dalam satuan target.

f. Perancangan Database

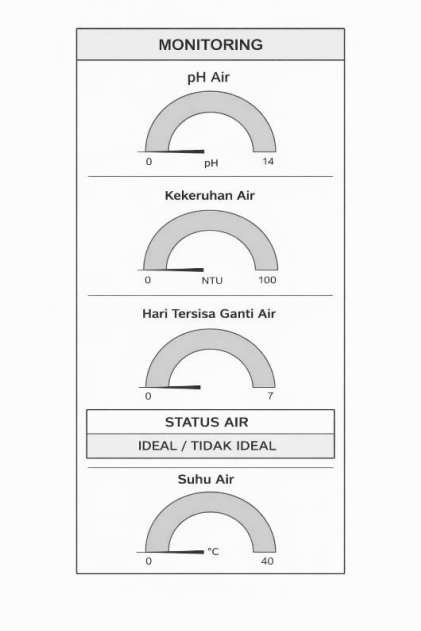
1) Tabel aquarium_data

Tabel 3. 2 aquarium_data

Nama Kolom	Tipe Data	Keterangan
id	INT(PK)	*Primary Key*
timestamp	DATETIME	Tanggal dan waktu pengambilan data
pH	FLOAT	Nilai pH Air
Suhu	INT	Suhu Air
Kekeruhan	INT	Tingkat Kekeruhan Air
raw_json JSON	JSON	Data mentah sensor dalam format JSON
hasil_prediksi	TINYINT	1 = Ideal, 0 = Tidak Ideal
hari_tersisa	INT	Hari tersisa

Nama Kolom	Tipe Data	Keterangan
model_used	VARCHAR(100)	Model yang dijalankan
status_prediksi	VARCHAR(20)	Hasil kodisi klasifikasi
status_rule	VARCHAR(20)	Hasil kondisi menggunakan rule

g. Perancangan Antarmuka Pengguna (*Wireframe*)



Gambar 3. 5 Wireframe

3.3.4 Analisis Sistem

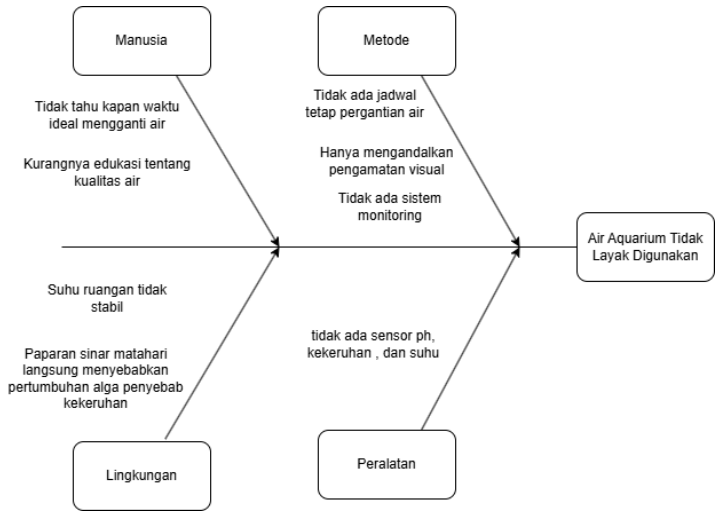
Analisis sistem dilakukan untuk mengetahui bagaimana seluruh komponen dalam sistem Smart Aquarium bekerja secara menyeluruh. Sistem ini terdiri dari perangkat keras berupa ESP32 dan tiga sensor utama (sensor pH, sensor suhu, dan sensor kekeruhan), serta perangkat lunak yang meliputi *server Flask*, *database* MySQL, dan *dashboard* visualisasi *Node-RED*. Data dari sensor akan dikirim ke *server Flask* melalui koneksi WiFi

menggunakan metode HTTP POST, Selanjutnya, data tersebut akan dianalisis menggunakan model klasifikasi *Decision Tree* untuk memberikan label ideal atau tidak ideal dan model regresi *Gradient Boosting* untuk menentukan status kondisi air serta prediksi hari tersisa sebelum air perlu diganti, kemudian disimpan ke dalam MySQL. Hasil prediksi dan data kemudian ditampilkan melalui *dashboard Node-RED.*

a. Hasil Analisis

Hasil dari analisis sistem dituangkan dalam bentuk diagram alir dan diagram blok sistem. Diagram alir menggambarkan proses yang terjadi mulai dari pembacaan data sensor, pengiriman data ke *server*, proses klasifikasi dan prediksi oleh model *Machine Learning*, penyimpanan data ke dalam *database*, hingga penyajian data melalui *dashboard*. Diagram blok sistem menunjukkan keterkaitan antara komponen ESP32, sensor, *server Flask*, *MySQL*, dan *Node-RED*. Ini memudahkan dalam memahami hubungan antar bagian dalam sistem serta alur data secara keseluruhan.

3.4 Metode Analisis Masalah : Diagram *Fishbone*



Gambar 3. 6 Diagram Fishbone

Diagram *Fishbone* atau diagram sebab-akibat digunakan dalam penelitian ini untuk membantu menganalisis penyebab utama dari masalah air akuarium yang tidak layak digunakan. Metode ini digunakan untuk mengidentifikasi berbagai faktor yang memengaruhi turunnya kualitas air secara terstruktur. Hasil dari analisis ini menjadi dasar dalam merancang solusi berupa sistem *Internet of Things* (IoT) yang dapat memantau kualitas air secara otomatis, serta menggunakan *Machine Learning* untuk mengetahui kualitas air dan memprediksi kapan waktu terbaik untuk mengganti air. Diagram *Fishbone* yang ditampilkan pada Gambar 3.6 memetakan empat kategori utama penyebab masalah:

3.4.1 Manusia

Masalah yang berasal dari faktor manusia biasanya disebabkan karena kurangnya pengetahuan. Banyak pemilik akuarium tidak tahu kapan waktu yang tepat untuk mengganti air. Selain itu, minimnya pemahaman tentang pentingnya kualitas air juga menjadi alasan utama kenapa perawatan air sering diabaikan atau dilakukan hanya saat air sudah terlihat sangat kotor. Padahal masih ada beberapa parameter air yang harus di terapkan untuk ikan dapat hidup di akuarium.

3.4.2 Metode

Dari sisi metode, pergantian air sering kali tidak dilakukan secara terjadwal dan hanya berdasarkan pengamatan mata saja. Tidak adanya sistem atau cara khusus untuk memantau kondisi air membuat keputusan untuk mengganti air menjadi tidak akurat. Hal ini dapat menyebabkan air terlalu lama tidak diganti hingga akhirnya membahayakan bagi ikan.

3.4.3 Lingkungan

Lingkungan sekitar juga bisa memengaruhi kualitas air di dalam akuarium. Misalnya, suhu ruangan yang berubah-ubah bisa memengaruhi suhu air. Selain itu, jika akuarium terkena sinar matahari langsung, maka pertumbuhan alga bisa meningkat dan menyebabkan air menjadi keruh lebih cepat.

3.4.4 Peralatan

Permasalahan juga muncul karena tidak tersedianya peralatan yang memadai. Misalnya, tidak adanya sensor untuk mengukur pH, suhu, dan kekeruhan air membuat pemilik akuarium tidak tahu kondisi air secara akurat. Padahal, alat-alat tersebut bisa membantu memberikan informasi penting untuk menentukan kapan air perlu diganti.

Dengan menganalisis keempat faktor di atas, peneliti menyimpulkan bahwa perlu adanya sistem pemantauan otomatis berbasis sensor *Internet of Things* (IoT) yang dapat mendeteksi kualitas air secara berkala, serta memanfaatkan *Machine Learning* untuk memprediksi kapan air perlu diganti. Hal ini diharapkan bisa membantu pemilik akuarium merawat air dengan lebih mudah dan tepat waktu.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas hasil implementasi dan pengujian sistem *Smart Aquarium* yang telah dirancang pada bab sebelumnya. Pengujian dilakukan untuk mengetahui kinerja sensor, proses pengiriman dan penyimpanan data, klasifikasi kualitas air, prediksi hari penggantian air, serta visualisasi *monitoring* sistem.

4.1 Implementasi Sistem Smart Aquarium

Implementasi sistem *Smart Aquarium* meliputi arsitektur sistem, backend server berbasis *Flask*, serta perancangan *database* sebagai media penyimpanan data sensor dan hasil pemrosesan.

4.1.1 Arsitektur Sistem

Sistem *Smart Aquarium* dirancang menggunakan arsitektur terdistribusi yang terdiri dari beberapa komponen utama, yaitu perangkat sensor, *mikrokontroler* ESP32, *server backend* berbasis *Flask, database* MySQL, serta sistem visualisasi menggunakan Node-RED.

Sensor pH, suhu air, dan kekeruhan air terhubung langsung ke mikrokontroler ESP32. Mikrokontroler berfungsi sebagai perangkat akuisisi data yang membaca nilai sensor secara periodik dan mengirimkan data tersebut ke server backend melalui protokol HTTP dalam format JSON.

Server backend Flask bertugas menerima data sensor dari ESP32, melakukan pemrosesan menggunakan metode *Machine Learning* dan *rule-based logic*, serta menyimpan hasil pemrosesan ke dalam *database* MySQL. Data yang

telah disimpan kemudian diambil oleh Node-RED untuk ditampilkan dalam bentuk *dashboard monitoring*.

Alur kerja dari sistem diawal ESP32 membaca data Ph, suhu, dan kekeruhan air, lalu data sensor dikirim ke *server Flask* melalui *endpoint* API, *server Flask* melakukan klasifikasi menggunakan *machine learning*. *Server Flask* mengevaluasi kualitas air menggunakan *rule-based logic*. *Server Flask* juga melakukan prediksi hari penggantian air menggunkan *Gradient Boosting*, lalu seluruh hasil diproses dan disimpan ke *database* MySQL, Node-RED mengambil data dari *database* dan menampilkan hasil *monitoring*.

4.1.2 Implementasi *Backend Flask*

Backend sistem dibangun menggunakan *framework Flask* yang berfungsi sebagai pusat pemrosesan data sensor dan integrasi antara perangkat IoT, *machine learning*, *database*, serta sistem visualisasi.

Flask digunakan untuk menyediakan layanan API berbasis HTTP yang menerima data sensor dari *mikrokontroler* ESP32 dalam format JSON. Data yang diterima kemudian diproses melalui beberapa tahapan, yaitu validasi data, penyimpanan data mentah ke *database*, klasifikasi kualitas air, serta prediksi hari penggantian air.

Proses klasifikasi kualitas air dilakukan menggunakan model *machine learning* yang telah dilatih sebelumnya berdasarkan parameter pH, suhu air, dan tingkat kekeruhan. Selain itu, *backend Flask* juga menerapkan *rule-based logic* untuk mengevaluasi kondisi air berdasarkan ambang batas parameter yang telah ditentukan. Hasil klasifikasi dan evaluasi ini digunakan untuk menentukan status kualitas air akuarium.

Untuk prediksi hari penggantian air, *backend Flask* menggunakan algoritma *Gradient Boosting* yang memanfaatkan data historis sensor sebagai input. Hasil prediksi berupa estimasi jumlah hari optimal sebelum dilakukan penggantian air akuarium.

Seluruh hasil pemrosesan, termasuk data sensor, hasil klasifikasi, status kualitas air, dan hasil prediksi penggantian air, disimpan ke dalam *database* MySQL dan dapat diakses oleh sistem visualisasi Node-RED.

4.1.3 Implementasi *Database*

Database yang digunakan dalam sistem *Smart Aquarium* adalah MySQL yang berfungsi sebagai media penyimpanan data sensor dan hasil pemrosesan sistem. *Database* dirancang untuk menyimpan data secara terstruktur sehingga dapat mendukung proses *monitoring*, dan analisis.

Tabel *database* menyimpan informasi berupa waktu pengambilan data, nilai pH air, suhu air, tingkat kekeruhan, hasil klasifikasi kualitas air, status evaluasi berdasarkan *rule-based logic*, serta hasil prediksi hari penggantian air. Penyimpanan data dilakukan secara otomatis setiap kali *server backend* menerima data baru dari ESP32.

Dengan penggunaan *database* MySQL, data historis sensor dapat digunakan kembali untuk proses pelatihan ulang model *machine learning*, analisis performa sistem, serta kebutuhan visualisasi *monitoring* pada *dashboard* Node-RED.

4.2 Pengujian Alat

Pengujian alat dilakukan untuk mengetahui kinerja dan akurasi sensor yang digunakan pada sistem *Smart Aquarium*. Pengujian ini bertujuan untuk

memastikan bahwa setiap sensor mampu membaca nilai parameter air dengan baik sebelum data digunakan dalam proses klasifikasi kualitas air dan prediksi hari penggantian air.

4.2.1 Pengujian Sensor pH

Pengujian ini dilakukan untuk mengetahui apakah sensor pH bekerja dengan semestinya. Pengujian dilakukan dengan bantuan pH *buffer* 4.01, 6.8, & 9.1 sebagai pedoman pengujian apakah sensor mendekati atau tidak dari nilai pH *buffer* tersebut.

4.2.2.1 Pengujian Menggunakan pH *Buffer* 4,01



Gambar 4. 1 Pengujian sensor pH menggunakan pH Buffer 4.01

Pengujian ini dilakukan menggunakan larutan pH *buffer* 4,01 Sensor pH dicelupkan ke dalam larutan *buffer*, kemudian sistem membaca nilai pH yang dikirimkan oleh sensor dan menyimpannya ke dalam *database* MySQL. Pengujian dilakukan beberapa kali untuk melihat kestabilan pembacaan sensor.

Tabel 4. 1 Pengujian Sensor pH dengan pH 4.01

No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-21 20:40:54	30.25	4	32	Tidak Ideal	Tidak Ideal
2	2026-01-21 20:41:04	30.25	4	33	Tidak Ideal	Tidak Ideal
3	2026-01-21 20:41:14	30.18	4	32	Tidak Ideal	Tidak Ideal
4	2026-01-21 20:41:25	30.25	4	33	Tidak Ideal	Tidak Ideal

Berdasarkan hasil pengujian pada Tabel 4.1, sensor pH mampu membaca nilai pH larutan *buffer* 4,01 dengan hasil pembacaan yang stabil pada nilai pH 4,00. Perbedaan nilai yang sangat kecil antara pH standar dan hasil pembacaan sensor menunjukkan bahwa sensor pH memiliki tingkat akurasi yang baik.

Nilai pH yang berada pada kondisi asam menyebabkan sistem mengklasifikasikan kualitas air sebagai Tidak Ideal, baik berdasarkan hasil klasifikasi *machine learning* maupun *rule-based logic.* Hal ini sesuai dengan kondisi larutan *buffer* 4,01 yang memang tidak berada pada rentang pH ideal untuk *aquarium*.

Dari hasil pengujian ini dapat disimpulkan bahwa sensor pH bekerja dengan baik dan mampu membaca nilai pH air sesuai dengan nilai referensi, sehingga layak digunakan pada sistem *Smart Aquarium.*

4.2.2.2 Pengujian Menggunakan pH *Buffer* 6,8



Gambar 4. 2 Pengujian sensor pH Meenggunakan pH Buffer 6.8

Pengujian ini dilakukan menggunakan larutan pH *buffer* 6,8 pada suhu 25°C. Pengujian bertujuan untuk mengetahui kemampuan sensor pH dalam membaca nilai pH netral serta kestabilan hasil pembacaan sensor. Sensor pH dicelupkan ke dalam larutan *buffer* dan sistem membaca nilai pH yang dihasilkan, kemudian data disimpan ke dalam *database* MySQL.

Tabel 4. 2 Pengujian Sensor pH dengan pH Buffer 6.8

No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-21 20:38:43	30.25	6.9	33	Ideal	Ideal
2	2026-01-21 20:38:45	30.18	6.9	33	Ideal	Ideal
3	2026-01-21 20:38:56	30.25	6.7	33	Ideal	Ideal
4	2026-01-21 20:39:06	30.25	6.7	33	Ideal	Ideal
5	2026-01-21 20:39:08	30.25	6.8	33	Ideal	Ideal

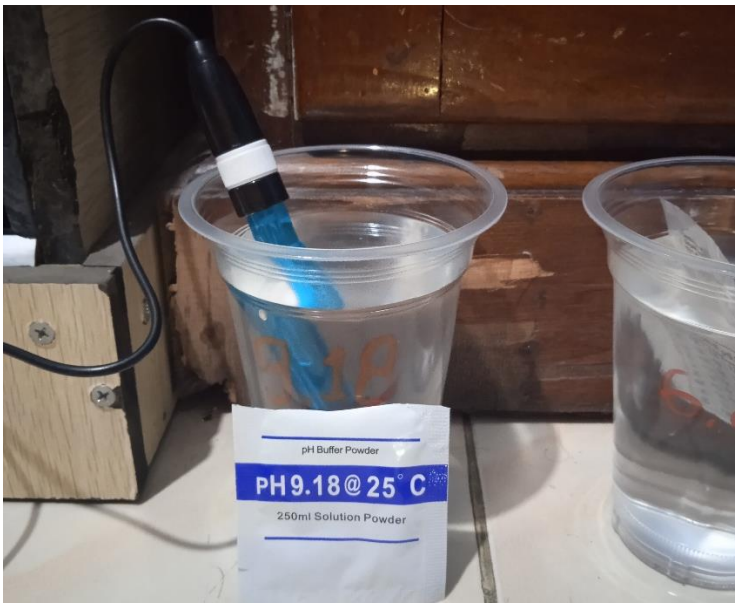
No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
6	2026-01-21 20:39:19	30.18	6.8	33	Ideal	Ideal

Berdasarkan hasil pengujian pada Tabel 4.2, sensor pH mampu membaca nilai pH larutan *buffer* 6,8 dengan hasil pembacaan yang relatif stabil, berada pada rentang pH 6,74 hingga 6,95. Selisih pembacaan yang kecil menunjukkan bahwa sensor pH memiliki tingkat akurasi yang baik dalam membaca pH mendekati nilai netral.

Nilai pH yang berada pada rentang tersebut diklasifikasikan oleh sistem sebagai Ideal, baik berdasarkan hasil klasifikasi *machine learning* maupun *rule-based logic*. Hal ini sesuai dengan kondisi pH buffer 6,8 yang mendekati rentang pH ideal untuk *aquarium*.

Dari hasil pengujian ini dapat disimpulkan bahwa sensor pH bekerja dengan baik pada kondisi pH netral dan layak digunakan sebagai parameter utama dalam sistem *Smart Aquarium*.

4.2.2.3 Pengujian Menggunakan pH *Buffer* 9,1



Gambar 4. 3 Pengujian Sensor pH Menggunakan pH Buffer 9.18

Pengujian ini dilakukan menggunakan larutan pH *buffer* 9,1 pada suhu 25°C. Tujuan pengujian adalah untuk mengetahui kemampuan sensor pH dalam membaca nilai pH basa serta kestabilan hasil pembacaan sensor. Sensor pH dicelupkan ke dalam larutan *buffer*, kemudian sistem membaca nilai pH yang dihasilkan dan menyimpannya ke dalam *database* MySQL.

Tabel 4. 3 Pengujian sensor pH dengan pH Buffer 9.1

No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-21 20:34:11	30.1	9.08	32	Ideal	Tidak Ideal
2	2026-01-21 20:34:	30.1	9.1	32	Ideal	Tidak Ideal
3	2026-01-21 20:34:	30.1	9.08	31	Ideal	Tidak Ideal
4	2026-01-21 20:34:	30.1	9.2	32	Ideal	Tidak Ideal

Berdasarkan hasil pengujian pada Tabel 4.3, sensor pH mampu membaca nilai pH larutan buffer 9,1 dengan hasil pembacaan yang berada pada rentang pH 9,08 hingga 9,20. Nilai tersebut menunjukkan bahwa

sensor pH dapat bekerja dengan baik dalam mendeteksi kondisi air yang bersifat basa.

Hasil klasifikasi menunjukkan perbedaan antara status prediksi machine learning dan rule-based logic. Model *machine learning* mengklasifikasikan kondisi air sebagai Ideal, sedangkan *rule-based logic* mengklasifikasikan kondisi air sebagai Tidak Ideal. Perbedaan ini disebabkan oleh pendekatan yang digunakan, di mana *rule-based logic* menggunakan batas ambang pH ideal akuarium yang lebih ketat, sedangkan model machine learning mempertimbangkan pola data historis secara keseluruhan.

Perbedaan hasil ini menunjukkan bahwa sistem mampu memberikan dua sudut pandang evaluasi kualitas air, sehingga pengguna dapat memperoleh informasi yang lebih komprehensif terkait kondisi air akuarium. Secara keseluruhan, sensor pH dapat membaca nilai pH basa dengan baik dan layak digunakan pada sistem *Smart Aquarium*.

4.2.2 Pengujian Sensor Kekeruhan

Pengujian sensor kekeruhan dilakukan untuk mengetahui kemampuan sensor dalam mendeteksi tingkat kekeruhan air pada sistem *Smart Aquarium*. Sensor kekeruhan digunakan sebagai parameter penting untuk menentukan kualitas air serta sebagai salah satu input dalam proses klasifikasi dan prediksi penggantian air.

4.2.2.1 Pengujian Dengan Menambahkan Cahaya (Ideal)



Gambar 4. 4 Pengujian Sensor Kekeruhan Dengan Memberikan Cahaya

Pengujian ini dilakukan dengan menambahkan sumber cahaya pada sensor kekeruhan. Kondisi ini mensimulasikan air dengan tingkat kejernihan yang baik, di mana cahaya dapat menembus air dengan optimal. Sensor kekeruhan dipasang dan diaktifkan, kemudian sistem membaca nilai kekeruhan dan menyimpan data hasil pengukuran ke dalam *database* MySQL.

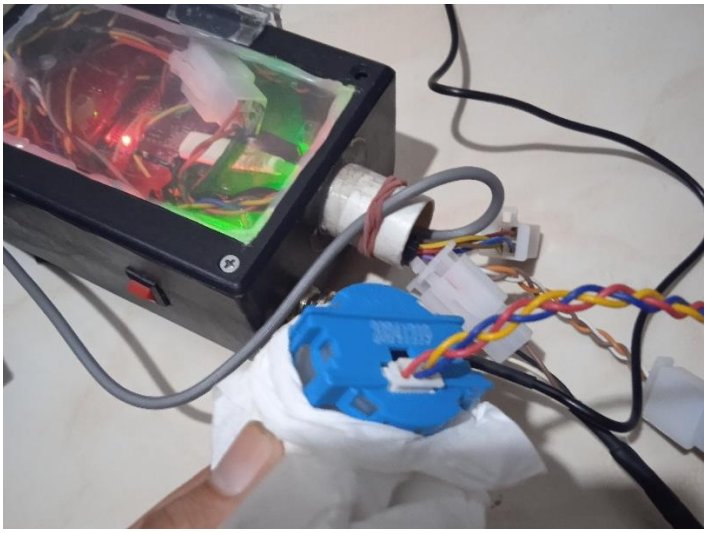
Tabel 4. 4 . Pengujian Sensor Kekeruhan Dengan Cahaya

No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-21 20:50:19	30.1	6.7	30	Ideal	Ideal
2	2026-01-21 20:50:29	30.0	6.8	30	Ideal	Ideal
3	2026-01-21 20:50:39	30.0	6.7	31	Ideal	Ideal
4	2026-01-21 20:50:50	30.0	6.7	31	Ideal	Ideal

Berdasarkan hasil pengujian pada Tabel 4.4, sensor kekeruhan menghasilkan nilai *NTU* yang relatif rendah, yaitu berada pada rentang 30 hingga 31 *NTU*. Nilai tersebut menunjukkan kondisi air yang jernih, sehingga sistem mengklasifikasikan kualitas air sebagai Ideal, baik berdasarkan hasil prediksi machine learning maupun evaluasi *rule-based logic*.

Hasil pengujian ini menunjukkan bahwa sensor kekeruhan mampu mendeteksi kondisi air yang jernih dengan baik ketika cahaya dapat menembus media air secara optimal. Dengan demikian, sensor kekeruhan dapat berfungsi dengan baik dalam mendukung sistem *Smart Aquarium*.

4.2.2.2 Pengujian Dengan Menutup Sensor Kekeruhan (Tidak Ideal)



Gambar 4. 5 Pengujian Sensor Kekeruhan Dengan Menutup Sensor

Pengujian ini dilakukan dengan menutup sensor kekeruhan sehingga cahaya tidak dapat masuk ke sensor. Kondisi ini mensimulasikan air dengan tingkat kekeruhan tinggi, di mana cahaya sulit menembus media air. Tujuan pengujian ini adalah untuk mengetahui respons sensor kekeruhan ketika berada pada kondisi tidak ideal.

Sensor kekeruhan ditutup menggunakan media penutup, kemudian sistem membaca nilai kekeruhan yang dihasilkan dan menyimpan data hasil pengukuran ke dalam *database* MySQL.

Tabel 4. 5 Pengujian Sensor Kekeruhan Tanpa Cahaya

No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-21 20:47:39	30.1	6.9	35	Ideal	Tidak Ideal
2	2026-01-21 20:47:42	30.1	6.9	35	Ideal	Tidak Ideal
3	2026-01-21 20:47:52	30.1	6.8	35	Ideal	Tidak Ideal
4	2026-01-21 20:48:02	30.1	6.8	35	Ideal	Tidak Ideal

Berdasarkan hasil pengujian pada Tabel 4.5, nilai kekeruhan (NTU) yang dihasilkan berada pada angka 35 NTU, yang menunjukkan kondisi air dengan tingkat kekeruhan tinggi. Kondisi ini menyebabkan sistem mengklasifikasikan kualitas air sebagai Tidak Ideal berdasarkan *rule-based logic*.

Namun, hasil prediksi *machine learning* masih menunjukkan status Ideal. Perbedaan hasil ini disebabkan oleh karakteristik model *machine learning* yang mempertimbangkan pola data historis secara keseluruhan, sedangkan *rule-based logic* menggunakan ambang batas kekeruhan yang telah ditentukan. Hal ini menunjukkan bahwa sensor kekeruhan mampu mendeteksi perubahan kondisi air secara signifikan, meskipun interpretasi hasil dapat berbeda tergantung metode evaluasi yang digunakan.

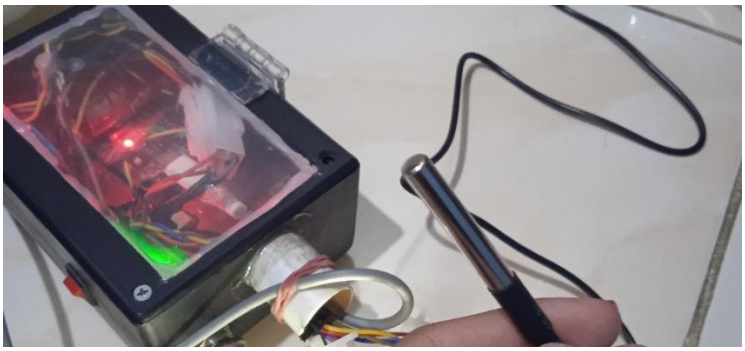
Secara keseluruhan, pengujian ini membuktikan bahwa sensor kekeruhan dapat merespons kondisi tidak ideal dengan baik dan mampu

membedakan tingkat kejernihan air berdasarkan intensitas cahaya yang diterima.

4.2.3 Pengujian Sensor Suhu

Pengujian sensor suhu dilakukan untuk mengetahui kemampuan sensor dalam membaca suhu lingkungan dan kestabilan nilai yang dihasilkan. Sensor suhu digunakan sebagai salah satu parameter dalam sistem *Smart Aquarium* untuk membantu menentukan kondisi kualitas air secara keseluruhan.

4.2.4.1 Pengujian Sensor Suhu Ruangan



Gambar 4. 6 Pengujian Sensor Suhu Dengan Suhu Ruangan

Pengujian ini dilakukan dengan menempatkan sensor suhu pada kondisi suhu ruangan tanpa perlakuan khusus. Sensor dihubungkan ke sistem dan dibiarkan membaca suhu lingkungan secara periodik. Data hasil pembacaan sensor kemudian dikirimkan ke *server backend* dan disimpan ke dalam *database* MySQL.

Tabel 4. 6 Pengujian Sensor Suhu Dengan Suhu Ruangan

No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-21 20:51:51	30	6.8	30	Ideal	Ideal
2	2026-01-21 20:52:01	30	6.8	30	Ideal	Ideal
3	2026-01-21 20:52:11	30	6.8	30	Ideal	Ideal

4	2026-01-21 20:52:22	30	6.7	31	Ideal	Ideal

Berdasarkan hasil pengujian pada Tabel 4.6, sensor suhu mampu membaca suhu ruangan dengan nilai yang stabil, yaitu berada pada suhu sekitar 30,06°C. Perbedaan nilai yang sangat kecil menunjukkan bahwa sensor suhu memiliki kestabilan dan konsistensi pembacaan yang baik.

Nilai suhu yang diperoleh masih berada pada rentang suhu ideal untuk akuarium, sehingga sistem mengklasifikasikan kondisi suhu sebagai Ideal, baik berdasarkan hasil prediksi *machine learning* maupun evaluasi *rule-based logic*.

Dari hasil pengujian ini dapat disimpulkan bahwa sensor suhu berfungsi dengan baik dan layak digunakan sebagai salah satu parameter utama dalam sistem *Smart Aquarium*.

4.2.4.2 Pengujian Sensor Menggunakan Es



Gambar 4. 7 Pengujian Sensor Suhu Menggunakan Es

Pengujian ini dilakukan untuk mengetahui kemampuan sensor suhu dalam membaca perubahan suhu pada kondisi suhu rendah. Pengujian dilakukan dengan menempatkan sensor suhu pada media yang mengandung es, sehingga sensor mengalami penurunan suhu secara signifikan dibandingkan dengan suhu ruangan.

Sensor suhu dicelupkan ke dalam media berisi es dan dibiarkan hingga nilai pembacaan sensor stabil. Data suhu yang terbaca kemudian dikirimkan ke *server backend* dan disimpan ke dalam *database* MySQL.

Tabel 4. 7 Pengujian Sensor Suhu Dengan Es

No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-21 20:59:36	25.8	6.7	27	Ideal	Tidak Ideal
2	2026-01-21 20:59:46	25.6	6.6	27	Ideal	Tidak Ideal
3	2026-01-21 20:59:56	25.4	6.7	27	Ideal	Tidak Ideal
4	2026-01-21 21:00:07	25.3	6.7	27	Ideal	Tidak Ideal
5	2026-01-21 21:00:17	25.1	6.8	27	Ideal	Tidak Ideal

Berdasarkan hasil pengujian pada Tabel 4.7, sensor suhu mampu mendeteksi penurunan suhu secara bertahap dari suhu ruangan menuju suhu yang lebih rendah, yaitu berada pada rentang 25,13°C hingga 25,81°C. Penurunan nilai suhu yang konsisten menunjukkan bahwa sensor suhu memiliki respons yang baik terhadap perubahan suhu lingkungan.

Hasil klasifikasi menunjukkan bahwa model *machine learning* mengklasifikasikan kondisi suhu sebagai Ideal, sedangkan *rule-based logic* mengklasifikasikan kondisi kualitas air sebagai Tidak Ideal. Perbedaan hasil ini disebabkan oleh batas ambang suhu ideal yang digunakan pada *rule-based logic* yang lebih ketat dibandingkan dengan pola yang dipelajari oleh model machine learning.

Dari hasil pengujian ini dapat disimpulkan bahwa sensor suhu mampu membaca perubahan suhu dengan baik dan layak digunakan dalam sistem *Smart Aquarium*.

4.2.4.3 Pengujian Sensor Menggunaan Api



Gambar 4. 8 Pengujian Sensor Suhu Menggunaka Api

Pengujian ini dilakukan untuk mengetahui respons sensor suhu terhadap kondisi suhu tinggi. Pengujian dilakukan dengan memberikan sumber panas berupa api pada bagian sensor suhu dalam waktu singkat. Tujuan pengujian ini adalah untuk melihat kemampuan sensor dalam mendeteksi kenaikan suhu secara cepat dan konsisten.

Sensor suhu didekatkan pada sumber api, kemudian sistem membaca nilai suhu yang dihasilkan dan menyimpan data hasil pembacaan ke dalam *database* MySQL.

Tabel 4. 8 Pengujian Sensor Suhu Dengan Api

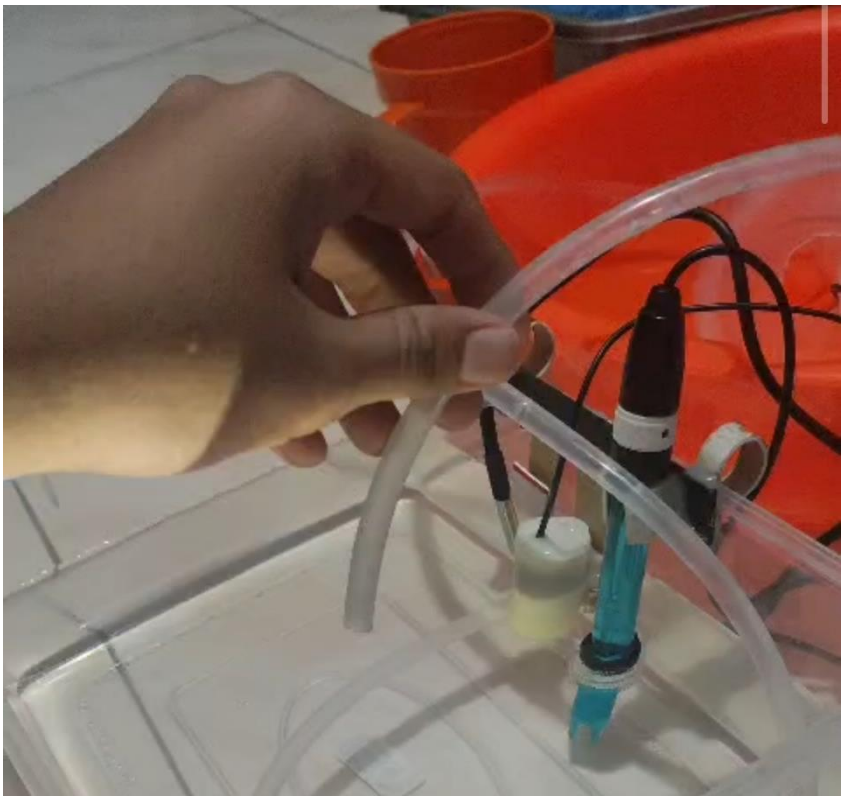
No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-21 20:54:30	44.8	7.3	26	Ideal	Tidak Ideal
2	2026-01-21 20:54:40	42.6	6.8	26	Ideal	Tidak Ideal
3	2026-01-21 20:54:51	41.4	6.8	27	Ideal	Tidak Ideal
4	2026-01-21 20:55:01	40.4	6.8	25	Ideal	Tidak Ideal

Berdasarkan hasil pengujian pada Tabel 4.8, sensor suhu menunjukkan respons yang cepat terhadap sumber panas. Nilai suhu meningkat secara signifikan dari suhu awal sekitar 30°C menjadi lebih dari 40°C ketika sensor didekatkan dengan api. Hal ini menunjukkan bahwa sensor suhu mampu mendeteksi perubahan suhu tinggi dengan baik.

Hasil klasifikasi menunjukkan bahwa model *machine learning* mengklasifikasikan kondisi suhu sebagai Ideal, sedangkan *rule-based logic* mengklasifikasikan kondisi kualitas air sebagai Tidak Ideal. Perbedaan hasil ini disebabkan oleh ambang batas suhu maksimum yang digunakan pada *rule-based logic,* di mana suhu tinggi dianggap tidak ideal untuk kondisi *aquarium*.

Dari hasil pengujian ini dapat disimpulkan bahwa sensor suhu memiliki respons yang baik terhadap perubahan suhu ekstrem dan dapat digunakan untuk mendeteksi kondisi suhu yang berpotensi membahayakan kualitas air akuarium.

4.2.4 Pengujian Keseluruhan Alat



Gambar 4. 9 Pengujian Keseluruhan ALat

Pengujian keseluruhan alat dilakukan untuk mengetahui kinerja sistem *Smart Aquarium* secara terpadu setelah seluruh sensor terpasang dan sistem berjalan secara penuh. Pengujian ini bertujuan untuk memastikan bahwa seluruh komponen, yaitu sensor pH, sensor suhu, dan sensor kekeruhan, dapat bekerja secara bersamaan dalam membaca kondisi air, mengirimkan data ke server, serta menghasilkan output klasifikasi dan prediksi yang sesuai.

Pada pengujian ini, seluruh sensor dipasang pada media air dan sistem dijalankan dalam dua kondisi, yaitu kondisi kualitas air Ideal dan Tidak Ideal.

Data hasil pengujian dikirimkan oleh ESP32 ke *server backend* berbasis *Flask* dan disimpan ke dalam *database* MySQL.

Tabel 4. 9 Pengujian Keseluruhan Alat

No	Waktu Pengujian	Suhu Terbaca	pH	NTU	Status Prediksi	Status Rule
1	2026-01-19 09:45:15	27.1	8.2	26	Ideal	Ideal
2	2026-01-19 09:45:25	28	8.2	27	Ideal	Ideal
3	2026-01-19 09:45:35	28.4	8.3	25	Ideal	Ideal
4	2026-01-19 09:45:46	28.8	8.2	25	Ideal	Ideal
5	2026-01-19 09:45:56	29	8.3	25	Ideal	Ideal
6	2026-01-19 09:17:30	36.6	4.0	32	Tidak Ideal	Tidak Ideal
7	2026-01-19 09:17:41	36.6	6.2	32	Tidak Ideal	Tidak Ideal
8	2026-01-19 09:17:51	36.6	4	33	Tidak Ideal	Tidak Ideal
9	2026-01-19 09:18:01	36.6	4.5	31	Tidak Ideal	Tidak Ideal
10	2026-01-19 09:18:11	36.6	4.4	32	Tidak Ideal	Tidak Ideal

Berdasarkan hasil pengujian keseluruhan alat, sistem *Smart Aquarium* mampu membaca data dari seluruh sensor secara bersamaan dan mengirimkan data tersebut ke *server backend* tanpa kendala. Pada kondisi kualitas air ideal, sistem secara konsisten mengklasifikasikan kondisi air sebagai Ideal, baik berdasarkan hasil prediksi machine learning maupun *rule-based logic*.

Sebaliknya, pada kondisi kualitas air tidak ideal, sistem mampu mendeteksi perubahan nilai pH, suhu, dan kekeruhan yang berada di luar batas normal, sehingga menghasilkan status Tidak Ideal. Konsistensi hasil antara data

sensor, hasil prediksi, dan *rule-based logic* menunjukkan bahwa sistem bekerja dengan baik secara terpadu.

Dari hasil pengujian ini dapat disimpulkan bahwa sistem *Smart Aquarium* telah berfungsi sesuai dengan perancangan dan mampu melakukan monitoring kualitas air secara menyeluruh.

4.3 Pengujian Prediksi Hari Penggantian Air

Pengujian prediksi hari penggantian air dilakukan untuk mengetahui kemampuan sistem dalam memberikan estimasi waktu optimal penggantian air *aquarium*.

4.3.1 Metode Prediksi

Metode prediksi hari penggantian air menggunakan algoritma *Gradient Boosting* yang memanfaatkan data historis sensor pH, suhu air, dan kekeruhan air. Model prediksi dilatih menggunakan dataset sehingga mampu mengenali pola perubahan kualitas air dari waktu ke waktu.

Hasil prediksi berupa estimasi jumlah hari sebelum air akuarium perlu diganti agar kualitas air tetap terjaga.

4.3.2 Hasil Prediksi

Hasil pengujian menunjukkan bahwa sistem mampu memberikan prediksi hari penggantian air secara konsisten berdasarkan perubahan nilai sensor. Prediksi yang dihasilkan membantu pengguna dalam menentukan jadwal penggantian air secara lebih efektif dan terencana.

Dengan adanya fitur prediksi ini, pengguna tidak perlu lagi mengandalkan perkiraan manual, sehingga perawatan akuarium menjadi lebih efisien.

4.4 Visualisasi Monitoring Menggunakan *Node-RED*

Visualisasi monitoring dilakukan untuk menampilkan data hasil pemantauan dan pemrosesan sistem *Smart Aquarium* dalam bentuk *dashboard*.

4.4.1 Implementasi *Dashboard*

Dashboard monitoring dikembangkan menggunakan *Node-RED* yang terhubung langsung dengan database MySQL. *Node-RED* mengambil data sensor, hasil klasifikasi kualitas air, serta hasil prediksi hari penggantian air dari *database* dan menampilkannya dalam bentuk indikator visual.

Dashboard dirancang agar mudah dipahami oleh pengguna dan dapat diakses melalui *web browser*.

4.4.2 Hasil Visualisasi

Hasil visualisasi menunjukkan bahwa dashboard *Node-RED* mampu menampilkan data sensor dan hasil pemrosesan sistem. Informasi kualitas air dan prediksi penggantian air dapat dipantau dengan jelas, sehingga memudahkan pengguna dalam melakukan perawatan *aquarium*.

4.5 Analisis Sistem

Analisis sistem dilakukan untuk mengevaluasi kinerja sistem *Smart Aquarium* secara keseluruhan.

4.5.1 Analisis Sistem

Berdasarkan hasil pengujian yang telah dilakukan, sistem *Smart Aquarium* mampu melakukan *monitoring* kualitas air, klasifikasi kondisi air, serta prediksi hari penggantian air dengan baik. Integrasi antara ESP32, *server Flask*, *database* MySQL, dan *Node-RED* berjalan secara optimal.

4.5.2 **Keterbatasan Sistem**

Meskipun sistem *Smart Aquarium* telah berjalan sesuai dengan perancangan dan mampu melakukan pengiriman data sensor, klasifikasi kualitas air, serta prediksi hari penggantian air, sistem ini masih memiliki beberapa keterbatasan.

Salah satu keterbatasan yang ditemukan pada tahap pengujian adalah kendala pada proses pemantauan data melalui *serial monitor* ketika perangkat ESP32 dihubungkan ke laptop. Pada kondisi tersebut, data sensor tidak ditampilkan dengan benar pada serial monitor dan muncul pesan *error* secara berulang. Kendala ini diduga disebabkan oleh keterbatasan proses komunikasi *serial* ketika ESP32 melakukan pembacaan sensor dan pengiriman data ke *server* secara bersamaan.

Keterbatasan ini menyebabkan proses *debugging* dan pemantauan data sensor secara langsung melalui *serial monitor* menjadi kurang optimal. Namun demikian, kendala tersebut tidak mempengaruhi fungsi utama sistem, karena data sensor tetap berhasil dikirimkan ke *server Flask*, diproses menggunakan metode *machine learning*, disimpan ke dalam *database* MySQL, serta ditampilkan pada *dashboard monitoring Node-RED*.

Selain itu, sistem masih bergantung pada kestabilan koneksi jaringan untuk proses pengiriman data dari ESP32 ke *server*. Apabila terjadi gangguan jaringan, proses pengiriman data dan pembaruan informasi pada *dashboard* dapat mengalami keterlambatan.

Keterbatasan lain pada sistem ini adalah jumlah data historis yang digunakan dalam proses pelatihan model *machine learning* masih terbatas,

sehingga akurasi hasil klasifikasi dan prediksi dapat ditingkatkan dengan penambahan data pada penelitian selanjutnya.

4.5.3 **Pembahasan Hasil**

Hasil penelitian menunjukkan bahwa penerapan teknologi *Internet of Things* dan *Machine Learning* pada sistem *Smart Aquarium* dapat membantu pengguna dalam memantau dan merawat *aquarium* secara otomatis dan efisien.

4.6 Ringkasan BAB

Bab ini membahas hasil implementasi dan pengujian sistem *Smart Aquarium.* Hasil pengujian menunjukkan bahwa sistem mampu mengirimkan data sensor, melakukan klasifikasi kualitas air, memprediksi hari penggantian air, serta menampilkan hasil *monitoring* melalui *dashboard Node-RED*. Sistem ini diharapkan dapat menjadi solusi dalam perawatan *aquarium* yang lebih efektif dan berbasis teknologi.

BAB V

KESIMPULAN & SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem *Smart Aquarium* berbasis *Internet of Things* (IoT) berhasil dibangun dan berfungsi sesuai dengan tujuan penelitian. Sistem ini memanfaatkan *mikrokontroler* ESP32 sebagai perangkat utama yang terhubung dengan sensor pH, sensor suhu, dan sensor kekeruhan untuk melakukan akuisisi data kualitas air *aquarium* secara berkala.

Proses pengiriman data sensor dari ESP32 ke *server backend* berbasis *Flask* dapat berjalan dengan baik melalui protokol HTTP dalam format JSON. Data yang diterima oleh *server* selanjutnya diproses dan disimpan ke dalam *database* MySQL secara otomatis dan terstruktur. *Backend Flask* berhasil mengintegrasikan proses pengolahan data sensor, evaluasi kualitas air menggunakan pendekatan *rule-based* dan prediksi hari penggantian air menggunakan algoritma Gradient Boosting

Hasil pengujian menunjukkan bahwa sensor pH, sensor suhu, dan sensor kekeruhan mampu membaca perubahan kondisi air pada berbagai skenario pengujian, baik pada kondisi ideal maupun tidak ideal. Sistem klasifikasi kualitas air mampu memberikan informasi kondisi air secara otomatis berdasarkan parameter pH, suhu, dan kekeruhan, sehingga dapat membantu pengguna dalam memantau kualitas air *aquarium* tanpa perlu melakukan pengukuran secara manual.

Selain itu, fitur prediksi hari penggantian air mampu memberikan estimasi waktu penggantian air yang dapat digunakan sebagai acuan dalam perawatan

aquarium agar kualitas air tetap terjaga. Informasi hasil pemantauan dan pemrosesan sistem berhasil ditampilkan melalui *dashboard monitoring* berbasis *Node-RED* dan mudah dipahami oleh pengguna. Secara keseluruhan, sistem *Smart Aquarium* yang dikembangkan mampu menjadi solusi monitoring dan perawatan *aquarium* yang efektif dengan memanfaatkan teknologi *IoT* dan *machine learning*.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa hal yang dapat menjadi saran untuk pengembangan sistem *Smart Aquarium* pada penelitian selanjutnya. Pengumpulan data sensor dalam jumlah yang lebih besar dan dengan variasi kondisi yang lebih beragam disarankan agar proses pelatihan model *machine learning* dapat menghasilkan tingkat akurasi klasifikasi dan prediksi yang lebih optimal.

Pengembangan sistem juga dapat ditingkatkan dengan menambahkan jenis sensor lain, seperti sensor oksigen terlarut atau sensor amonia, sehingga penilaian kualitas air akuarium dapat dilakukan secara lebih komprehensif. Selain itu, fitur visualisasi *monitoring* dapat dikembangkan lebih lanjut dengan menambahkan sistem notifikasi atau peringatan otomatis apabila kondisi kualitas air berada pada status tidak ideal.

Pada penelitian selanjutnya, sistem *Smart Aquarium* juga dapat dikembangkan dengan menambahkan mekanisme kontrol otomatis, seperti pengaturan aerator, penggantian air, atau pemberian pakan otomatis yang terintegrasi dengan hasil klasifikasi dan prediksi sistem. Untuk meningkatkan keandalan sistem, disarankan pula adanya mekanisme komunikasi dan penyimpanan data yang lebih stabil,

terutama ketika terjadi gangguan jaringan, sehingga sistem tetap dapat berjalan dengan optimal.

DAFTAR PUSTAKA

Abdurohman, M., Putrada, A. G., & Deris, M. M. (2022). A Robust Internet of Things-Based Aquarium Control System Using Decision Tree Regression Algorithm. *IEEE Access*, *10*, 56937–56951. https://doi.org/10.1109/ACCESS.2022.3177225

Admin Alatuji. (2024, June 11). *Pengertian Turbidity Sensor dan Cara Kerjanya*. ALAT UJI.

Bagus, I., Peling, A., Made, I., Agus Ariawan, P., Subiksa, G. B., Elektro, J. T., Terapan, S., Rekayasa, T., Lunak, P., & Bali, P. N. (2023). *3520 https://ejournal.sidyanusa.org/index.php/jkdn Deteksi Bahasa Isyarat Menggunakan Tensorflow Lite dan American Sign Language (ASL)* (Vol. 3). https://ejournal.sidyanusa.org/index.php/jkdn

Damayanti, E., Listiana, R., Ahmad Efendi, A., Romi Sihombing, D., Otomasi, T., Elektronika, T., & TEDC Bandung, P. (n.d.). Agung Ahmad Efendi. In *Daniel Romi Sihombing TEDC* (Vol. 17, Issue 2).

ElectronicWings. (2024). *DAC (Konverter Digital ke Analog) ESP32*. ElectronicWings.

Emir Garden. (2023, February 7). *Pengertian Akuarium, Sejarah, Jenis dan Manfaatnya*. Emir Garden.

Galih Agus Saputra. (2022, May 13). *Studi: Akuarium Bermanfaat Pada Kesehatan*. MEDIA INDONESIA.

Ganda Saputra, I., Ronggo Waluyo, J. H., Telukjambe Timur, K., & Karawang, K. (n.d.). *PERANCANGAN WEBSITE BERBASIS INTERNET OF THINGS MENGUNAKAN FRAMEWROK FLASK* (Issue 2). Bulan. www.jurnalteknik@unisla.ac.id/index.php/elektronika

Hasya Akhlaqah Harmie, R. D. D. P. (2024). PERANCANGAN APLIKASI SMART FENCE MENGGUNAKAN VISUAL STUDIO CODE BERBASIS IOT. *Jurnal Inovesi Teknologi Terapan*, *2*.

Ibnu Daqiqi Id. (2021). *Machine Learning : Teori , Studi Kasus dan Implementasi Menggunakan Python* . UR PRESS .

Imran, A., & Rasul, M. (2020). *PENGEMBANGAN TEMPAT SAMPAH PINTAR MENGGUNAKAN ESP32* (Vol. 17, Issue 2).

Kusumah, H., & Pradana, R. A. (n.d.). *PENERAPAN TRAINER INTERFACING MIKROKONTROLER DAN INTERNET OF THINGS BERBASIS ESP32 PADA MATA KULIAH INTERFACING*.

Meika Puspita Sari, , Helmizuryani, Sasua Hustati, Dyckhi Andriani, Pandu Sapta Nugraha,. (2019). *PELATIHAN PEMBUATAN AKUARIUM MINI DAN TEKNIK PEMELIHARAAN IKAN HIAS DI KECAMATAN ALANG-ALANG LEBAR*.

Mulyono, S., Qomaruddin, M., & Syaiful Anwar, M. (2018). Penggunaan Node-RED pada Sistem Monitoring dan Kontrol Green House berbasis Protokol

MQTT. In *Jurnal Transistor Elektro dan Informatika (TRANSISTOR EI* (Vol. 3, Issue 1).

ngodingdata. (2021, October 16). *Tutorial Flask – Web Framework Python*. Ngodingdata.

Nurazizah, E., Ramdhani, M., & Rizal, A. (n.d.). *RANCANG BANGUN TERMOMETER DIGITAL BERBASIS SENSOR DS18B20 UNTUK PENYANDANG TUNANETRA (DESIGN DIGITAL THERMOMETER BASED ON SENSOR DS18B20 FOR BLIND PEOPLE)*.

Pengaruh Temperatur terhadap pH. (2023, February 17). Sumber Aneka Karya Abadi.

Prasetio, A., Hasibuan, M. H., & Sitompul, P. (2021). Simulasi Penerapan Metode Decision Tree (C4.5) Pada Penentuan Status Gizi Balita. *Jurnal Nasional Komputasi Dan Teknologi Informasi*, *4*(3).

Prasetiyo, I. B., Riadi, A. A., & Chamid, A. A. (2021). Perancangan Smart Aquarium Menggunakan Sensor Turbidity Dan Sensor Ultrasonik Pada Akuarium Ikan Air Tawar Berbasis Arduino Uno. *Jurnal Teknologi*, *13*(2), 193–200.

Putra Nasyuli, L., Lubis, I., Marwan Elhanafi, A., author, corresponding, & Teknik Dan Komputer, F. (2023). *Penerapan Model Machine Learning Algoritma Gradient Boosting dan Linear Regression Melakukan Prediksi Harga Kendaraan Bekas Application Of Machine Learning Models and Gradient Boosting Algorithms Doing Linear Regression Vehicle Price*

Prediction Used (Vol. 2023, Issue 2). https://jurnal.unity-academy.sch.id/index.php/jirsi/index

Ramadhan, R. F., & Mukhaiyar, R. (2020). Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. In *JTEIN: Jurnal Teknik Elektro Indonesia* (Vol. 1, Issue 2).

Rosyidah Ilmi, R., Kurniawan, F., Harini, S., & Korespondensi, P. (2023). *PREDIKSI RATING FILM IMDb MENGGUNAKAN DECISION TREE IMDb MOVIE RATING PREDICTION USING DECISION TREE*. *10*(4). https://doi.org/10.25126/jtiik.2023106615

Saputra, P. A., Irawan, S. S., Rahmaddeni, R., Prianto, R., & Hidayat, T. (2025). PREDIKSI DAN ANALISIS POLA PERUBAHAN IKLIM MENGGUNAKAN ALGORITMA GRADIENT BOOSTING. *Jurnal Informatika Dan Teknik Elektro Terapan*, *13*(2). https://doi.org/10.23960/jitet.v13i2.6230

Selay, A., Andgha, G. D., Alfarizi, M. A., Izdhihar, M., Wahyudi, B., Falah, M. N., Khaira, M., & Encep, M. (2022). INTERNET OF THINGS. In *Karimah Tauhid* (Vol. 1).

SKRIPSI RAHMAT DONGORAN terbaru7. (n.d.).

Taufiq Tamam, M., Nugroho Aji, D., & Ahmad Dahlan Dukuhlawuh Purwokerto, J. K. (n.d.). *Perancangan dan Pembuatan Sistem Pengaturan pH dan Suhu Air pada Kolam Ikan*. *5*(1).

Wasito, E., Prahara, T., Nursyahid, A., Anggraeni, S. K., & Miftah Delaili Muntaha, G. (2024). Implementation Of IoT In Nila Fish Cultivation With Bioflock System. In *Journal of Applied Information and Communication Technologies* (Vol. 9, Issue 1). http://smartbioflok.com,

Website, P., Iot, B., Pendataan, P., Tinggi, A. U., Berat, D., Balita, B., & Posyandu, D. (2023). *Based Website Design On Data Collection For Height And Weight Measurement Tools Of Toddlers At Integrated Healthcare Center* (Vol. 9, Issue 5).