

**DESAIN INTERNET OF THINGS DALAM PROTOTIPE SISTEM
MONITORING KUALITAS AIR PADA TAMBAK UDANG**

TUGAS AKHIR



**ADAM LAZUARDI HARIS
NIM : 312210026**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
MALANG
2024**

LEMBAR PENGESAHAN
TUGAS AKHIR

DESAIN INTERNET OF THINGS DALAM PROTOTIPE SISTEM
MONITORING KUALITAS AIR PADA TAMBAK UDANG

Oleh:

ADAM LAZUARDI HARIS

312210026

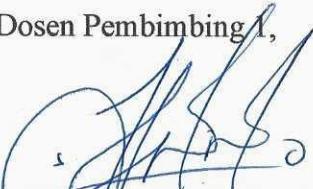
dari:

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS DESAIN DAN TEKNOLOGI UNIVERSITAS MA CHUNG

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan
dan berhak mendapatkan gelar Sarjana Komputer (S.Kom)

Dosen Pembimbing 1,


Mochamad Subianto, S.Kom.,
M.Cs.
NIP. 20100002

Dosen Pembimbing 2,


Paulus Lucky Tirma Irawan,
S.Kom., MT.
NIP. 20100005

Dekan Fakultas Sains dan Teknologi,


Prof. Dr. Eng. Romy Budhi Widodo
NIP. 20070035

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Desain Internet Of Things Dalam Prototipe Sistem Monitoring Kualitas Air Pada Tambak Udang” adalah benar benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, 14 Agustus 2024



Adam Lazuardi Haris
NIM. 312210026

DESAIN INTERNET OF THINGS DALAM PROTOTIPE SISTEM MONITORING KUALITAS AIR PADA TAMBAK UDANG

Adam Lazuardi Haris, Mochamad Subianto, Paulus Lucky Tirma Irawan

Abstrak

Indonesia memiliki potensi budidaya udang sebesar 3 juta hektar, namun hanya 300 ribu hektar yang dimanfaatkan dengan produksi rata-rata satu ton per hektar dan mayoritas budidaya masih tradisional dan memerlukan modernisasi. Tantangan utama dari permasalahan tersebut termasuk kurangnya teknologi modern dan sarana produksi memerlukan sistem pemantauan berbasis *Internet of Things* (IoT) untuk meningkatkan efisiensi dan akurasi dalam mengelola kualitas air tambak udang. Pembuatan prototipe alat pengukur kualitas air menggunakan sensor suhu, pH, kekeruhan, amonia, dan *Total Dissolve Solids* (TDS) diintegrasikan dengan IoT untuk akses data *real-time*, bertujuan meningkatkan pengelolaan tambak udang. Pengujian prototipe alat ukur kualitas air berbasis IoT menunjukkan hasil yang positif dengan beberapa kalibrasi yang diperlukan. Sensor suhu, pH, amonia, dan salinitas menunjukkan hasil akurat dalam kondisi air jernih: suhu rata-rata 25,61°C (standar deviasi 0,019°C), pH 8,158 (standar deviasi 0,31), amonia 0,013 ppm (standar deviasi 0,00065), dan salinitas 19,32 ppt (standar deviasi 0,14). Sensor kekeruhan memerlukan kalibrasi tambahan dengan nilai -2,9 NTU (standar deviasi 1,44). Dalam kondisi air terkontaminasi tanah, terdapat variasi signifikan: suhu 25,57°C (standar deviasi 0,062°C), pH 7,43 (standar deviasi 0,44), amonia 0,0233 ppm (standar deviasi 0,00384), salinitas 19,74 ppt (standar deviasi 0,15), dan kekeruhan 25,7 NTU (standar deviasi 2,31). Kecepatan respon API untuk mengirim data ke aplikasi web rata-rata 522,06 ms, dengan standar deviasi 137,29 ms, sehingga memerlukan optimasi lebih lanjut. Survei terhadap 7 responden profesional menunjukkan penerimaan positif terhadap prototipe dengan antarmuka *web* yang intuitif dan data yang akurat. Sebanyak 85,7% responden menemukan data akurat dan 85,8% menganggap akses data *real-time* sangat penting.

Kata Kunci : Amonia, ESP32, *Internet of Things*, kekeruhan, kualitas air, Laravel, pH, salinitas, suhu, udang

DESIGN OF INTERNET OF THINGS IN PROTOTYPE WATER QUALITY MONITORING SYSTEM FOR SHRIMP PONDS

Adam Lazuardi Haris, Mochamad Subianto, Paulus Lucky Tirma Irawan

Abstract

Indonesia has a shrimp farming potential of 3 million hectares, but only 300,000 hectares are utilized, with an average production of one ton per hectare. The majority of farming remains traditional and requires modernization. The main challenges include the lack of modern technology and production facilities, necessitating an Internet of Things (IoT)-based monitoring system to improve efficiency and accuracy in managing shrimp pond water quality. A prototype water quality measuring device was developed using temperature, pH, turbidity, ammonia, and total dissolve solids sensors integrated with IoT for real-time data access, aiming to enhance shrimp pond management. Testing of the IoT-based water quality measuring prototype showed positive results, with some calibrations needed. Temperature, pH, ammonia, and salinity sensors demonstrated accurate results in clear water conditions: average temperature of 25.61°C (standard deviation 0.019°C), pH 8.158 (standard deviation 0.31), ammonia 0.013 ppm (standard deviation 0.00065), and salinity 19.32 ppt (standard deviation 0.14). The turbidity sensor requires additional calibration with a value of -2.9 NTU (standard deviation 1.44). In water contaminated with soil, there were significant variations: temperature 25.57°C (standard deviation 0.062°C), pH 7.43 (standard deviation 0.44), ammonia 0.0233 ppm (standard deviation 0.00384), salinity 19.74 ppt (standard deviation 0.15), and turbidity 25.7 NTU (standard deviation 2.31). The API response speed to send data to the web application averaged 522.06 ms, with a standard deviation of 137.29 ms, requiring further optimization. A survey of 7 professional respondents showed positive acceptance of the prototype, with an intuitive web interface and accurate data. A total of 85.7% of respondents found the data accurate, and 85.8% considered real-time data access very important.

Keywords : Ammonia, ES32, Internet of Things, laravel, pH, salinity, shrimp, temperature, turbidity, water quality

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena atas rahmat dan penyertaan-Nya sehingga Tugas Akhir ini dapat berjalan dengan baik. Laporan ini disusun untuk memenuhi salah satu persyaratan dalam menyelesaikan perkuliahan bagi para Mahasiswa dari Prodi Teknik Informatika Universitas Ma Chung Malang. Tugas Akhir ini merupakan salah satu upaya dalam menjalin kerja sama yang baik dengan pihak Universitas Ma Chung Malang. Kami harap Tugas Akhir ini akan memberi banyak manfaat bagi para mahasiswa maupun bagi pembaca.

Atas dukungan moral dan materil yang diberikan dalam pembuatan laporan ini, penulis ingin mengucapkan terima kasih kepada:

1. Orang tua yang saya sayangi dan kasih yang telah mendukung penyelesaian Tugas Akhir ini.
2. Bapak Mochamad Subianto, S.Kom., M.Cs. selaku dosen pembimbing I yang membantu dalam pemberian arahan dalam perakitan alat dengan teliti dan penulisan laporan tugas akhir yang detil.
3. Bapak Paulus Lucky Tirma Irawan, S.Kom., MT. selaku dosen pembimbing II dan dosen penasehat akademik yang membantu dalam pemberian arahan untuk menulis laporan tugas akhir dengan teliti sesuai dengan panduan.
4. Rekan-rekan mahasiswa maupun alumni prodi Teknik Informatika Universitas Ma Chung yang sudah meluangkan waktu untuk membantu dalam penelitian ini.

Susunan laporan Tugas Akhir ini sudah dibuat dengan sebaik-baiknya, namun tentu masih banyak kekurangan. Oleh karena itu jika ada kritik atau saran apapun yang sifatnya membangun bagi penulis, dengan senang hati akan penulis terima.

Malang, 15 Agustus 2024

Adam Lazuardi Haris

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	x
DAFTAR PERSAMAAN	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	2
1.4 Rumusan Masalah	3
1.5 Tujuan	3
1.6 Manfaat	3
1.7 Sistematika Penulisan	3
1.8 Luaran	4
BAB II TINJAUAN PUSTAKA	5
2.1 Kualitas Air Tambak	5
2.1.1 Suhu	5
2.1.2 pH Air	5
2.1.3 Tingkat Keasaman Amonia	6
2.1.4 Derajat Kekaruan Air	6
2.1.5 Salinitas	7
2.2 Mikrokontroler	7
2.2.1 ESP32	7
2.3 Sensor	8
2.3.1 Sensor Suhu	9
2.3.2 Sensor pH	10
2.3.3 Sensor Gas	11
2.3.4 Sensor Kekaruan	13
2.3.5 Sensor TDS	15

2.4 Internet Of Things (IOT)	16
2.5 MySQL	17
2.6 Application Programming Interface (API)	17
2.7 Postman	18
2.8 Framework	18
2.9 Laravel	18
2.10 Penelitian Terdahulu	19
BAB III ANALISIS DAN PERANCANGAN SISTEM	20
3.1 Tahapan Penelitian	20
3.2 Identifikasi Masalah	21
3.3 Analisis Kebutuhan	21
3.4 Perancangan Desain Sistem	22
3.4.1 Rangkaian Prototipe Alat	24
3.4.2 Rancangan Peletakan Alat	25
3.4.3 Arduino IDE	26
3.4.4 Data Kualitas Air	27
3.4.5 Basis Data <i>MySQL</i>	27
3.4.6 Framework Laravel	28
3.4.7 Tampilan Halaman <i>Web</i>	28
3.5 Pengujian Prototipe	30
3.5.1 Pengujian Pembanding	30
3.5.2 Pengujian Kelayakan Perangkat	31
3.5.3 Pengujian Menggunakan Air Laut Tiruan	32
BAB IV HASIL DAN PEMBAHASAN	33
4.1 Alat / Komponen	33
4.2 Kalibrasi Sensor	34
4.2.1 Kalibrasi Sensor pH	34
4.2.2 Kalibrasi Sensor Kekeruhan	39
4.2.3 Kalibrasi Sensor TDS	42
4.2.4 Kalibrasi Sensor Gas	44
4.2.5 Kalibrasi Sensor Suhu	48
4.3 Perangkaian Alat	51

4.3.1 Rangkaian Sensor	51
4.3.2 Kemasan Alat	52
4.4 Pembuatan Aplikasi <i>Web</i>	53
4.4.1 Pembuatan <i>Database</i>	54
4.4.2 Pembuatan Aplikasi <i>Web</i>	56
4.4.3 Integrasi API Untuk <i>Database</i>	61
4.4.4 Akses Daring Apikasi <i>Web</i>	63
4.5 Penggunaan Alat	64
4.6 Pengujian	65
4.6.1 Pengujian <i>Transfer</i> Data Sensor	65
4.6.2 Pengujian Simulasi Pada Air Tambak Tiruan	66
4.6.3 Pengujian Aplikasi <i>Web</i>	71
BAB V PENUTUP	74
5.1 Simpulan	74
5.2 Saran	74
DAFTAR PUSTAKA	75

DAFTAR GAMBAR

Gambar 2. 1 Struktur Pin pada ESP32 (circuitstate.com).	8
Gambar 2. 2 Sensor Suhu DS18B20 (randomnerdtutorials.com).	9
Gambar 2. 3 Sensor pH 4502C (guguslangit.wordpress.com).	10
Gambar 2. 4 Sensor Gas MQ-135 (quartzcomponents.com).	12
Gambar 2. 5 Grafik Rasio Sensor (components101.com).	13
Gambar 2. 6 Sensor Kekaruan SEN-0189 (bukalapak.com).	14
Gambar 2. 7 Grafik Nilai Sensor Kekaruan (wiki.dfrobot.com).	15
Gambar 2. 8 Sensor TDS KS0429. (www.techmasterkw.com)	15
Gambar 3. 1 Bagan Tahapan Perancangan Prototipe Alat.	20
Gambar 3. 2 Use-Case Diagram Untuk Pengguna.	22
Gambar 3. 3 Desain Sistem IoT pada Tambak Udang.	23
Gambar 3. 4 Rangkaian Prototipe Sistem Pada ESP32.	24
Gambar 3. 5 Rancangan Peletakan Prototipe Alat.	26
Gambar 3. 6 Skema Database Pada Sistem.	28
Gambar 3. 7 Rancangan Tampilan Utama Halaman Web.	29
Gambar 3. 8 Halaman Web Menampilkan Kumpulan ESP32.	30
Gambar 4. 1 Tampilan Awal Arduino IDE.	34
Gambar 4. 2 Komponen Untuk Sensor pH 4502C.	35
Gambar 4. 3 Cairan Buffer Solution Untuk Menetralkan Probe Sensor pH.	35
Gambar 4. 4 Pembacaan sensor pH untuk cairan pH 7 (Kiri) dan cairan pH 4 (Kanan)	36
Gambar 4. 5 Hasil Kalibrasi dari cairan pH 4 (Kiri) dan cairan pH 7 (Kanan)	37
Gambar 4. 6 LAQUA PH2000 (Kiri) dan Prototipe Alat Sensor pH (Kanan)	38
Gambar 4. 8 Komponen Sensor Kekaruan SEN-0189.	39
Gambar 4. 9 Pengujian Air Bersih (Kiri) dan Pengujian Air Susu (Kanan).	40
Gambar 4. 10 Fungsi Pembacaan Sensor Keruh.	40
Gambar 4. 11 Hasil Air Bersih (Kiri) dan Hasil Simulasi Air Keruh (Kanan).	41
Gambar 4. 12 TB 211 IR (Kiri) dan Sensor Kekaruan SEN-0189 (Kanan).	41
Gambar 4. 14 Komponen Sensor TDS KS0429.	42
Gambar 4. 15 Probe Sensor TDS Dicelupkan Ke Air Garam.	43
Gambar 4. 16 Hasil Dari Pembacaan Air Garam Menggunakan Sensor TDS.	44

Gambar 4. 17 Sensor Gas MQ-135.	44
Gambar 4. 18 Corong Tambahan Sensor Gas.	45
Gambar 4. 19 Sensor Gas Pada Udara Terbuka (Kiri) dan Dicelupkan Pada Air Bersih (Kanan).	46
Gambar 4. 20 Grafik Perhitungan NH4 (Amonia).	47
Gambar 4. 21 Hasil Pengukuran Amonia Udara Terbuka (Kiri) dan Permukaan Air Bersih (Kanan).	48
Gambar 4. 22 Sensor Suhu DS18B20 (randomnerdtutorials.com).	48
Gambar 4. 23 Kalibrasi Sensor Suhu Menggunakan Air Bersih.	49
Gambar 4. 24 Hasil Pengukuran Suhu Pada Air Bersih.	50
Gambar 4. 27 Rangkaian Prototipe Alat.	51
Gambar 4. 28 Sensor Dikemas Dalam Wadah Apung.	53
Gambar 4. 29 Kondisi Wadah Mengapung di Air.	53
Gambar 4. 30 Struktur Database Untuk Perangkat Sensor.	55
Gambar 4. 31 Tampilan Awal Halaman Web.	57
Gambar 4. 32 Fungsi Menampilkan Data Sensor Bentuk Numerik.	58
Gambar 4. 33 Fungsi Untuk Menampilkan Data Dalam Interval 2 Detik.	58
Gambar 4. 34 Halaman Web Tampil Semua Device (Kiri) dan Halaman Web Setelah Memilih Device Yang Ingin Ditampilkan (Kanan).	59
Gambar 4. 35 Tampilan Awal Halaman Grafik.	59
Gambar 4. 36 Fungsi Menampilkan Data Bentuk Grafik.	60
Gambar 4. 37 Fungsi Untuk Mengirim dan Menerima Data dari Database.	61
Gambar 4. 39 Percobaan API Menggunakan Postman.	62
Gambar 4. 38 Fungsi Untuk Memanggil Data API.	62
Gambar 4. 40 Tampilan Web Menerima Data Realtime Dari Postman.	63
Gambar 4. 41 Tampilan Dashboard DomaiNesia.	64
Gambar 4. 42 Fungsi Untuk Menambah Jaringan Internet Baru.	64
Gambar 4. 43 Fungsi Registrasi Alat Baru.	65
Gambar 4. 44 Fungsi Network Pada Developer Tools.	65
Gambar 4. 45 Garam Dicampur Dengan Air Jernih.	67
Gambar 4. 46 Alat Mengapung Di Air Tambak Tiruan Kondisi Jernih.	67
Gambar 4. 47 Tanah Dicampur Dengan Air Garam Bersih.	69

DAFTAR TABEL

Tabel 2. 1 Penyambungan Sensor Suhu Pada Mikrokontroler.	9
Tabel 2. 2 Penyambungan Sensor pH Pada Mikrokontroler.	11
Tabel 2. 3 Penyambungan Sensor Gas Pada Mikrokontroler.	12
Tabel 2. 4 Penyambungan Sensor Kekeruhan Pada Mikrokontroler.	14
Tabel 2. 5 Penyambungan Sensor TDS Pada Mikrokontroler.	16
Tabel 3. 1 Paramter Kualitas Air Tambak.	30
Tabel 4. 1 Perbandingan Data pH Dengan Interval Waktu 2 Detik (2000 Milliseconds).	38
Tabel 4. 2 Perbandingan Data Keruh Dengan Interval Waktu 2 Detik (2000 Milliseconds).	41
Tabel 4. 3 Perbandingan Data Suhu Dengan Interval Waktu 2 Detik (2000 Milliseconds).	50
Tabel 4. 4 Contoh Isi Tabel <i>device</i> .	55
Tabel 4. 5 Contoh Isi Tabel <i>sensor</i> .	56
Tabel 4. 6 Hasil Pengujian Kecepatan Transfer Data Pada Web Dalam Millisecond.	66
Tabel 4. 7 Hasil Pengujian Air Tambak Tiruan Kondisi Jernih.	68
Tabel 4. 8 Hasil Pengujian Air Tambak Tiruan Kondisi Kotor.	70
Tabel 4. 9 Profil Responden.	71
Tabel 4. 10 Hasil Survei Kelayakan Prototipe.	72

DAFTAR PERSAMAAN

2- 1 Persamaan Perhitungan Tegangan Sensor pH.	11
2- 2 Persamaan Perhitungan Nilai Sensor pH.	11
2- 3 Persamaan Perhitungan Resistansi Sensor.	12
2- 4 Persamaan Perhitungan RO.	13
2- 5 Persamaan Perhitungan ratio.	13
2- 6 Persamaan Perhitungan PPM.	13
2- 7 Persamaan Mencari Nilai m.	13
2- 8 Persamaan Mencari Nilai b.	13
2- 9 Persamaan Perhitungan Tegangan Sensor Kekeruhan.	14
4- 1 Persamaan Perhitungan pH Step.	37
4- 2 Persamaan Perhitungan pH Netral.	37
4- 3 Persamaan Perhitungan Tegangan Sensor Amonia.	46
4- 4 Persamaan Perhitungan Resistansi Sensor.	46
4- 5 Persamaan Perhitungan RO.	46
4- 6 Persamaan Perhitungan ratio.	47
4- 7 Persamaan Mencari Nilai m.	47
4- 8 Persamaan Mencari Nilai b.	47
4- 9 Persamaan Perhitungan PPM.	48

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia, sebagai negara maritim, memiliki garis pantai sepanjang 95.185 km dengan potensi lahan budidaya udang mencapai 3 juta hektar. Saat ini, luas tambak udang di Indonesia mencapai 300 ribu hektar, dengan produksi rata-rata satu ton udang per hektar yang dikelola oleh para nelayan. Namun, beberapa faktor menghambat optimalisasi industri perudangan, seperti kesulitan dalam memperoleh sarana produksi dan kurangnya penerapan teknologi modern (Nanda Rizky, Hari Saputra, Supendar, & Budiawan, 2020).

Penelitian menunjukkan bahwa kepadatan tebar yang tinggi dapat mempengaruhi performa komposisi tubuh udang, serta kualitas air dalam sistem bioflok atau mengandalkan mikroorganisme sebagai sumber produktivitasnya dalam mencegah beragam jenis penyakit pada udang. Penggunaan probiotik dan kontrol kualitas air yang ketat terbukti meningkatkan pertumbuhan dan mengurangi bahan organik dalam tambak (Inayah, Musa, & Arfiati, 2023).

Untuk memajukan industri ini, diperlukan sinergi antara pemerintah, sektor swasta, dan masyarakat diharapkan dapat bekerja sama membangun Indonesia yang lebih baik. Sekitar 80% entitas budidaya perairan di Indonesia masih menggunakan praktik tradisional atau ekstensif. Menurut data Badan Pusat Statistik (BPS), supaya budidaya perairan tetap kompetitif, diperlukan adopsi peralatan dan teknik produksi yang lebih modern. Namun, tantangan utama adalah mayoritas pelaku budidaya perairan merupakan industri rumah tangga dengan modal yang minim dan keterampilan yang kurang memadai untuk memodernisasi teknik budidaya (Nanda Rizky, Hari Saputra, Supendar, & Budiawan, 2020). Proses pemantauan yang dilakukan secara periodik tersebut cenderung tidak praktis, memerlukan biaya tenaga kerja yang tinggi, dan memiliki tingkat kesalahan manusia yang tinggi. Selain itu, keterbatasan dalam penyimpanan data yang besar membuatnya tidak dapat digunakan untuk memprediksi karakteristik kualitas air pada tambak udang. Untuk itu, diperlukan sebuah sistem pemantauan kualitas air menggunakan sebuah sistem *Internet of Things* (IoT). IoT adalah konsep yang mengoptimalkan

konektivitas internet dengan berbagai perangkat di sekitar kita. Istilah ini pertama kali diperkenalkan oleh Kevin Ashton pada tahun 1999. Untuk dianggap sebagai IoT, sebuah sistem harus memiliki sensor, konektivitas (baik melalui jaringan lokal maupun internet), dan media penyimpanan. Manusia memainkan peran integral dalam sistem IoT, namun sistem ini tetap dapat beroperasi dengan baik meskipun tanpa interaksi manusia (Ansarullah & Nurwasito, 2022).

1.2 Identifikasi Masalah

Budidaya udang di Indonesia berkembang pesat namun menghadapi tantangan signifikan terkait fluktuasi kualitas air, kepadatan stok, dan perubahan lingkungan yang mempengaruhi pertumbuhan dan kelangsungan hidup udang. Penelitian menunjukkan bahwa kepadatan tebar yang tinggi dan kualitas air yang buruk dapat menghambat pertumbuhan komposisi tubuh udang. Untuk permasalahan tersebut, diperlukan kontrol atau pemantauan kualitas air yang ketat yang dapat diakses dengan mudah.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini yaitu:

1. Menggunakan mikrokontroller ESP32.
2. Menggunakan tampilan *web* sebagai hasil luaran dari sistem.
3. Hanya mengulas kualitas air dalam lingkup budidaya.
4. Menggunakan sensor-sensor untuk keperluan studi pada kadar suhu air, pH air, kekeruhan air, kandungan amonia, dan *Total Dissolve Solids* (TDS) pada air kolam tambak tiruan.
5. Tidak mengulas kondisi eksternal seperti faktor cuaca.
6. Tidak melakukan pengukuran pada budidaya tambak yang sesungguhnya.
7. Kekurangan alat-alat pembanding untuk sensor TDS dan sensor amonia untuk acuan pengukuran yang akurat.
8. Mengabaikan tindakan lanjut atau aksi di lapangan berupa peringatan jika terdapat nilai pengukuran yang melebihi ambang batas bawah atau atas.

1.4 Rumusan Masalah

Rumusan masalah yang didapat adalah bagaimana mengetahui kualitas air dalam budidaya tambak udang dan membuat sebuah *platform* atau sistem secara efektif maupun efisien dalam pengumpulan data kualitas air, sebagai luaran untuk pemantauan kualitas air tambak sesuai standar.

1.5 Tujuan

Tujuan dari pelaksanaan tugas akhir ini adalah untuk pengembangan prototipe alat dari sensor-sensor yang beredar dalam lingkup studi atau pengembangan untuk kualitas air yang berupa sensor suhu, sensor pH, sensor kekeruhan, sensor amonia, dan sensor *Total Dissolve Solids* (TDS) dalam menganalisis kualitas air menggunakan *Internet of Things* (IOT) sebagai media dengan menjadikan data pengukuran dapat diakses dengan mudah secara daring dan *realtime*.

1.6 Manfaat

Manfaat yang diperoleh dari penelitian adalah sebagai berikut:

- a. Bagi Human-Machine Interaction Research Center yaitu dapat menjadi penelitian untuk tugas akhir mahasiswa.
- b. Bagi mahasiswa, yaitu dapat memahami cara kerja dan dapat mengoperasikan.
- c. Bagi pengusaha atau pembudidaya, yaitu dapat mengimplementasikan alur pengumpulan sampel dan kemudahan akses secara cepat untuk memberikan informasi data air dalam pengelolaan kualitas air tambak.

1.7 Sistematika Penulisan

Penulisan laporan tugas akhir ini ditulis dengan sistematika penulisan sebagai berikut:

1. Bab I Pendahuluan

Berisi latar belakang, permasalahan, dan tujuan penelitian yang berkaitan dengan perancangan perangkat pemantauan kualitas air dari faktor-faktor yang mempengaruhi kualitas air tambak udang.

2. Bab II Tinjauan Pustaka

Berisi teori-teori terkait yang digunakan untuk mendukung penelitian kualitas air tambak udang menggunakan parameter suhu air, pH air, kekeruhan air, kandungan amonia dan tingkat padatan terlarut berupa salinitas, dengan menggunakan perangkat seperti ESP32, sensor suhu, sensor pH, sensor amonia, sensor TDS (*Total Dissolved Solids*), dan sensor kekeruhan, serta metode luarannya berupa sistem berbasis *Internet of Things* (IOT).

3. Bab III Analisis dan Perancangan Sistem

Berisi analisis dan alur perancangan sistem yang menjelaskan tahapan-tahapan dalam perancangan dengan rinci beserta instrumentasi penilaian yang digunakan seperti, tahapan penelitian, analisis kebutuhan, desain sistem, prototipe alat, translasi data, pengolahan data, rancangan alur penggunaan sistem, serta rancangan pengujian.

4. Bab IV Hasil dan Pembahasan

Berisi pembahasan dan hasil dari pembuatan serta pengujian prototipe. Pengujian dilakukan untuk mengetahui tingkat keberhasilan prototipe alat dalam mengetahui tingkat suhu air, kandungan pH air, kekeruhan air, tingkat padatan terlarut dan kandungan amonia dan memberikan hasil berupa luaran dalam bentuk aplikasi yang dapat diakses menggunakan jaringan internet.

5. Bab V Penutup

Berisi kesimpulan dari penelitian yang telah dilakukan dan saran untuk peneliti selanjutnya yang akan menindaklanjuti atau mengembangkan hasil penelitian ke tingkat berikutnya.

1.8 Luaran

Luaran yang dihasilkan pada penelitian ini berupa prototipe alat pemantauan kualitas air pada tambak udang, aplikasi *web* sebagai wadah pembacaan data alat, dan artikel ilmiah.

BAB II

TINJAUAN PUSTAKA

2.1 Kualitas Air Tambak

Kualitas air tambak merupakan penunjang utama bagi kelangsungan hidup dan produktivitas budidaya udang. Adanya parameter dalam menentukan kualitas air berupa parameter fisika seperti suhu air, salinitas, kekeruhan air, dan sebagainya yang dapat menjadi penentu untuk menentukan buruknya kualitas air yang kemudian akan berdampak pada turunnya nafsu makan udang, pertumbuhan udang yang lambat, dan penyakit yang mudah terjangkit pada udang. Dampak buruk tersebut dapat diberi tindakan pencegahan dengan diperlukannya sistem untuk memantau kualitas air pada tambak sehingga air dapat terkendali dengan baik. (Samura, Ayu, Wijaya, & Setyawan, 2018).

2.1.1 Suhu

Suhu di dalam air merupakan faktor penentu yang menggunakan cakupan biologi, fisika, dan kimia yang bertujuan menjadi penentu kelangsungan hidup, pertumbuhan morfologi, tingkah laku, dan metabolisme makhluk hidup perairan yang ada di dalamnya. Besarnya pengaruh suhu air terhadap kehidupan ditentukan pula oleh daya toleransi serta kecepatan perubahan pada suhu di dalam air. Suhu Optimal untuk air tambak dalam budidaya udang berkisar antara 26-30°C, jika suhu berubah secara mendadak meskipun suhu air masih dalam kisaran normal, udang dapat mengalami stress dan bahkan dapat berakibat fatal. (Suwarsih, Marsoedi, Nuddin, & Mohammad, 2016)

2.1.2 pH Air

Salah satu aspek penting dalam budidaya udang adalah tingkat keasaman air, karena hal ini memengaruhi kesuburan perairan serta memengaruhi ekosistem di dalamnya. Materi organik seperti alga mati, sisa pakan, dan kotoran udang akan terakumulasi dan mengalami dekomposisi di dasar tambak, terbantu oleh aktivitas bakteri. Dalam kondisi oksigen yang terbatas, dekomposisi akan berlangsung secara anaerobik dan menghasilkan asam organik yang dapat larut dalam air. Nilai pH

optimal pada air tambak udang berkisar antara 7,5 - 8,8 untuk pertumbuhan ideal bagi udang. Fluktuasi pH yang signifikan dalam waktu singkat dapat menyebabkan stres pada udang, mengakibatkan penurunan nafsu makan dan penurunan sistem kekebalan tubuh, sehingga membuat udang lebih rentan terhadap penyakit. Jika kondisi ini berlanjut dalam jangka panjang, pH air yang tidak optimal akan menghambat pertumbuhan dan bahkan menyebabkan kematian udang, mengakibatkan tingkat kelangsungan hidup yang rendah (Imrana, et al., 2023).

2.1.3 Tingkat Keasaman Amonia

Amonium (NH_4) adalah senyawa yang terbentuk melalui proses dekomposisi protein, sisa-sisa pakan, dan hasil metabolisme udang dalam tambak. Pada kondisi pH tinggi, ammonium dapat berubah menjadi senyawa amoniak (NH_3) yang bersifat beracun (Lusiana, Ritonga, Sudrajat, & Arifin, 2021). Kondisi air tambak udang sesuai dengan persyaratan biologis memiliki kandungan amonia sebesar kurang dari 0.1 ppm (*parts per million*) (Anggraeni, Sri, Slamet, Endro, & Khamami, 2022), sehingga diperlukan perhatian khusus terhadap tingkat ammonium dalam tambak udang untuk memastikan kondisi lingkungan yang optimal bagi pertumbuhan dan kesejahteraan udang.

2.1.4 Derajat Kekeruhan Air

Kesalahan yang sering terjadi dalam budidaya udang adalah kurangnya perhatian terhadap kualitas air selama masa pemeliharaan. Untuk memastikan pertumbuhan yang optimal, udang membutuhkan kondisi lingkungan air tambak dengan kejernihan/kekeruhan 15 – 40 NTU (*Nephelometric Turbidity Unit*) (Anggraeni, Sri, Slamet, Endro, & Khamami, 2022). Kekeruhan air yang tinggi dapat mengganggu kesehatan dan pertumbuhan udang. Dalam konteks budidaya tambak udang, kekeruhan air yang tinggi dapat menghambat penetrasi cahaya matahari ke dalam air, mempengaruhi proses fotosintesis fitoplankton yang menjadi pakan alami udang. Selain itu, kekeruhan yang tinggi juga dapat mempengaruhi tingkat oksigen terlarut dalam air, yang merupakan faktor penting dalam respirasi udang. Oleh karena itu, pemantauan dan pengendalian kekeruhan air dalam tambak udang menjadi faktor kunci dalam memastikan lingkungan yang optimal bagi

pertumbuhan dan kelangsungan hidup udang. Penerapan standar kualitas air yang mengacu pada batas maksimal kekeruhan dapat menjadi pedoman dalam mengelola tambak udang secara efektif dan berkelanjutan.

2.1.5 Salinitas

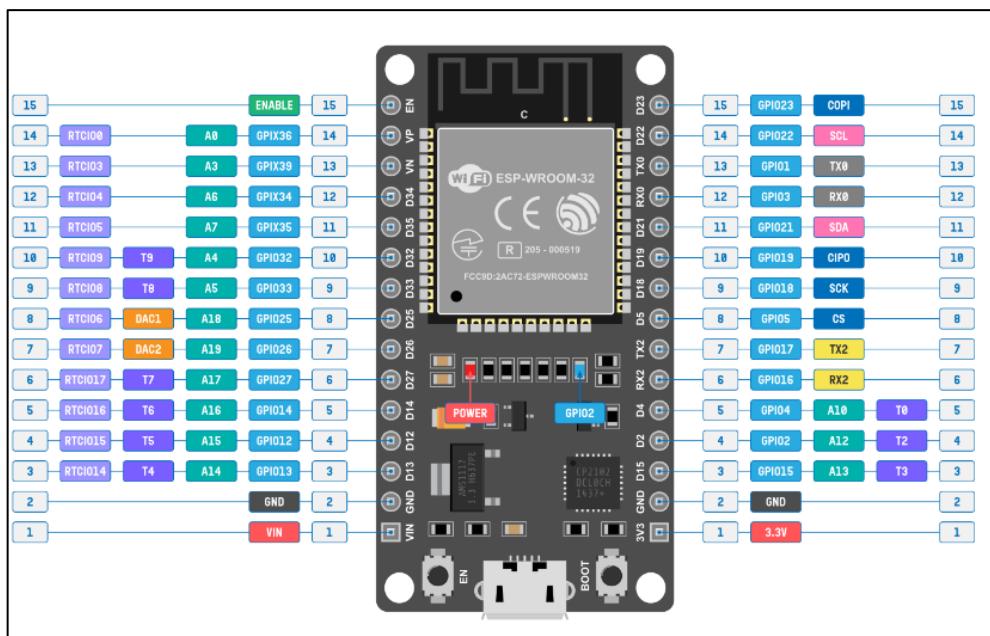
Penguapan air pada tambak yang disebabkan oleh suhu udara yang naik dan mengakibatkan kenaikan pada kandungan garam adalah proses yang disebut salinitas. Kandungan salinitas yang optimal untuk budidaya udang adalah beriksa di 10-30 ppt (*part per thousand*). Kenaikan salinitas dapat disebabkan oleh cuaca panas yang menyebabkan salinitas yang tinggi dan cuaca hujan yang menyebabkan konsentrasi garam turun karena adanya kandungan air tawar, tetapi dua faktor tersebut tidak menyebabkan kematian pada udang dan tidak mempengaruhi produksi (Purnamasari, I., Purnama, D., & Utami, 2017).

2.2 Mikrokontroler

Mikrokontroler adalah sistem komputer yang dikemas dalam satu chip IC (*Integrated Circuit*) dan biasa disebut sebagai single chip microcomputer. Sistem ini memiliki tugas spesifik, berbeda dengan PC (*Personal Computer*) yang memiliki berbagai fungsi. Mikrokontroler memiliki sistem mikroprosesor yang di dalamnya sudah ada CPU (*Central Processing Unit*), ROM (*Read Only Memory*), RAM (*Random Access Memory*), I/O (*Input/Output*), Clock dan peralatan internal lainnya yang sudah terhubung dan terorganisasi dengan baik oleh pabrik pembuatnya. Teknologi yang digunakan pada mikrokontroler AVR adalah RISC (*Reduced Instruction Set Computer*), berbeda dengan seri MCS-51 yang menggunakan teknologi CISC (*Complex Instruction Set Computer*). Mikrokontroler AVR memiliki beberapa kelas seperti keluarga ATtiny, keluarga AT90Sxx, keluarga ATMega, dan keluarga AT89RFxx. Perbedaan antar kelas terletak pada memori, kelengkapan peripheral dan fungsinya. Meski begitu, dari segi arsitektur dan instruksi yang digunakan hampir sama. Salah satu produk AVR dari Atmel yang digunakan adalah ATMega8535.

2.2.1 ESP32

Mikrokontroler *ESP32* adalah sebuah sistem terpadu dalam satu chip SoC (*System on Chip*) yang dilengkapi dengan Wi-Fi 802.11 b/g/n, Bluetooth versi 4.2, dan berbagai peripheral. *ESP32* adalah chip yang cukup lengkap karena memiliki prosesor, penyimpanan, dan akses ke GPIO (*General Purpose Input Output*). *ESP32* bisa digunakan sebagai pengganti Arduino karena memiliki kemampuan untuk terkoneksi ke Wi-Fi secara langsung (Wagyana, 2019). *ESP32* mempunyai 18 pin ADC (*Analog to Digital Converter*) (12-bit) yang diilustrasikan pada Gambar 2.2. Dalam penggunaan *ESP32*, pengukuran nilai analog memungkinkan untuk mendeteksi tegangan yang bervariasi dari 0 V hingga 3,3 V. Tegangan yang terukur dikonversikan ke dalam nilai numerik yang berkisar antara 0 hingga 4095, di mana tegangan 0 V diasosiasikan dengan nilai 0, sedangkan tegangan 3,3 V diasosiasikan dengan nilai 4095. Setiap tegangan di antara rentang tersebut akan diberikan nilai sesuai dengan tingkat tegangannya.



Gambar 2. 1 Struktur Pin pada ESP32 (circuitstate.com).

Semua pin diberi label di bagian atas board sehingga mudah untuk dikenali. Board ini memiliki interface USB (*Universal Serial Bus*) ke UART (*Universal Asynchronous Receiver/Transmitter*) yang mudah diprogram dengan program

pengembangan aplikasi seperti *Arduino IDE*. Sumber daya board bisa diberikan melalui konektor microUSB.

2.3 Sensor

Sensor adalah komponen penting dalam sistem yang berinteraksi langsung dengan proses yang sedang diukur. Sensor mengubah variabel input menjadi output dalam format tertentu, memungkinkan sistem pengukuran lainnya untuk mengetahui nilai variabel tersebut. Dalam sistem instrumentasi yang digunakan untuk pengukuran, input adalah nilai aktual dari variabel yang sedang diukur, sementara output adalah nilai variabel yang telah diukur.

2.3.1 Sensor Suhu

Sensor suhu adalah alat yang dapat mengubah panas menjadi listrik, memungkinkan kita untuk mendeteksi perubahan suhu. Sensor ini mengukur jumlah energi panas atau dingin yang dihasilkan oleh objek, sehingga kita bisa mengetahui perubahan suhu yang terjadi. Sensor suhu yang digunakan dalam penelitian ini adalah model DS18B20, yang dicelupkan ke dalam kolam untuk melakukan pengukuran dalam air. Output dari sensor ini kemudian diterima oleh rangkaian pengkondisian sinyal dan diperkuat sehingga hasil pembacaannya bisa dikonversi menjadi nilai suhu dalam celcius (Wardhana, 2022).



Gambar 2. 2 Sensor Suhu DS18B20 (randomnerdtutorials.com).

Pada Gambar 2.2, diilustrasikan bahwa sensor ini memiliki tiga kabel output: positif, negatif, dan data. Kabel data disambungkan dengan resistor sebelum

disambungkan ke pin digital yang terdapat pada mikrokontroller, mengingat bahwa sensor suhu yang digunakan menggunakan input berupa input digital.

Tabel 2. 1 Penyambungan Sensor Suhu Pada Mikrokontroler.

Sensor Suhu	Port ESP32
GND	GND
DO	Pin Analog Kompatibel
VDD	3.3V atau 5V

Pembacaan output pada sensor suhu DS18B20 menggunakan rangkaian seperti pada Gambar 2.2 , menggunakan resistor sebagai penunjang daya tambahan antara pin untuk data dan pin VCC. Kemudian untuk pembacaan data pada sensor suhu, diperlukan program untuk mengambil datanya dan menampilkannya pada tampilan serial pada program Arduino IDE.

2.3.2 Sensor pH

Sensor pH adalah perangkat elektronik yang dirancang khusus untuk mengukur pH air dan berbagai larutan lainnya. Sensor ini dilengkapi dengan elektroda yang berperan penting dalam proses pengukuran dan penampilan pembacaan pH.



Gambar 2. 3 Sensor pH 4502C (guguslangit.wordpress.com).

Bagian yang paling vital dari sensor ini adalah probe atau elektroda. Modul sensor pH beroperasi dengan membaca nilai pH dari elektroda sensor, yang kemudian diubah menjadi nilai tegangan analog. Nilai ini selanjutnya disambungkan ke pin digital pada mikrokontroller (Wardhana, 2022).

Pada sisi lain, papan konektor sinyal dilengkapi dengan LED yang berfungsi sebagai indikator daya. Papan ini juga memiliki konektor jenis BNC dan antarmuka untuk sensor pH. Untuk mengoperasikannya, hanya perlu menghubungkan sensor pH dengan konektor BNC, kemudian menyambungkan antarmuka pH ke port input analog dari mikrokontroler apa pun. Jika telah diprogram, Anda dapat dengan mudah mendapatkan nilai pH.

Tabel 2. 2 Penyambungan Sensor pH Pada Mikrokontroler.

Sensor pH	Port ESP32
PO	Pin Analog Kompatibel
GND	GND
VCC	3.3V

Sensor pH 4502c digunakan dengan metode pengukuran yang melibatkan penggunaan fungsi rumus untuk menghitung tegangan. Pertama, nilai ADC (*Analog to Digital Converter*) dari sensor pH diambil untuk menghitung tegangan sensor dengan rumus berikut:

$$\text{tegangan pH} = \text{nilai ADC pH} \times \left(\frac{\text{tegangan mikrokontroler}}{\text{analog level mikrokontroler}} \right) \quad (2- 1)$$

Tegangan sensor diatur dengan mengatur *potensiometer* yang terletak pada papan konektor sensor, sebagaimana ditunjukkan pada Gambar 2.3, hingga tegangan sensor mencapai 2,5V. Untuk menghitung nilai pH cairan, digunakan rumus berikut:

$$\text{nilai pH} = - 5.70 \times \text{tegangan pH} + \text{nilai kalibrasi} \quad (2- 2)$$

Dimana nilai kalibrasi didapatkan dengan melihat nilai cairan pH yang terukur oleh sensor (Usina, 2018).

2.3.3 Sensor Gas

Sensor gas adalah perangkat elektronik yang dirancang untuk mengukur kadar konsentrasi gas seperti amonia pada permukaan air. Hal ini penting karena kadar amonia yang tinggi pada permukaan air kolam atau air tambak, dapat menunjukkan adanya ketidakseimbangan dan dapat menghambat pertumbuhan makhluk hidup yang hidup di dalam budidaya perairan.



Gambar 2. 4 Sensor Gas MQ-135 (quartzcomponents.com).

Sensor MQ-135 pada Gambar 2.5 berfungsi untuk mendeteksi konsentrasi gas pada udara. Sensor ini terdiri dari tabung aluminium yang dikelilingi oleh silikon dan di pusatnya terdapat elektroda yang terbuat dari aurum, tempat elemen pemanas berada. Proses pemanasan menyebabkan kumparan dipanaskan sehingga keramik SnO_2 menjadi semikonduktor atau penghantar, melepaskan elektron. Ketika amonia dideteksi oleh sensor dan mencapai elektroda aurum, sensor MQ-135 akan menghasilkan tegangan analog (Budiman, Rivai, & Nugroho, 2019).

Tabel 2. 3 Penyambungan Sensor Gas Pada Mikrokontroler.

Sensor Gas	Port ESP32
AO	Pin Analog Kompatibel
GND	GND
VCC	3.3V atau 5V

Untuk mendapatkan nilai amonia pada sensor adalah mengetahui nilai resistansi sensor sebesar 100ppm pada NH₃ terhadap udara segar atau yang disebut dengan RO, dan nilai resistansi sensor terhadap berbagai macam konsentrasi gas atau yang disebut dengan RS. Sensor perlu di kalibrasi dengan mengetahui nilai RO.

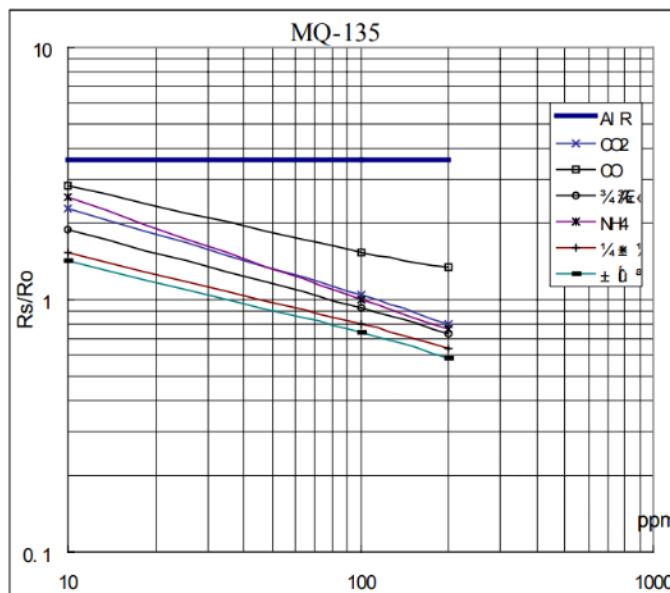
$$\text{resistansi sensor (RS)} = \left(\left(\frac{\text{tegangan sirkuit}}{\text{tegangan sensor}} \right) - 1 \right) \times RL \quad (2-3)$$

Nilai RO didapatkan dengan menggunakan persamaan berikut:

$$RO = \frac{RS}{3,6} \quad (2-4)$$

Nilai 3,6 didapat dengan melihat grafik pada gambar 2.5, yang mempunyai arti bahwa udara pada umumnya mempunyai rasio sebesar 3,6 dan untuk mendapatkan nilai rasio dari kandungan gas yang lain setelah mendapatkan nilai Ro.

$$\text{ratio} = \frac{RS}{RO} \quad (2-5)$$



Gambar 2. 5 Grafik Rasio Sensor (components101.com).

Untuk mendapatkan nilai PPM, dapat menggunakan persamaan berikut:

$$PPM = 10^{\left\{ \left[\frac{\log(\text{ratio}) - b}{m} \right] \right\}} \quad (2-6)$$

Nilai m didapatkan dengan menghitung perpotongan grafik pada gas yang ingin dihitung nilainya.

$$m = \frac{[\log(y_2) - \log(y_1)]}{[\log(x_2) - \log(x_1)]} \quad (2-7)$$

Untuk menghitung nilai b dapat menggunakan persamaan berikut: (Sanjaya, 2020).

$$b = \log(y) - m \times \log(x) \quad (2-8)$$

2.3.4 Sensor Kekeruhan

Sensor kekeruhan berfungsi sebagai alat pengukur tingkat kekeruhan air dalam kolam tambak. Prinsip kerja sensor ini didasarkan pada pemancaran cahaya oleh LED inframerah yang selanjutnya diterima oleh rangkaian sensor fotodiode. Nilai yang dibaca oleh sensor fotodiode ini akan berubah sesuai dengan tingkat kekeruhan air.

Tabel 2. 4 Penyambungan Sensor Kekeruhan Pada Mikrokontroler.

Sensor Kekeruhan	Port ESP32
AO	Pin Analog Kompatibel
GND	GND
VCC	3.3V atau 5V

Sensor kekeruhan adalah jenis sensor dengan output data dalam bentuk analog. Sensor ini dilengkapi dengan tiga kabel keluaran, yaitu pin positif (VCC), pin negatif (GND), dan pin data yang dapat dilihat pada Gambar 2.6.

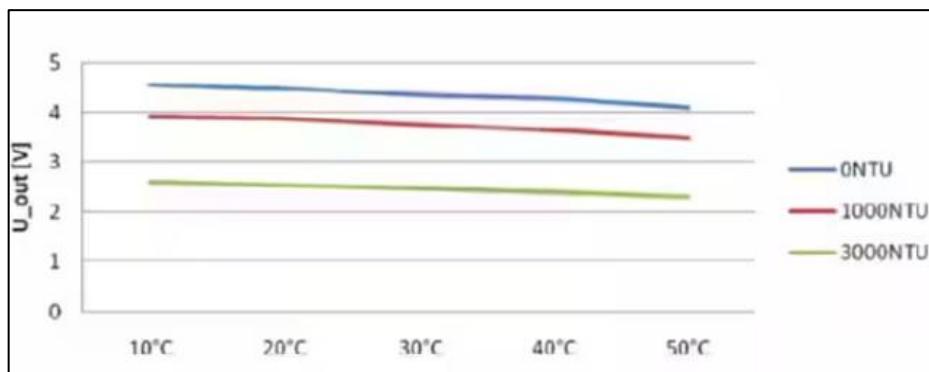
Nilai ADC didapatkan dengan membaca nilai analog pada sensor dengan menghubungkannya dengan mikrokontroler, dan pembacaan air bersih seharusnya menghasilkan NTU dengan nilai kurang dari 0,5 dan tegangannya di 4,1V sampai 4,3V.

$$\text{tegangan kekeruhan} = \text{nilai ADC kekeruhan} \times \left(\frac{\text{tegangan mikrokontroler}}{\text{analog level mikrokontroler}} \right) \quad (2-9)$$



Gambar 2. 6 Sensor Kekeruhan SEN-0189 (bukalapak.com).

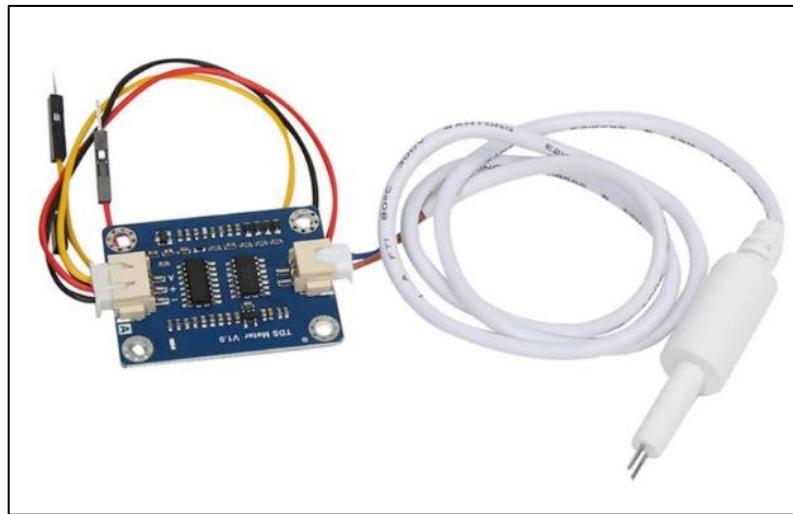
Pada gambar 2.6, semakin tinggi tegangan yang dicapai, maka pembacaan nilai NTU juga akan meningkat pada temperatur 10-50°C (DFRobot, 2017).



Gambar 2. 7 Grafik Nilai Sensor Kekeruhan (wiki.dfrobot.com).

2.3.5 Sensor TDS

Sensor TDS (*Total Dissolved Solids*) adalah sebuah perangkat elektronik yang berfungsi untuk mengukur konsentrasi zat terlarut dalam suatu larutan. Zat-zat kimia yang paling sering ditemukan dan berpotensi memengaruhi tingkat TDS dalam air mencakup unsur-unsur seperti fosfat, kalsium, natrium, kalium, dan timbal (Wardhana, 2022).



Gambar 2. 8 Sensor TDS KS0429. (www.techmasterkw.com)

Peningkatan konsentrasi TDS dalam air seringkali berasal dari limbah maupun dari faktor alam seperti cuaca panas dan hujan. Sensor TDS ini biasanya merupakan sensor dengan output data dalam bentuk analog dan memiliki tiga kabel yang terdiri dari positif, negatif, dan saluran data yang diilustrasikan pada Gambar 2.6.

Tabel 2. 5 Penyambungan Sensor TDS Pada Mikrokontroler.

Sensor TDS	Port ESP32
A	Pin Analog Kompatibel
-	GND
+	5V

2.4 Internet Of Things (IOT)

Internet of Things (IoT) merupakan konsep di mana berbagai objek atau "Things" dapat terhubung dengan internet. Contohnya termasuk orang yang memiliki monitor jantung terimplan, hewan ternak dengan transponder biochip, atau bahkan mobil dengan sensor bawaan untuk memberi peringatan kepada pengemudi tentang tekanan ban yang rendah. IoT erat kaitannya dengan komunikasi *machine-to-machine* (M2M) terutama dalam sektor manufaktur, listrik, perminyakan, dan gas. Produk yang dikembangkan dalam konteks ini sering disebut

sebagai sistem cerdas atau "smart" seperti smart label, smart meter, dan sensor grid pintar. Meskipun konsep ini belum begitu populer hingga tahun 1999, IoT telah mengalami perkembangan selama beberapa dekade. Salah satu contohnya adalah mesin minuman Coke di Carnegie Mellon University pada awal 1980-an, di mana para programer dapat terhubung ke mesin melalui internet untuk memeriksa status dan ketersediaan minuman dingin tanpa harus pergi ke mesin tersebut. Istilah "IoT" pertama kali dikenalkan oleh Kevin Ashton, salah satu pendiri dan direktur eksekutif dari Auto-ID Center di MIT pada tahun 1999. Dengan infrastruktur internet yang semakin berkembang, kita memasuki era di mana tidak hanya *smartphone* atau komputer yang dapat terhubung ke internet. Berbagai objek fisik seperti mesin produksi, mobil, peralatan elektronik, hingga peralatan yang dapat dikenakan oleh manusia (*wearables*), semuanya akan terhubung ke jaringan lokal dan global melalui sensor dan/atau aktuator yang tertanam. Ini menandai transformasi besar dalam cara berinteraksi dengan objek-objek di sekitar kita, membuka potensi baru untuk efisiensi dan kecerdasan dalam berbagai aspek kehidupan sehari-hari (Adani, Farhan, & Salma, 2019).

2.5 MySQL

MySQL adalah sebuah sistem manajemen basis data (*Database Management System* - DBMS) yang bersifat open source, mendukung penggunaan multiuser dan multithreading, dan tersedia secara gratis. Berdasarkan teori ini, dapat disimpulkan bahwa SQL (*Structured Query Language*) adalah bahasa permintaan yang digunakan dalam pengelolaan basis data. SQL memungkinkan pengguna untuk membuat, memanipulasi, dan mengelola data di dalam basis data. Bahasa SQL digunakan untuk menjalankan berbagai tugas, termasuk operasi pembaruan data, yang sesuai dengan prinsip-prinsip dasar dari Sistem Manajemen Basis Data Relasional (*Relational Database Management System* – RDBMS) (Noviana & Rina, 2022). MySQL sangat populer di berbagai perusahaan ternama seperti Google, Wikipedia, Facebook, Twitter, Flickr, Wordpress, Joomla, dan sebagainya. Koneksi antara MySQL dan Web dapat dilakukan melalui bahasa pemrograman PHP. Kedua teknologi ini sering digunakan bersamaan, dengan PHP memungkinkan penyediaan syntax query untuk MySQL.

2.6 Application Programming Interface (API)

Application Programming Interface (API) adalah antarmuka yang memfasilitasi komunikasi antara dua aplikasi, dalam konteks sebagai fungsi di sisi *backend* yang melakukan *query* data spesifik pada sumber penyimpanan yang persisten melalui antarmuka yang didefinisikan sebagai layanan web API. Konsep layanan web ini muncul untuk menjembatani sistem informasi yang ada (*middleware*) tanpa mempermasalahkan perbedaan platform yang digunakan oleh setiap sumber (Lawi, Panggabean, & Yoshida, 2021). Penggunaan API dalam IoT mempunyai standar pada *response time* atau waktu bagi API dalam menerima data dan menampilkannya pada sebuah luaran, dan untuk mendapatkan luaran yang efisien, API harus menerima data kurang dari 100 ms (*milliseconds*) atau 0,1 detik (Gregersen, 2023).

2.7 Postman

Postman merupakan alat pengembangan API (*Application Programming Interface*) yang menyediakan fitur untuk mempermudah proses pengembangan API. *Postman* mendukung berbagai metode permintaan HTTP (*Hyper Text Transfer Protocol*) seperti GET, POST, PUT, DELETE, dan PATCH, yang memungkinkan pengembang berinteraksi secara mendalam dengan API. Fleksibilitas ini penting karena berbagai API memerlukan metode yang berbeda untuk pengambilan data, pengiriman data, pembaruan data, atau penghapusan data. (Hooda, 2024)

2.8 Framework

Framework adalah struktur yang digunakan sebagai dasar dalam membangun dan mengembangkan aplikasi atau sistem. Dengan menyediakan komponen dasar, pola desain, dan praktik terbaik, framework mempercepat proses pengembangan perangkat lunak. *Framework* juga menyediakan API (*Application Programming Interface*) yang memungkinkan aplikasi berinteraksi dengan komponen lainnya. Selain itu, framework juga memiliki komponen pengelolaan sumber daya, seperti database, file, dan jaringan yang membantu dalam

penyimpanan data, pengelolaan pengguna, serta pengiriman dan penerimaan informasi melalui jaringan.

2.9 Laravel

Laravel adalah sebuah *framework* berbahasa PHP yang digunakan untuk membuat *website* yang basisnya menggunakan arsitektur MVC (*Model-View-Controller*). Arsitektur MVC yang digunakan dalam *laravel* dapat didefinisikan sebagai berikut:

- *Model*: Komponen *model* digunakan untuk mengatur *input* dan *output* data yang ada di dalam *database*.
- *View*: Komponen *view* digunakan untuk membuat tampilan antarmuka yang ada di dalam aplikasi beserta komponen-komponennya.
- *Controller*: Komponen *controller* digunakan untuk membuat fungsi atau perintah untuk mengambil data dari *database* dan menampilkannya di halaman antaramuka aplikasi.

Penggunaan MVC di dalam *laravel* dapat mempermudah alur kerja dalam memodifikasi kode atau program ketika terjadi kendala atau perubahan fitur tanpa mengubah kode atau program dari komponen lain (Psathwara, 2024).

2.10 Penelitian Terdahulu

Untuk mendukung penelitian yang sedang dikerjakan, penulis merujuk pada beberapa penelitian terdahulu. Penelitian oleh (Pauzi, Syafira, Surtono, & Supriyanto, 2017) menggunakan sistem monitoring kualitas air tambak udang berbasis IoT dengan aplikasi Blynk dan Arduino Uno. (Faruq & Hirawan, 2018) mengembangkan sistem monitoring kualitas air di tambak udang Vaname menggunakan teknologi Internet of Things. Penelitian oleh (Intan, Nurlina, Pangerang, & Mulyawan, 2020) fokus pada sistem monitoring sirkulasi air dalam budidaya udang Vaname berbasis IoT. Penelitian oleh (Wardhana, 2022) yang fokus pada *interfacing* dan membuat rancang bangun prototipe menggunakan sensor-sensor yang berhubungan dengan kualitas budidaya air tambak.

Persamaan yang dimiliki pada penelitian yang sedang dikerjakan dan yang digunakan sebagai refensi adalah sebagai berikut:

- Mengukur kualitas air pada kolam tambak budidaya udang.
- Menggunakan IoT sebagai media *input* dan *output*.
- Menggunakan mikrokontroler sebagai perangkat perekaman data.
- Menggunakan sensor ukur yang berhubungan dengan kualitas air.

Adapun perbedaan yang dikembangkan pada penelitian yang sedang dikerjakan untuk mengembangkan sebuah alat penelitian baru adalah sebagai berikut:

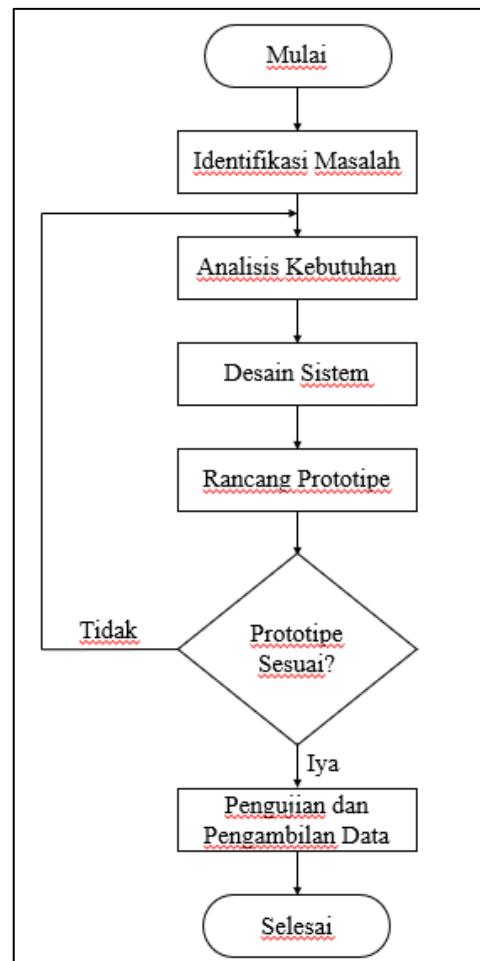
- Menggunakan mikrokontroler dengan spesifikasi lebih tinggi dan mempunyai fitur untuk dapat terhubung dengan jaringan internet.
- Menggunakan tambahan sensor ukur yang berhubungan dengan kualitas air.
- Menggunakan *framework* dalam mengembangkan media *output* berbasis IoT.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Tahapan Penelitian

Gambar 3.1 menunjukkan tahapan penelitian dalam pembuatan prototipe perangkat IoT untuk pengukuran kualitas air.



Gambar 3. 1 Bagan Tahapan Perancangan Prototipe Alat.

Tahap awal penelitian dimulai dengan melakukan analisis terhadap permasalahan yang ada pada studi kasus yang diteliti dan kebutuhan terkait alat dan bahan yang diperlukan untuk merancang sebuah prototipe alat kualitas air pada tambak udang. Tahap selanjutnya adalah melakukan tahap analisis terhadap perangkat keras dan perangkat lunak yang diperlukan untuk disesuaikan dengan kebutuhan penelitian dengan gambaran latar belakang yang sudah dibuat.

Kemudian tahap berikutnya adalah merancang sebuah sistem dalam bentuk gambar diagram dengan tujuan untuk mendeskripsikan garis besar bagaimana sistem dari prototipe bekerja. Selanjutnya adalah pembuatan perangkat keras dari rancangan sistem yang sudah direncanakan sesuai dengan gambar diagram rancangan. Jika rancangan tidak sesuai maka dilakukan analisis kebutuhan lebih lanjut untuk melihat apakah ada kebutuhan yang kurang baik dari faktor perangkat keras maupun perangkat lunak sampai prototipe tersebut sesuai dengan sistem yang sudah dibuat. Setelah semua tahapan sebelumnya sudah berhasil dilalui, tahapan selanjutnya meruapakan tahapan pengujian sistem secara keseluruhan berupa pengambilan data yang bertujuan untuk pengujian metode yang kemudian dapat ditarik sebuah kesimpulan, serta dapat mengukur sejumlah parameter pada penelitian pengukuran kualitas air pada tambak udang.

3.2 Identifikasi Masalah

Masalah yang didapat dari pembudidayaan tambak udang yang masif adalah kolektifitas data kualitas air secara masif yang masih dilakukan dengan cara konvensional dengan melakukan pengukuran oleh tenaga manusia dan diukur satu persatu kolam. Data yang sudah didapat juga harus dikirim melalui akomodasi yang cukup memakan waktu mengingat data kualitas air yang sudah tercatat akan diolah melalui departemen yang mempunyai jarak fasilitas yang cukup jauh dari kolam yang sudah dilakukan pengukuran.

3.3 Analisis Kebutuhan

Prototipe sistem membutuhkan komponen serta elemen-elemen pendukung yang bertujuan untuk menunjang keberhasilan dalam pembuatan prototipe. Berikut adalah perangkat keras dan perangkat lunak yang dibutuhkan dalam pembuatan prototipe. Perangkat keras yang dibutuhkan yaitu:

- a. Mikrontroler *ESP32*.
- b. Sensor Suhu DS18B20.
- c. Sensor pH 4502C.
- d. Sensor Amonia MQ-135.
- e. Sensor Kekaruan SEN-0189.

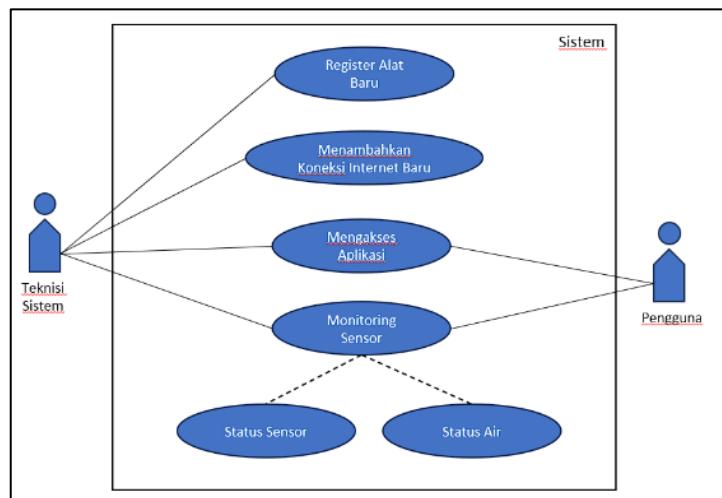
- f. Sensor TDS KS0429.
- g. Laptop atau *Personal Computer*.
- h. Kabel USB-A ke Micro-USB.

Laptop atau *Personal Computer* diperlukan sebagai perantara untuk menanamkan sebuah program yang terhubung *ESP32* menggunakan kabel USB-A ke Micro USB dengan tujuan untuk menghubungkan *ESP32* dengan Kumpulan sensor-sensor yang dibutuhkan untuk membaca data. Sebab itu, diperlukan beberapa perangkat lunak tambahan untuk membaca data dan melakukan simulasi IoT, yaitu:

- a. *Arduino IDE*.
- b. *Visual Studio Code*.
- c. *Framework Laravel*.
- d. *XAMPP*.

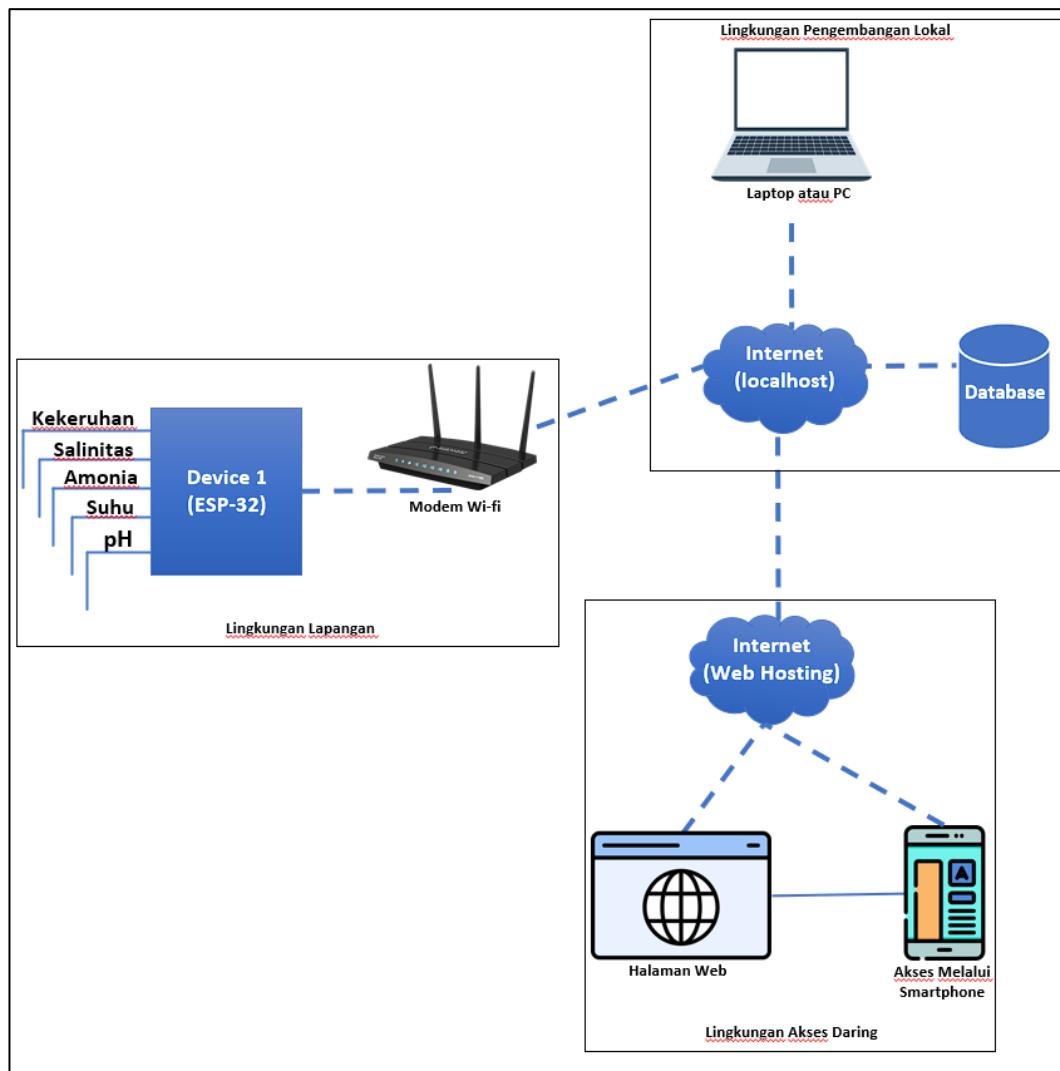
3.4 Perancangan Desain Sistem

Prototipe alat yang sudah dibuat dihubungkan dengan sistem IoT yang dapat memungkinkan pengguna untuk melakukan pemantauan terhadap kualitas air dengan mengakses aplikasi *Web* yang digunakan sebagai luaran untuk menampilkan data kualitas air. Ilustrasi penggunaan aplikasi *web* diilustrasikan pada Gambar 3.2.



Gambar 3. 2 Use-Case Diagram Untuk Pengguna.
Teknisi sistem mempunyai wewenang dalam melakukan registrasi alat baru berupa pemasangan serta pengoptimalan alat pengukuran baru yang dipasangkan

kepada kolam. Halaman *web* dapat diakses oleh teknisi sistem karena semua perubahan data pengkuran terjadi hanya di alat dan sensor yang sudah terpasang. Pengguna hanya dapat mengakses aplikasi *web* yang mempunyai fitur untuk menampilkan data status air secara *realtime* dan status sensor yang mengindikasikan sensor tersebut masih berfungsi atau tidak.



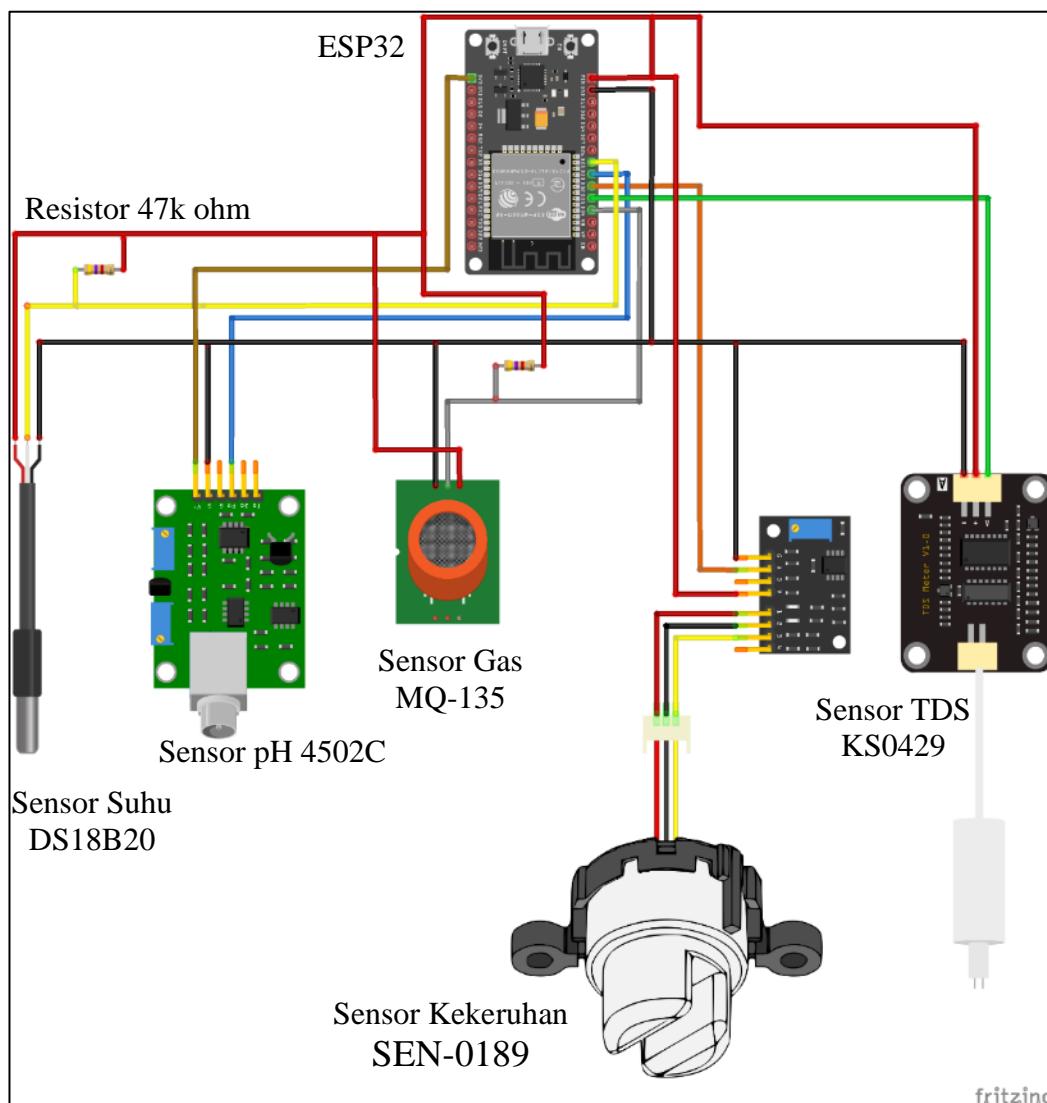
Gambar 3. 3 Desain Sistem IoT pada Tambak Udang.

Gambar 3.3 mengilustrasikan desain sistem pada prototipe alat yang sudah dibuat. *ESP32* dinamakan *Device* sebagai perangkat utama yang menampung data dari sensor-sensor yang sudah dipasang, lalu *ESP32* dihubungkan menggunakan modem wi-fi atau jaringan internet. Data dari *ESP32* ditampung di dalam *database*

lokal di dalam perangkat Laptop atau *PC*. Menggunakan *Web Hosting*, data yang sudah tertampung di *database* lokal dapat diakses menggunakan halaman *web* yang sudah dibuat dan dapat diakses daring oleh pengguna.

3.4.1 Rangkaian Prototipe Alat

Rancangan prototipe yang telah dikembangkan memegang peranan utama dalam pemantauan kualitas air dalam tambak melalui integrasi teknologi *Internet of Things* (IoT) dan sistem cerdas.



Gambar 3. 4 Rangkaian Prototipe Sistem Pada ESP32.
Prototipe ini mencakup sejumlah komponen sensor krusial yang bekerja bersama-sama untuk mengukur dan memantau parameter kualitas air dengan

akurat. Mikrokontroller *ESP32* menjadi inti dari sistem ini, memungkinkan pengolahan dan pengiriman data antara sensor-sensor yang digunakan.

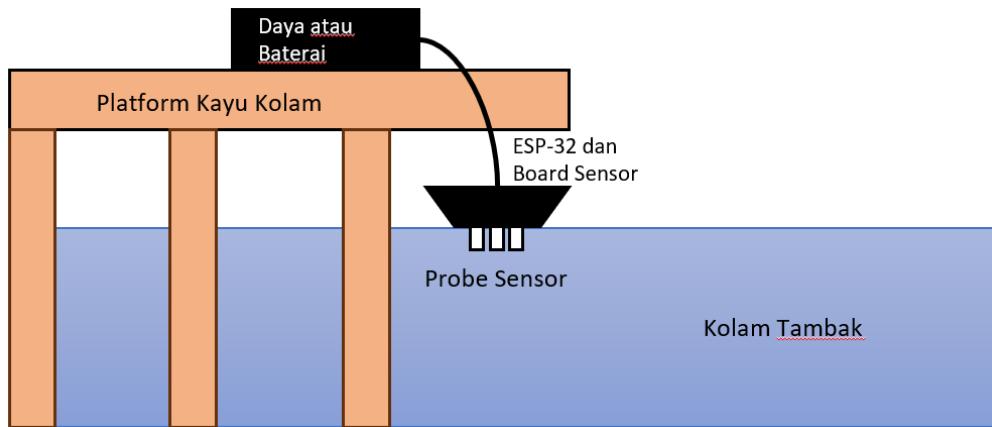
Sensor-sensor yang terlibat dalam prototipe ini mencakup sensor suhu DS18B20, sensor pH 4502C, sensor gas MQ-135, sensor kekeruhan SEN-0189, dan sensor TDS KS0429. Masing-masing sensor telah dirancang dengan spesifikasi khusus yang memungkinkannya untuk melakukan pengukuran dengan akurat. Ada beberapa hal penting yang perlu diperhatikan terkait dengan penggunaan sensor-sensor ini. Sebagai contoh, sensor suhu DS18B20, sensor pH 4502C, sensor kekeruhan SEN-0189, dan sensor TDS KS0429 memerlukan pencelupan langsung ke dalam sampel air tambak untuk memantau parameter tertentu. Sementara itu, sensor gas *MQ-135* ditempatkan di atas permukaan air, memungkinkannya untuk membaca kadar gas yang dihasilkan dari sampel air. Integrasi yang efektif antara mikrokontroller dan sensor-sensor ini memastikan bahwa data terkait kualitas air dapat diakses secara real-time melalui *serial monitor* pada aplikasi *Arduino IDE*.

Gambar 3.4 menyajikan dengan cermat konfigurasi perangkat fisik dan elektronik yang digunakan dalam rancangan prototipe ini. Mikrokontroller *ESP32* terhubung dengan sensor-sensor utama melalui sejumlah pin yang memfasilitasi aliran daya dan transmisi data. Masing-masing sensor memiliki tiga pin utama, yaitu pin *VCC* sebagai sumber daya, *ground* (GND), dan pin untuk mentransmisikan data ke *ESP32*. Sensor-sensor dengan pin *VCC* dapat dihubungkan secara langsung ke pin 3v3 (3.3 volt) pada *ESP32*, memungkinkan aliran daya yang stabil. Penting untuk dicatat bahwa sensor suhu *DS18B20* memerlukan resistor tambahan sebesar 4.7k ohm untuk memastikan stabilitas daya selama proses pengukuran.

3.4.2 Rancangan Peletakan Alat

Peletakan prototipe alat menjadi penentu untuk pengukuran kolektif data air dikarenakan prototipe alat merupakan alat otomasi yang mengumpulkan data air secara otomatis selamat prototipe alat tersambung dengan daya. Peletakan prototipe alat yang dapat diilustrasikan pada Gambar 3.5 merupakan skema perencanaan prototipe alat dipasang pada kolam tambak dengan memanfaatkan objek sekitar yang ada pada kolam tambak dan menggunakan komponen tambahan seperti

baterai atau daya listrik untuk alat dapat menyala dan bekerja semestinya.



Gambar 3. 5 Rancangan Peletakan Prototipe Alat.

Prototipe alat dibagi menjadi beberapa komponen seperti pada Gambar 3.5. *ESP32* dan *board* pembaca sensor diletakkan diatas *platform* kayu yang sudah tersedia pada kolam tambak, dan *ESP32* sudah terhubung dengan baterai. *Probe* sensor dimuat pada sebuah media apung dimana *probe* yang mempunyai ketahanan terhadap air akan dicelupkan kedalam air, sedangkan *probe* yang tidak tahan air akan ditempelkan di atas permukaan media apung. Media apung yang berisi Kumpulan *probe* sensor kemudian dihubungkan dengan *ESP32* yang berada di atas *platform* kayu.

3.4.3 Arduino IDE

Perangkat prototipe yang sudah dibuat akan dihubungkan dengan perangkat *personal computer* berupa laptop yang sudah terpasang program *Arduino IDE* yang bertujuan sebagai perantara perangkat prototipe untuk menampilkan data dari sensor-sensor yang sudah dijalankan untuk akusisi data kadar air. Data numerik yang tampil di *serial monitor* dalam perangkat lunak *Arduino IDE* tidak tertampung di dalam perangkat lunak maupun perangkat keras. Karena fungsi dari *Arduino IDE* hanya sebagai perangkat lunak untuk menanamkan sebuah sistem atau program di dalam mikrokontroler *ESP32* untuk dapat berkomunikasi dengan sensor-sensor yang sudah terpasang dan sebagai media luaran untuk menampung data numerik secara sementara.

3.4.4 Data Kualitas Air

Pengukuran kualitas pada air tambak udang mempunyai parameter untuk menunjang pertumbuhan serta perkembangan udang. Berikut adalah parameter kualitas air yang dibutuhkan untuk perkembangan tambak udang.

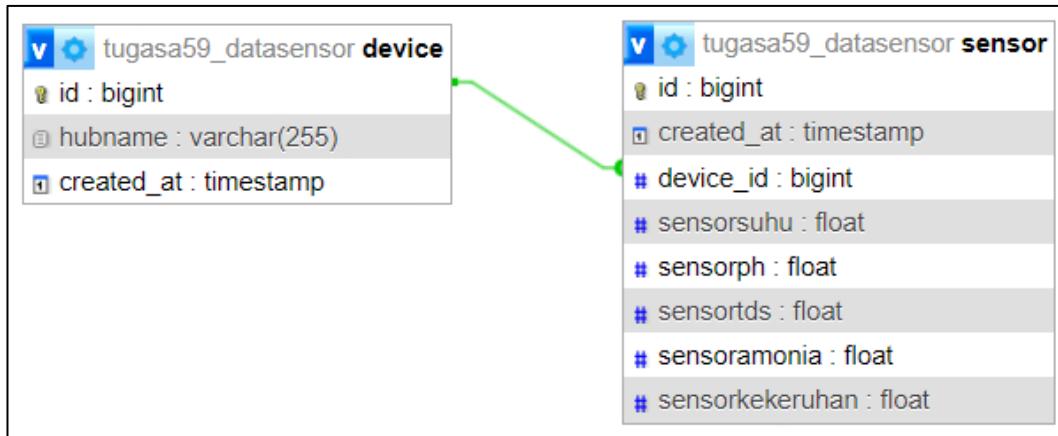
Tabel 3. 1 Parameter Kualitas Air Tambak.

No	Parameter	Batas Bawah	Batas Atas
1.	Suhu	26°C	30°C
2.	pH	7.5	8.8
3.	Amonia	-	0.1 ppm
4.	Salinitas	10 ppt	30 ppt
5.	Kekeruhan	15 NTU	40 NTU

3.4.5 Basis Data *MySQL*

Perancangan sistem untuk menampilkan melalui halaman *web* adalah dengan menyimpan data yang sudah terukur dari prototipe perangkat kedalam *database* yang menggunakan *engine database* jenis *MySQL*.

Database terdiri dari 2 tabel dengan nama tabel *hub* dan tabel *datasensor* yang mempunyai fungsinya masing-masing di dalam sistem. Pada Gambar 3.6, *database* yang mempunyai tabel *device* mempunyai relasi *one to many* yang mengindikasikan data dari sensor-sensor tersimpan dalam tabel *sensor* yang terdiri dari banyaknya kolom yang akan merepresentasikan nama-nama sensor yang akan tersimpan di dalam baris dari tiap kolom.



Gambar 3. 6 Skema Database Pada Sistem.

Tabel *device* digunakan untuk menampung kumpulan data perangkat *ESP32* yang terdaftar ke dalam sistem, dan setiap perangkat *ESP32* yang terdaftar di dalam tabel *hub* mempunyai tabelnya masing-masing untuk menampung data dari sensor-sensor yang sudah terpasang di *ESP32*. Skema dari database sistem yang diilustrasikan pada gambar 3.6 dan ilustrasi tabel *device* pada tabel *datasensor*.

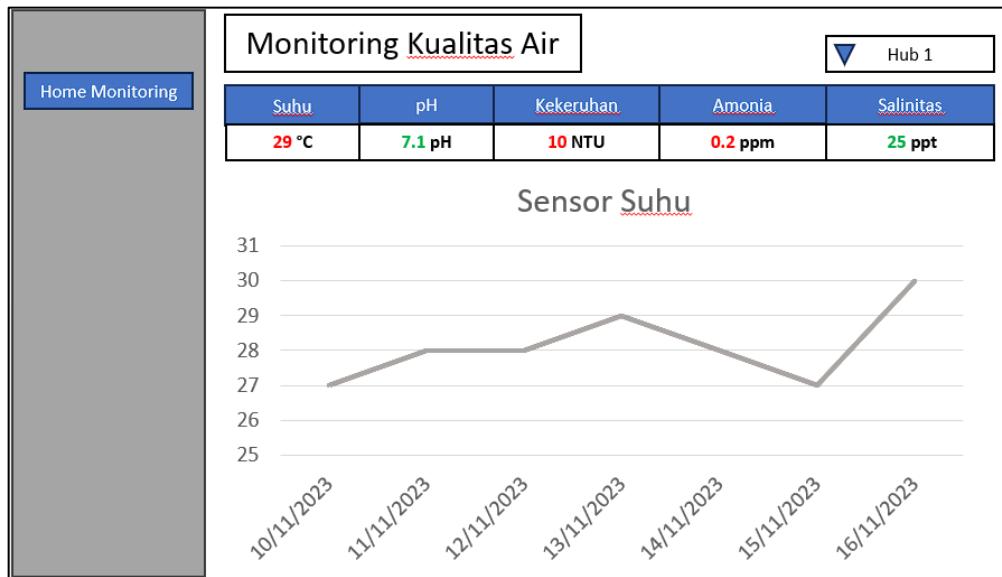
3.4.6 Framework Laravel

Prototipe perangkat yang sudah dirancang memerlukan sebuah media luaran untuk menampilkan hasil data yang sudah terekam di dalam *database*. Media yang digunakan untuk menampilkan hasil data yang sudah tersimpan di dalam *database* adalah dengan menggunakan aplikasi berbasis web menggunakan bahasa pemrograman PHP dan menggunakan sebuah *framework* bernama *laravel*. Kegunaan dari *laravel* adalah memudahkan untuk membuat lingkungan kerja dalam menyambungkan database ke halaman *web* menjadi lebih ringkas dan mengurangi adanya *downtime* atau kurang akuratnya pembacaan data yang terdapat di dalam *database*.

3.4.7 Tampilan Halaman *Web*

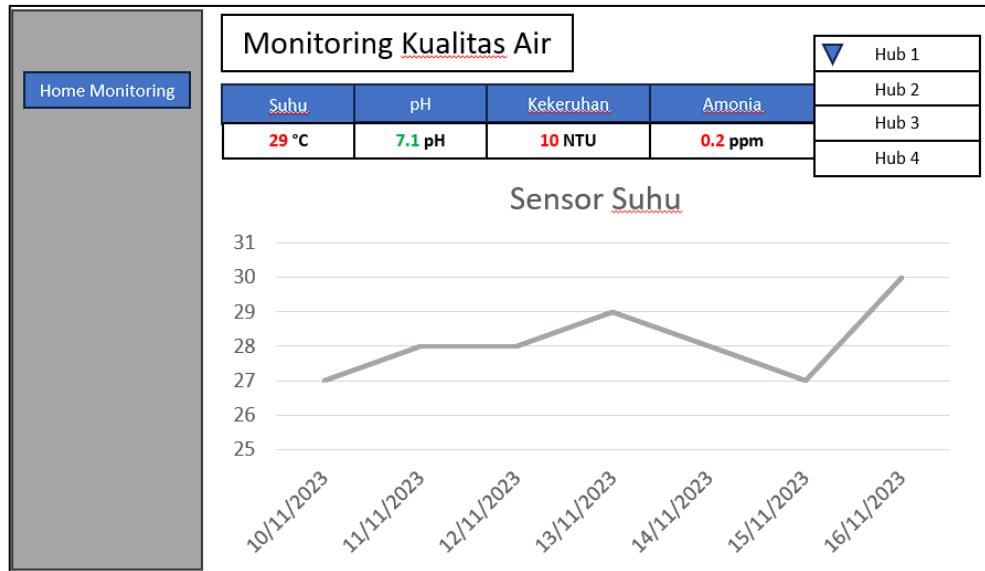
Data kualitas air yang sudah dibaca dan dikumpulkan oleh sensor dan ditampung di dalam *database*, kemudian ditampilkan menjadi luaran berupa sistem monitoring menggunakan *platform* aplikasi berupa halaman *web*. Halaman *web*

berfungsi menampilkan data-data dari sensor yang melakukan pengukuran di lapangan secara *realtime*. Pada halaman utama pada aplikasi *web* yang diilustrasikan pada Gambar 3.7, pembacaan data pada sensor diberi luaran berupa pembacaan angka dan satuannya, lalu diberi parameter warna merah yang mengindikasikan bahwa terjadi penurunan atau peningkatan kualitas air melewati batas bawah maupun batas atas. Parameter angka diberi warna hijau yang mengindikasikan bahwa kualitas air mempunyai batas normal.



Gambar 3. 7 Rancangan Tampilan Utama Halaman Web.

Pemberian grafik pada halaman *web* bertujuan untuk memberi informasi mengenai kenaikan atau penurunan dalam waktu harian. Grafik memberikan informasi data harian berupa perhitungan rata-rata dari sensor yang sudah melakukan pengukuran pada kurun waktu harian, sehingga memudahkan pengguna untuk melacak apabila terdapat anomali dalam kolam tambak.



Gambar 3. 8 Halaman Web Menampilkan Kumpulan ESP32.

Pada Gambar 3.8, diberikan sebuah fitur *dropdown* untuk memberikan pilihan untuk menampilkan *ESP32* yang terpasang pada kolam tambak yang lain

3.5 Pengujian Prototipe

Rancangan Pengujian untuk prototipe dibagi menjadi beberapa pengujian, yakni pengujian kelayakan perangkat dan pengujian menggunakan air laut tiruan.

3.5.1 Pengujian Pembanding

Untuk menguji keakuratan sensor dalam melakukan pengukuran untuk mendapatkan data-data kualitas air, diperlukan alat pembanding yang sudah teruji untuk melakukan pengukuran dalam lingkup industrial dan laboratorium. Dalam penelitian ini, alat yang digunakan adalah LAQUA PH2000 yang mempunyai spesifikasi sebagai berikut:

- Rentang Pengukuran pH: -2.000 hingga 20.000 pH.
- Resolusi pH: 0.1 / 0.01 / 0.001 pH.
- Akurasi pH: ± 0.003 pH.
- Rentang Pengukuran Suhu: -30.0 hingga 130.0 °C / -22.0 hingga 266.0 °F.
- Resolusi Suhu: 0.1 °C.

- Akurasi Suhu: $\pm 0.5^{\circ}\text{C}$ / $\pm 0.9^{\circ}\text{F}$.
- Memori Internal: Menyimpan hingga 2000 set data.
- Antarmuka Komunikasi: RS-232 / USB untuk koneksi PC dan *printer*.
- Tampilan Layar: LCD 5 inci dengan *backlight*.
- Indikator Status Elektroda: Menunjukkan kondisi elektroda berdasarkan data kalibrasi.
- Mode Pengukuran: *Auto Stable, Auto Hold, Real Time*.
- Stand Elektroda: Fleksibel, bisa menampung hingga 2 elektroda, berputar 360° dan bisa digerakkan naik turun.

Perangkat berikutnya adalah perangkat yang bernama LOVIBOND TB 211 IR yang mempunyai spesifikasi sebagai berikut:

- Rentang Pengukuran: 0.01 hingga 1100 NTU (*Nephelometric Turbidity Units*)
- Prinsip Pengukuran: *Nephelometric* (pengukuran cahaya tersebar pada sudut 90°)
- Akurasi: $\pm 2.5\%$ dari bacaan atau ± 0.01 NTU, yang mana lebih besar, dalam rentang 0.01 hingga 500 NTU dan $\pm 5\%$ dari bacaan dalam rentang 500 hingga 1000 NTU
- Tampilan: LCD dengan lampu latar
- Memori Internal: Kapasitas penyimpanan hingga 125 set data
- Kompatibilitas Standar: Memenuhi standar EN ISO 7027

3.5.2 Pengujian Kelayakan Perangkat

Prototipe alat mempunyai kumpulan kabel dan sambungan pin yang saling terhubung. Pengujian dilakukan dengan menggunakan aplikasi *Arduino IDE* sebagai media untuk melakukan *troubleshooting* pada prototipe alat. *Troubleshooting* dilakukan untuk melihat adanya sambungan kabel yang tidak sesuai dengan pin *GPIO* pada mikrokontroller *ESP32* dengan melihat rekam data yang tampil di *serial monitor* pada aplikasi *Arduino IDE* berupa data yang mempunyai nilai yang lebih kecil atau rendah dari parameter, data dengan rekam hasil yang sama, atau data tidak menampilkan nilai pada saat pengukuran.

Pengujian fungsi perangkat ditunjukkan kepada pengguna dengan melihat beberapa aspek kelayakan dalam menggantikan alat-alat pengukuran konvesional dan mengubah pola penggunaan dalam melakukan pengukuran. Uji Kelayakan dilakukan dengan melakukan penyebaran angket kepada pengguna untuk melihat tolok ukur kelayakan prototipe alat dengan menilai dari kecepatan akses data dalam mengakses halaman *web*.

3.5.3 Pengujian Menggunakan Air Laut Tiruan

Pengujian perangkat dilakukan dengan menggunakan sebagai air laut tiruan sebagai tolok ukur bagaimana alat mampu bekerja sebagaimana mestinya dan dapat menampilkan data yang sudah terekam secara akurat melalui tampilan *output* berupa aplikasi *web*.

BAB IV

HASIL DAN PEMBAHASAN

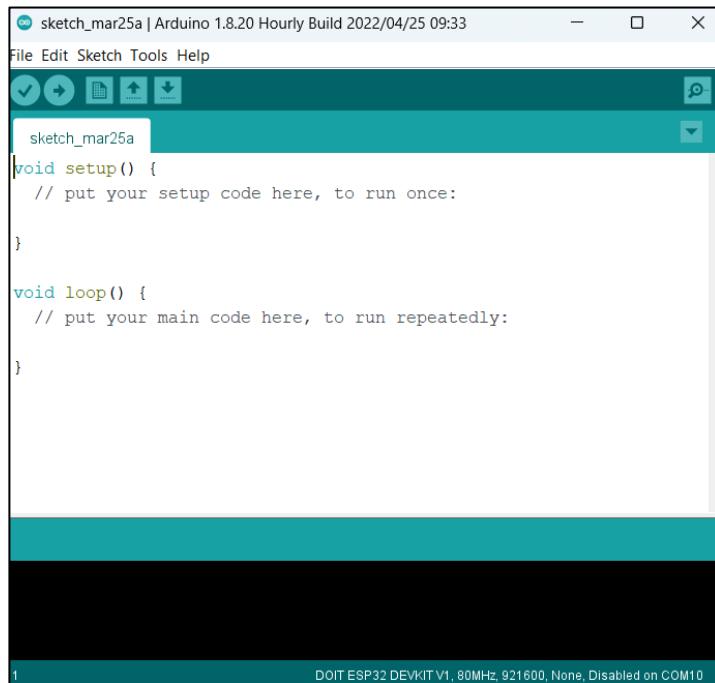
4.1 Alat / Komponen

Pengembangan alat ukur kualitas air berbasis IoT dimulai dengan menyiapkan kumpulan *hardware* yang diperlukan dan dilakukan proses *interfacing* atau pada dasarnya menghubungkan perangkat komputer dengan perangkat sensor-sensor untuk mengetahui nilai yang dibaca dari sensor yang digunakan. Berikut merupakan *hardware* yang digunakan untuk keperluan *interfacing*, yaitu:

1. Mikrokontroler *DOIT ESP32 DevKit V1*.
2. Sensor Suhu DS18B20.
3. Sensor pH 4502C.
4. Sensor Amonia MQ-135.
5. Sensor Kekaruan SEN-0189.
6. Sensor TDS KS0429.
7. Resistor 47k ohm.
8. Kabel *Jumper Male to Male* dan *Male to Female*.
9. *Laptop* dengan spesifikasi sebagai berikut:
 - *Processor Ryzen 7 5800U*
 - 16GB *RAM*
 - 512GB *NVMe*
 - 1 Buah Port *USB-A*
10. Kabel Data *USB-A* ke *Micro USB* sebagai sambungan daya sementara dan input program untuk mikrokontroler.
11. *Breadboard*.

Interfacing terhadap sensor dan mikrokontroler juga membutuhkan sebuah *software* atau program yang bernama *arduino IDE*. *Arduino IDE* adalah sebuah *software* yang menggunakan bahasa C++ sebagai dasarnya yang berfungsi untuk menanamkan sebuah program ke dalam mikrokontroler melalui laptop yang dihubungkan dengan kabel data, sehingga mikrokontroler dapat membaca data input

dari semua sensor yang dihubungkan dengan mikrontoler. Berikut adalah tampilan awal dari program *arduino IDE* yang dapat dilihat pada gambar 4.1.



Gambar 4. 1 Tampilan Awal Arduino IDE.

4.2 Kalibrasi Sensor

Sebelum seluruh perangkat sensor disusun menjadi satu perangkat, terlebih dahulu dilakukan kalibrasi pada tiap sensor dengan tujuan untuk mengetahui apakah sensor bekerja dengan baik dalam membaca data sesuai dengan fungsinya dan mengetahui batasan dari tiap sensor ketika melakukan pengukuran yang tidak sesuai dengan kemampuan perangkat. Pembacaan data dari sensor diberikan jarak waktu atau *interval* sebesar 2 detik per data sensor yang terekam. Rangkaian koneksi sensor dan mikrontroler dapat dilihat kembali pada gambar 3.4.

4.2.1 Kalibrasi Sensor pH

Sensor pH yang digunakan merupakan Sensor pH bernama pH 4502C, yang terdiri dari sebuah *probe* pada gambar 4.2, yang digunakan untuk membaca pH cairan dan sebuah *controller board* yang digunakan untuk menerjemahkan kelistrikan yang ada pada *probe* pH menjadi sinyal analog yang kemudian akan dibaca pada program *arduino IDE* sebagai data luaran.



Gambar 4. 2 Komponen Untuk Sensor pH 4502C.

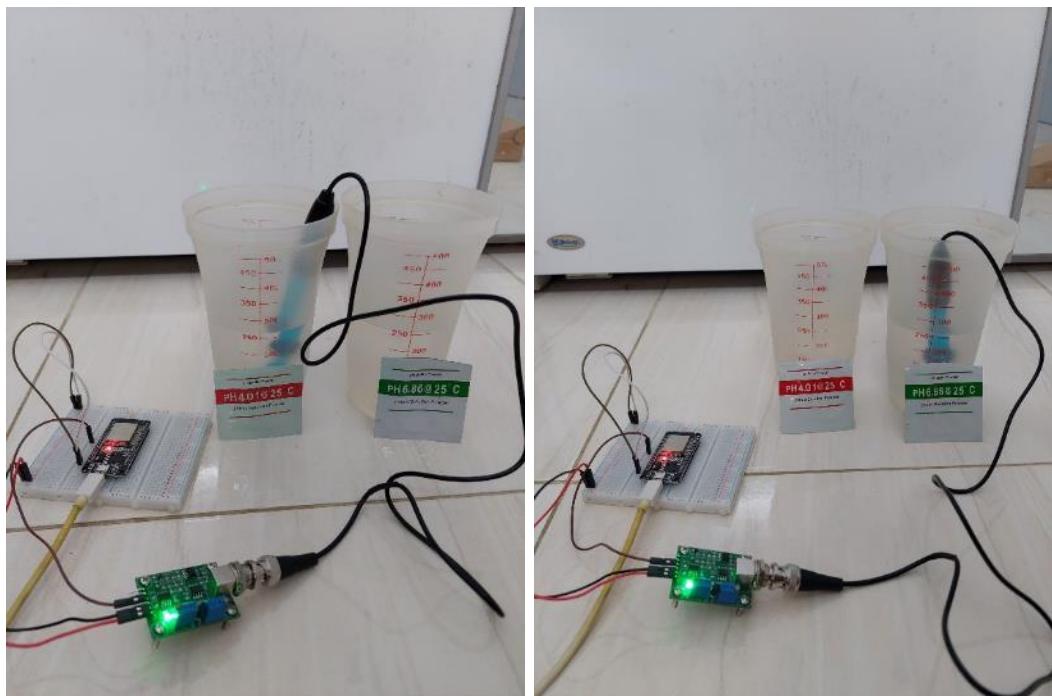
Probe pada sensor pH merupakan bagian yang sensitif sehingga ketika selesai melakukan pengukuran, harus segera dibersihkan dengan cairan *buffer solution* pada gambar 4.3, yang digunakan untuk menetralkan *probe* untuk pembacaan selanjutnya. Sensor pH hanya tahan pada cairan untuk *probe*-nya saja, tidak untuk *board controller*-nya.



Gambar 4. 3 Cairan Buffer Solution Untuk Menetralkan Probe Sensor pH.

Kalibrasi untuk sensor pH dilakukan dengan menggunakan 2 cairan yang mempunyai kandungan pH 4 dan pH 7 pada gambar 4.4. Cairan yang mempunyai

kandungan pH merupakan air minum dengan takaran 250ml yang sudah dicampur oleh bubuk pH *buffer solution* untuk mensimulasikan cairan yang mengandung pH 4 dan pH 7.



Gambar 4. 4 Pembacaan sensor pH untuk cairan pH 7 (Kiri) dan cairan pH 4 (Kanan)

Cara kerja *probe* pada sensor pH adalah dengan mengukur nilai *Analog to Digital Converter* (ADC) dari cairan yang sudah dicelupkan oleh *probe*, dan untuk pembacaan nilai ADC dapat dilakukan dengan menghubungkan sensor pH dengan *ESP32* yang kemudian akan dihubungkan dengan *laptop* dan ditanamkan program menggunakan *arduino IDE* untuk melihat hasil dari nilai analog yang terbaca pada *probe* yang sudah dicelupkan ke cairan menggunakan *serial monitor* yang ada pada program *arduino IDE*. Nilai ADC yang sudah terbaca, kemudian akan dihitung menggunakan persamaan berikut:

$$\text{tegangan pH} = \text{nilai ADC pH} \times \left(\frac{3.3}{4095} \right) \quad 2-1$$

Nilai tegangan pH yang sudah didapat, kemudian akan dibandingkan melalui cairan yang mempunyai kandungan pH 4 dan pH 7 yang dapat dilihat pada gambar 4.4. Masing-masing cairan yang dengan kandungan pH 4 dan pH 7, mempunyai

tegangan yang berbeda dan untuk mendapatkan nilai pH yang sesungguhnya dari sensor dengan menggunakan persamaan berikut:

$$pH\ step = \frac{(tegangan\ pH\ 4 - tegangan\ pH\ 7)}{(7 - 4)} \quad (4-1)$$

Kemudian saat pH step sudah didapatkan nilainya, dilanjutkan dengan persamaan berikut:

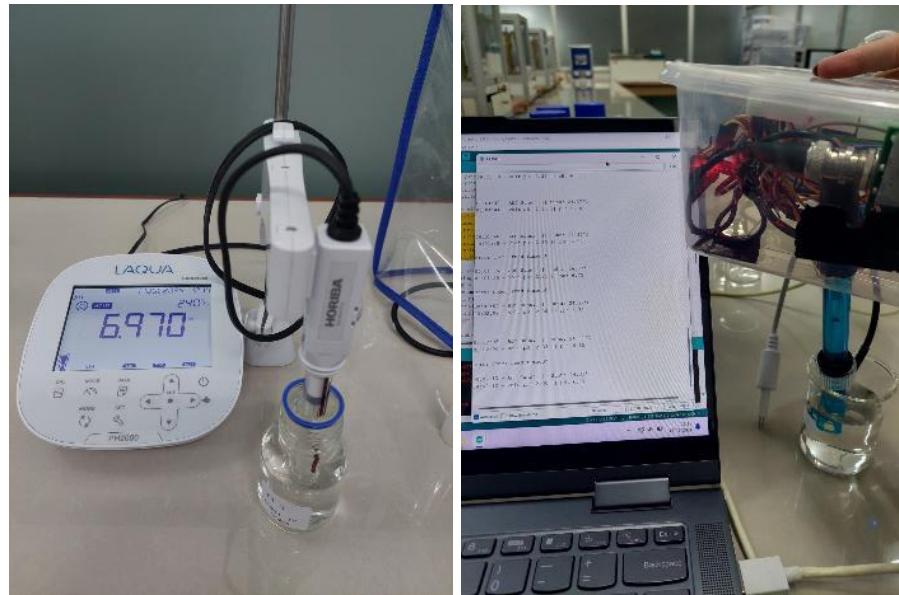
$$pH\ netral = 7.00 + \left(\frac{(tegangan\ pH\ 7 - tegangan\ pH)}{pH\ step} \right) \quad (4-2)$$

Hasil dari perhitungan rumus dapat dilihat pada gambar 4.5 ketika sensor membaca pada cairan yang mengandung pH 4 dan pH 7 menggunakan program *arduino IDE* yang ditampilkan dalam *serial monitor*.

00:00:19 - Volt pH: 2.83 pH: 4.02	00:00:35 - Volt pH: 2.24 pH: 7.06
00:00:21 - Volt pH: 2.81 pH: 4.06	00:00:37 - Volt pH: 2.26 pH: 6.96
00:00:23 - Volt pH: 2.84 pH: 3.97	00:00:39 - Volt pH: 2.27 pH: 6.90
00:00:25 - Volt pH: 2.85 pH: 3.97	00:00:41 - Volt pH: 2.27 pH: 6.95
00:00:27 - Volt pH: 2.82 pH: 4.07	00:00:43 - Volt pH: 2.28 pH: 6.87
00:00:29 - Volt pH: 2.85 pH: 3.98	00:00:45 - Volt pH: 2.27 pH: 6.94
00:00:31 - Volt pH: 2.85 pH: 4.00	00:00:47 - Volt pH: 2.21 pH: 7.22
00:00:33 - Volt pH: 2.85 pH: 3.97	

Gambar 4.5 Hasil Kalibrasi dari cairan pH 4 (Kiri) dan cairan pH 7 (Kanan)

Untuk mengetahui sensor pH dapat mengukur kandungan pH cairan secara akurat, sensor pH dibandingkan menggunakan alat ukur dengan standar laboratorium untuk membandingkan keakuratan sensor dalam membaca kandungan pH dalam cairan. Dalam pengujian perbandingan, digunakan cairan dengan kandungan pH sebesar pH 7 dan sensor pH serta alat ukur digunakan secara bersamaan untuk melihat hasil dari masing-masing sensor dan alat untuk dapat melakukan pembacaan kandungan pH terhadap pH dengan kandungan pH 7.



Gambar 4. 6 LAQUA PH2000 (Kiri) dan Prototipe Alat Sensor pH (Kanan)

Pada gambar 4. 6, alat ukur pH dapat membaca nilai kandungan pH 7 dengan akurat, dan pada sensor pH yang sudah di kalibrasi, sensor dapat membaca kandungan pH dengan nilai yang cukup sebanding dengan LAQUA PH2000 yang digunakan sebagai perbandingan.

Tabel 4. 1 Perbandingan Data pH Dengan Interval Waktu 2 Detik (2000 *Milliseconds*).

No.	LAQUA PH2000 (pH)	Sensor pH 4502C (pH)
1	6,97	6,58
2	6,97	6,75
3	6,97	6,57
4	6,97	6,75
5	6,97	6,75
6	6,97	6,65
Rata-Rata	6,97	6,67
Standar Deviasi	0	0,086

Dapat di simpulkan dari hasil pembacaan pada gambar 4.5, didapatkan bahwa sensor dapat membaca nilai pH 7 dengan rata-rata sebesar 6,67 dan standar

deviasi sebesar 0,086, yang dapat di simpulkan bahwa sensor pH dapat membaca nilai pH sesuai dengan fungsinya dan mempunyai keakuratan yang cukup dekat dengan LAQUA PH2000, dan untuk pH kurang dari atau sama dengan 7, sensor dapat membaca data nilai pH dengan akurat.

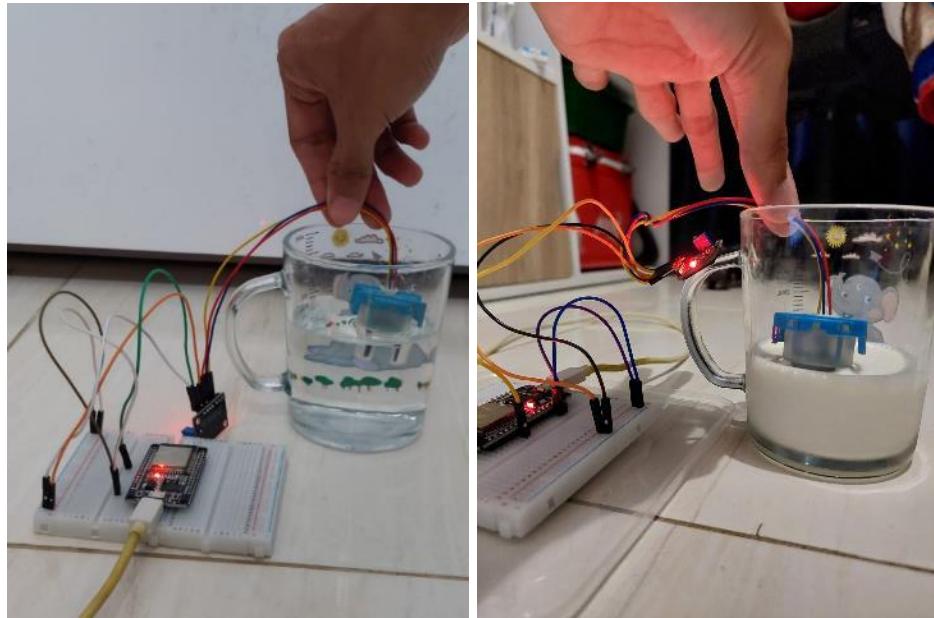
4.2.2 Kalibrasi Sensor Kekeruhan

Sensor kekeruhan yang digunakan mempunyai tipe yang bernama SEN-0189 pada gambar 4.6, yang terdiri dari sebuah *probe* yang mempunyai pengait di kanan dan kiri dikarenakan *probe* dari sensor kekeruhan tidak sepenuhnya dapat dicelupkan ke dalam cairan, sehingga butuh dikaitkan oleh mulut wadah atau gelas. SEN-0189 juga dilengkapi dengan *board controller* yang digunakan untuk menghubungkan sensor dengan mikrontroler untuk dilakukan pembacaan data.



Gambar 4. 7 Komponen Sensor Kekeruhan SEN-0189.

Cara kerja dari *probe* sensor kekeruhan adalah dengan mengetahui nilai ADC terlebih dahulu dengan menghubungkan sensor dengan mikrontroler dan mencelupkan *probe* sensor ke dalam cairan berupa air bersih yang di ilustrasikan pada gambar 4.7 untuk dibaca ADC-nya menggunakan *arduino IDE*. Untuk mensimulasikan air keruh, percobaan menggunakan cairan berupa air susu untuk membaca data nilai keruh pada sensor kekeruhan.



Gambar 4. 8 Probe Dicelupkan Air Bersih (Kiri) dan Probe Dicelupkan Air Susu (Kanan).

Nilai ADC sensor didapatkan dengan mencelupkan *probe* pada air bersih dan didapatkan nilai sebesar 1245 sebagai acuan untuk pembacaan nilai kekeruhan pada air bersih. Kemudian nilai ADC yang sudah didapat kemudian diterapkan pada fungsi *map* pada Arduino IDE untuk menetapkan batas bawah dan batas atas nilai kekeruhan yang ingin dibaca.

```
float readKeruh() {
    valueTurbidity = analogRead(turbidity_pin);
    float turbidity = map(valueTurbidity, 0, 1245, 50, 1);
    return turbidity;
}
```

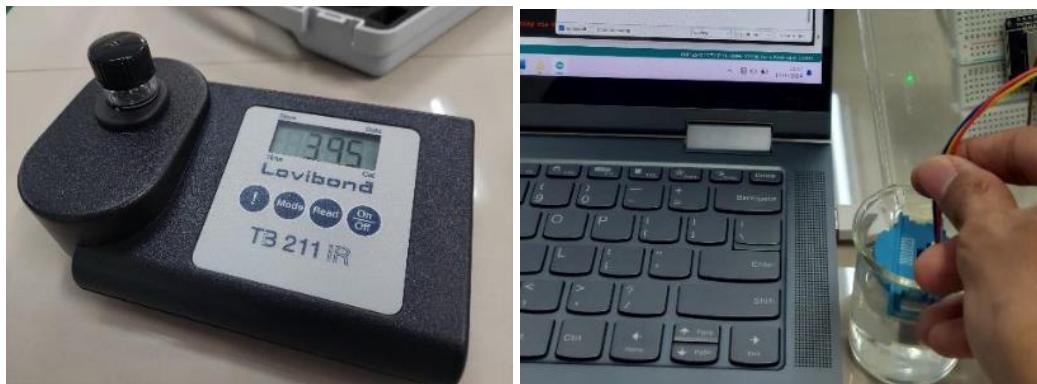
Gambar 4. 9 Fungsi Pembacaan Sensor Keruh.

Air yang sudah terukur menggunakan sensor dapat diatur di dalam program dengan memberi nilai 1 untuk nilai minimum bahwa air semakin jernih dan nilai 50 untuk nilai maksimum jika air semakin keruh. Hasil dari pengukuran air jernih dapat dilihat pada gambar 4.8 yang ditampilkan pada *serial monitor* dalam program *arduino IDE*.

00:01:57 - ADC Keruh: 1098 Kekeruhan: 2.00 NTU	00:04:11 - ADC Keruh: 0 Kekeruhan: 50.00 NTU
00:01:59 - ADC Keruh: 1133 Kekeruhan: 0.00 NTU	00:04:13 - ADC Keruh: 0 Kekeruhan: 50.00 NTU
00:02:01 - ADC Keruh: 1119 Kekeruhan: 2.00 NTU	00:04:15 - ADC Keruh: 0 Kekeruhan: 50.00 NTU
00:02:03 - ADC Keruh: 1147 Kekeruhan: 0.00 NTU	00:04:17 - ADC Keruh: 0 Kekeruhan: 50.00 NTU
00:02:05 - ADC Keruh: 1141 Kekeruhan: 1.00 NTU	00:04:19 - ADC Keruh: 0 Kekeruhan: 50.00 NTU
00:02:07 - ADC Keruh: 1130 Kekeruhan: 0.00 NTU	00:04:21 - ADC Keruh: 0 Kekeruhan: 50.00 NTU
00:02:09 - ADC Keruh: 1163 Kekeruhan: 0.00 NTU	00:04:23 - ADC Keruh: 0 Kekeruhan: 50.00 NTU

Gambar 4. 10 Hasil Pengukuran Air Bersih (Kiri) dan Hasil Pengukuran Simulasi Air Keruh (Kanan).

Untuk dapat menentukan nilai kekeruhan secara akurat, dilakukan pengujian dengan menggunakan LOVIBOND TB 211 IR untuk mengukur nilai kekeruhan NTU pada cairan berupa air bersih.



Gambar 4. 11 TB 211 IR (Kiri) dan Sensor Kekeruhan SEN-0189 (Kanan).

TB 211 IR dapat mengukur air bersih dengan nilai kekeruhan seperti pada gambar 4. 12 dengan nilai sebesar 3,95 NTU, sedangkan sensor kekeruhan yang sudah terkalibrasi dapat mengukur air bersih dengan kisaran nilai 1-2 NTU.

Tabel 4. 2 Perbandingan Data Keruh Dengan Interval Waktu 2 Detik (2000 Milliseconds).

No.	TB 211 IR (NTU)	Sensor SEN-0189 (NTU)
1	3,95	2,00
2	3,95	2,00

Tabel 4. 2 Perbandingan Data Keruh Dengan Interval Waktu 2 Detik (2000 Milliseconds) (Lajutan).

No.	TB 211 IR	Sensor SEN-0189
3	3,95	2,00
4	3,95	1,00
5	3,95	1,00
6	3,95	1,00
Rata-Rata	3,95	1,5
Standar Deviasi	0	0,54

Dapat disimpulkan bahwa sensor kekeruhan dengan tipe SEN-0189 dapat melakukan pengukuran sesuai fungsinya, namun terdapat perbedaan rentang nilai yang cukup signifikan dari alat ukur laboratorium sehingga diperlukan penelitian lebih lanjut mengenai hal tersebut.

4.2.3 Kalibrasi Sensor TDS

Sensor TDS (*Total Dissolved Solids*) yang digunakan mempunyai tipe yang bernama KS0429 yang terdiri dari beberapa komponen pada gambar 4.10. seperti *probe* tahan air yang dapat dicelupkan ke dalam cairan dan sebuah *board controller* yang berfungsi untuk menghubungkan sensor dengan mikrokontroler untuk dilakukan pembacaan data.



Gambar 4. 12 Komponen Sensor TDS KS0429.

Cara kerja dari sensor TDS adalah dengan mencelupkan *probe* ke dalam cairan seperti pada gambar 4.10. Cairan yang digunakan adalah campuran air bersih yang dicampurkan dengan satu sendok garam, yang bertujuan untuk mensimulasikan kandungan benda pada yang terlarut di dalam cairan tersebut.



Gambar 4. 13 Probe Sensor TDS Dicelupkan Ke Air Garam.

Pengukuran nilai sensor dilakukan dengan menghubungkan *board controller* sensor pada mikrokontroler dan kemudian dihubungkan ke perangkat laptop untuk ditanamkan sebuah program ke dalam sensor menggunakan *arduino IDE*. Pengukuran data pada sensor TDS dimudahkan dengan bantuan *library* yang sudah ada bernama *GravityTDS* dan digunakan di dalam program tersebut, hanya saja nilai hasil dari *library* yang sudah digunakan perlu dimodifikasi dengan mengkalikan nilai 100 dikarenakan perhitungan pada *library* hanya menghasilkan nilai dua angka dibelakang koma. Hasil pengukuran air garam yang sudah ditampilkan di dalam *serial monitor arduino IDE* dapat dilihat pada gambar 4.11.

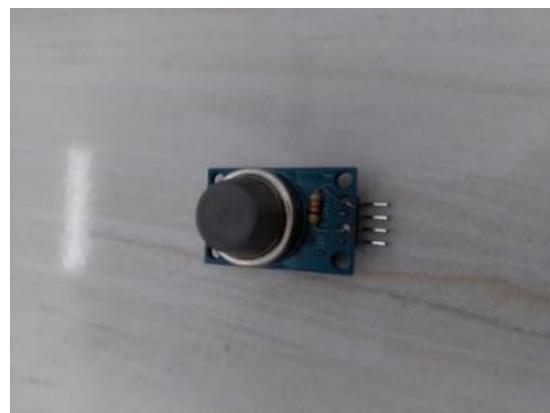
```
00:00:05 - ADC TDS: 3583 || TDS: 34.1 ppt
00:00:07 - ADC TDS: 3595 || TDS: 33.5 ppt
00:00:09 - ADC TDS: 3600 || TDS: 34.2 ppt
00:00:11 - ADC TDS: 3632 || TDS: 34.0 ppt
00:00:13 - ADC TDS: 3613 || TDS: 35.3 ppt
00:00:15 - ADC TDS: 3573 || TDS: 34.5 ppt
00:00:17 - ADC TDS: 3637 || TDS: 34.5 ppt
```

Gambar 4. 14 Hasil Dari Pembacaan Air Garam Menggunakan Sensor TDS.

Dapat disimpulkan bahwa sensor TDS KS0429 dapat melakukan pengukuran terhadap benda padat yang larut dalam cairan dengan akurat sesuai dengan fungsi perangkat.

4.2.4 Kalibrasi Sensor Gas

Sensor amonia yang digunakan mempunyai tipe yang bernama MQ-135. Sensor dilengkapi dengan *board controller* yang sudah dilengkapi dengan modul pada gambar 4.12 yang berfungsi untuk membaca kandungan gas berbahaya yang ada pada udara.



Gambar 4. 15 Sensor Gas MQ-135.

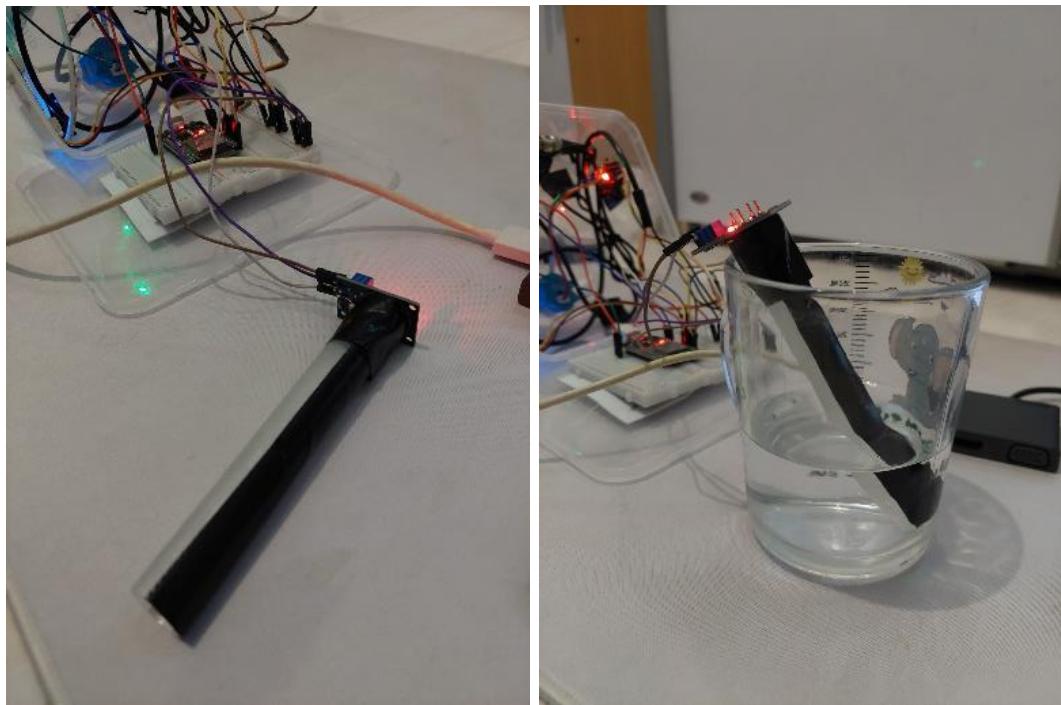
Modul yang ada pada sensor amonia tidak tahan air, sehingga untuk mengukur kadar amonia pada suatu cairan, modul harus diletakkan pada permukaan cairan sehingga modul akan mendeteksi kandungan amonia yang dihasilkan oleh cairan.

Untuk menambah keakuratan dalam pengukuran kandungan amonia, *probe* sensor diberikan sebuah tabung atau corong untuk memfokuskan *probe* dalam mengukur nilai amonia pada permukaan air dan tidak terdistrupsi oleh pengukuran gas amonia secara luas karena *probe* yang dapat mengukur kandungan gas dari segala arah.



Gambar 4. 16 Corong Tambahan Sensor Gas.

Kalibrasi dari sensor gas dilakukan dengan menghubungkan sensor dengan mikrontroler dan selanjutnya dihubungkan dengan perangkat laptop untuk ditanamkan program menggunakan *arduino IDE* ke dalam mikrontroler.



Gambar 4. 17 Sensor Gas Pada Udara Terbuka (Kiri) dan Dicelupkan Pada Air Bersih (Kanan).

Langkah awal adalah dengan mencari tegangan dari sensor dengan mengetahui nilai ADC ketika sensor mendeteksi adanya kandungan gas pada udara .

$$\text{tegangan sensor} = \text{nilai ADC} \times \left(\frac{3.3}{4095} \right) \quad (4- 3)$$

Nilai tegangan yang sudah didapatkan kemudian dilanjutkan dengan penggunaan rangkaian rumus, diawali dengan persamaan berikut:

$$RS = \left(\left(\frac{3.3}{\text{tegangan sensor}} \right) - 1 \right) \times 4.7 \quad (4- 4)$$

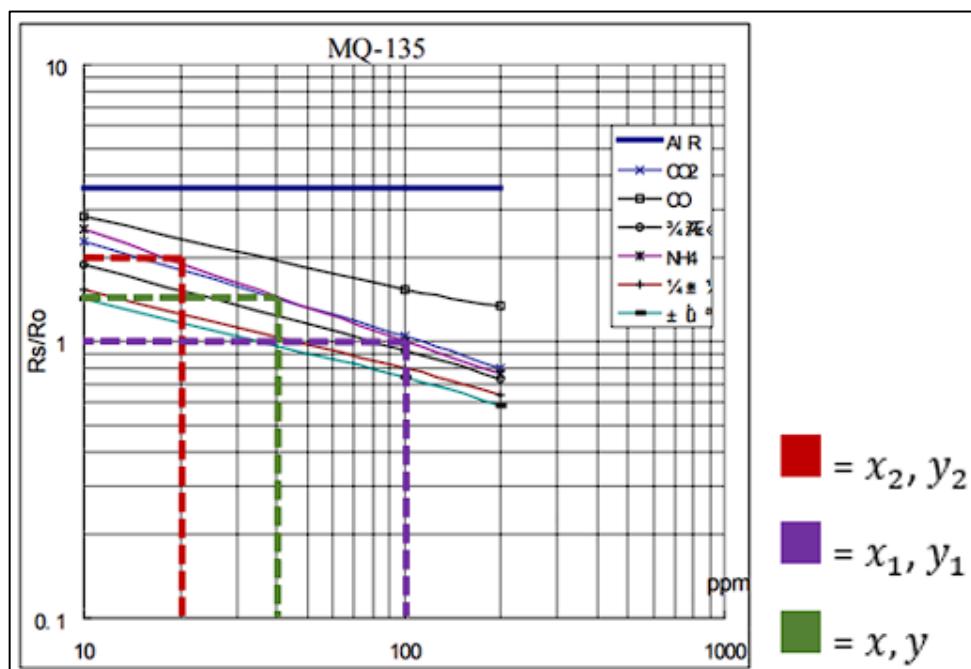
Nilai RL berasal dari tegangan resistor yang digunakan pada sensor amonia, yaitu menggunakan resistor sebesar 47k Ohm yang berfungsi untuk menambah sensitivitas dalam pengukuran nilai gas yang diinginkan. Nilai RO didapatkan dengan menggunakan persamaan berikut:

$$RO = \frac{RS}{3.6} \quad (4- 5)$$

Setelah mendapatkan nilai RO baru, kemudian mencari ratio tegangan menggunakan persamaan berikut:

$$ratio = \frac{RS}{RO} \quad (4- 6)$$

Untuk menghitung nilai NH4 (Amonia) yang diukur pada sensor, hal yang harus dilakukan adalah mencari nilai m, yaitu garis yang bersinggungan dengan garis NH4 pada gambar 4. 14, dan didapatkan nilai untuk $x_1 = 100, x_2 = 19, y_1 = 1$, dan $y_2 = 2$.



Gambar 4. 18 Grafik Perhitungan NH4 (Amonia).

Nilai x dan y yang sudah didapatkan kemudian diimplementasikan menggunakan berikut untuk mendapatkan nilai m:

$$m = \frac{[\log(y_2) - \log(y_1)]}{[\log(x_2) - \log(x_1)]} \quad (4- 7)$$

Tahap selanjutnya adalah mencari nilai b, yaitu nilai tengah dari garis NH4 pada grafik dengan menggunakan persamaan berikut:

$$b = \log(y) - m \times \log(x) \quad (4- 8)$$

Setelah mendapatkan nilai m dan b, untuk mencari nilai PPM pada NH4 dapat digunakan persamaan berikut:

$$PPM = 10^{\left\{ \left[\frac{\log(ratio) - b}{m} \right] \right\}} \quad (4-9)$$

Nilai hasil PPM sudah dapat mengukur kandungan amonia dalam udara, hanya saja perlu dilakukan modifikasi dengan membagi nilai hasil PPM dengan nilai 100000 untuk mencocokan dengan standar kualitas air tambak yang diperlukan.

00:09:34 - ADC Amonia: 2880 Ammonia: 0.0115 ppm	00:13:10 - ADC Amonia: 2677 Ammonia: 0.0088 ppm
00:09:36 - ADC Amonia: 2896 Ammonia: 0.0126 ppm	00:13:12 - ADC Amonia: 2720 Ammonia: 0.0087 ppm
00:09:38 - ADC Amonia: 2877 Ammonia: 0.0126 ppm	00:13:14 - ADC Amonia: 2722 Ammonia: 0.0076 ppm
00:09:40 - ADC Amonia: 2879 Ammonia: 0.0129 ppm	00:13:16 - ADC Amonia: 2686 Ammonia: 0.0076 ppm
00:09:42 - ADC Amonia: 2825 Ammonia: 0.0134 ppm	00:13:18 - ADC Amonia: 2719 Ammonia: 0.0088 ppm
00:09:44 - ADC Amonia: 2869 Ammonia: 0.0114 ppm	00:13:20 - ADC Amonia: 2725 Ammonia: 0.0087 ppm
00:09:46 - ADC Amonia: 2819 Ammonia: 0.0113 ppm	00:13:22 - ADC Amonia: 2706 Ammonia: 0.0081 ppm

Gambar 4. 19 Hasil Pengukuran Amonia Udara Terbuka (Kiri) dan Permukaan Air Bersih (Kanan).

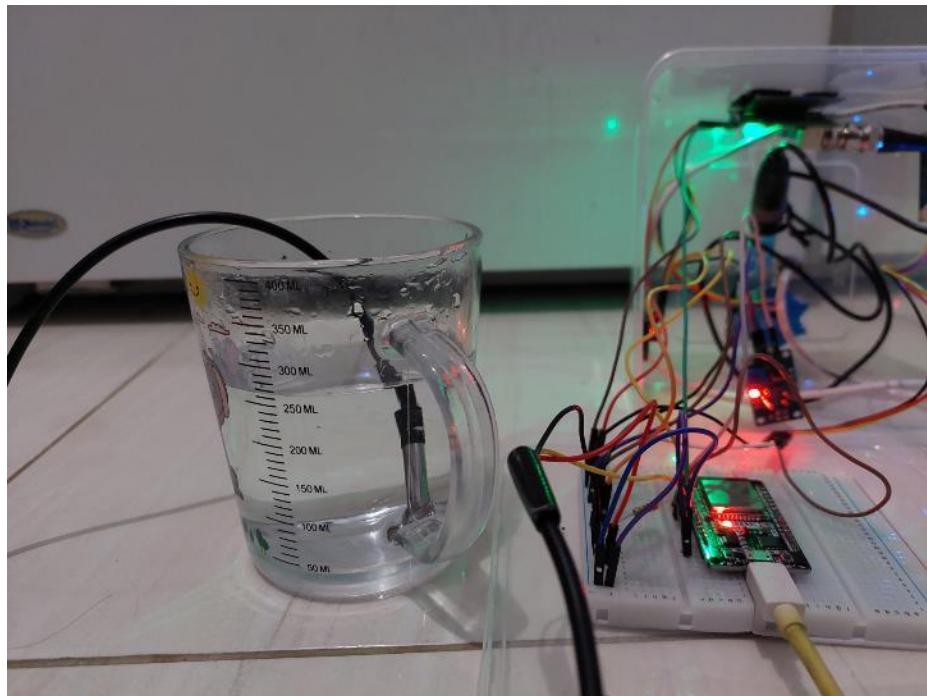
4.2.5 Kalibrasi Sensor Suhu

Sensor suhu yang digunakan merupakan sensor suhu dengan nomor seri DS18B20 yang mempunyai spesifikasi berupa ketahanan jika dicelupkan ke dalam air atau melakukan pengukuran ke dalam suatu cairan untuk mengetahui kondisi suhu dari cairan tersebut, dapat juga melakukan pengukuran suhu pada ruangan maupun benda yang menempel pada *probe* sensor.



Gambar 4. 20 Sensor Suhu DS18B20 (randomnerdtutorials.com).

Kalibrasi sensor suhu pada gambar 4.17 dilakukan dengan mencelupkan *probe* sensor pada air kran bersih. Sensor suhu mempunyai 3 kabel yang digunakan untuk menghubungkan sensor dengan mikrokontroler, dan kabel pada sensor suhu dilakukan modifikasi dengan menggabungkannya dengan kabel *jumper* sehingga sensor suhu dapat terhubung dengan mikrokontroler. Perangkat sensor pada gambar 4.15 tidak mempunyai kontroler board, sehingga diperlukan resistor 47k ohm yang berfungsi untuk mentoleransi tegangan untuk sensor dapat membaca nilai suhu.



Gambar 4. 21 Kalibrasi Sensor Suhu Menggunakan Air Bersih.

Pengukuran nilai sensor dilakukan dengan menghubungkan 3 kabel sensor pada mikrokontroler sesuai dengan *port* dan kemudian dihubungkan ke perangkat laptop untuk ditanamkan sebuah program ke dalam sensor menggunakan *arduino IDE*. Pengukuran data pada sensor suhu dimudahkan dengan bantuan *library* yang sudah ada bernama *DallasTemperature* dan digunakan di dalam program tersebut. Hasil pengukuran air kran bersih yang sudah ditampilkan di dalam *serial monitor arduino IDE* dapat dilihat pada gambar 4.18.

```

00:00:42 - ADC Suhu: || Suhu: 28.94°C
00:00:44 - ADC Suhu: || Suhu: 28.94°C
00:00:47 - ADC Suhu: || Suhu: 28.94°C
00:00:49 - ADC Suhu: || Suhu: 28.94°C
00:00:51 - ADC Suhu: || Suhu: 28.94°C
00:00:53 - ADC Suhu: || Suhu: 29.00°C
00:00:55 - ADC Suhu: || Suhu: 29.00°C

```

Gambar 4. 22 Hasil Pengukuran Suhu Pada Air Bersih.

Untuk mengetahui nilai suhu pada cairan dengan lebih akurat, sensor suhu dibandingkan dengan alat ukur suhu yang sudah terintegrasi dengan LAQUA PH2000 pada gambar 4. 6. Sensor suhu dan alat ukur suhu dicelupkan pada cairan berupa air bersih yang terlihat pada gambar 4.6.

Tabel 4. 3 Perbandingan Data Suhu Dengan Interval Waktu 2 Detik (2000 Milliseconds).

No.	LAQUA PH2000 (°C)	Sensor Suhu DS18B20 (°C)
1	24,0	24,37
2	24,0	24,37
3	24,0	24,37
4	24,0	24,37
5	24,0	24,31
6	24,0	24,31
Rata-Rata	24	24,35
Standar Deviasi	0	0,031

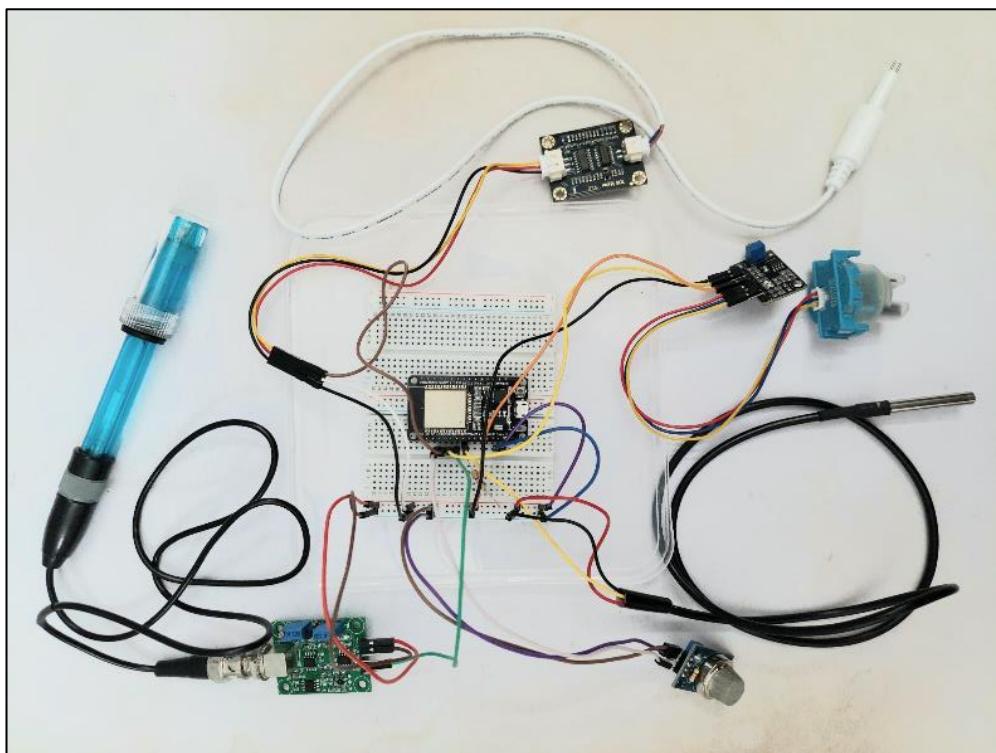
Dapat disimpulkan bahwa, sensor suhu dapat melakukan pengukuran cairan sudah sesuai dengan fungsinya dan dapat menampilkan hasil ukur nilainya, dan perbandingan menggunakan alat ukur suhu dari laboratorium juga menunjukkan hasil yang sesuai dengan sensor suhu.

4.3 Perangkaian Alat

Sensor-sensor yang sudah dikalibrasi kemudian dihubungkan dengan mikrontroler dan dikemas dalam suatu wadah apung yang berfungsi sebagai kemasan untuk alat dapat mengapung di permukaan air.

4.3.1 Rangkaian Sensor

Kumpulan sensor dihubungkan menggunakan kabel *jumper* sesuai dengan pin atau kaki-kaki dari masing-masing sensor. Media penghubung sensor dan mikrontroler menggunakan *breadboard* karena hanya menghubungkan kabel *jumper* dengan sensor dan mikrokontroler.



Gambar 4. 23 Rangkaian Prototipe Alat.

Berikut adalah koneksi pada sensor yang disesuaikan dengan port pada mikrontroler:

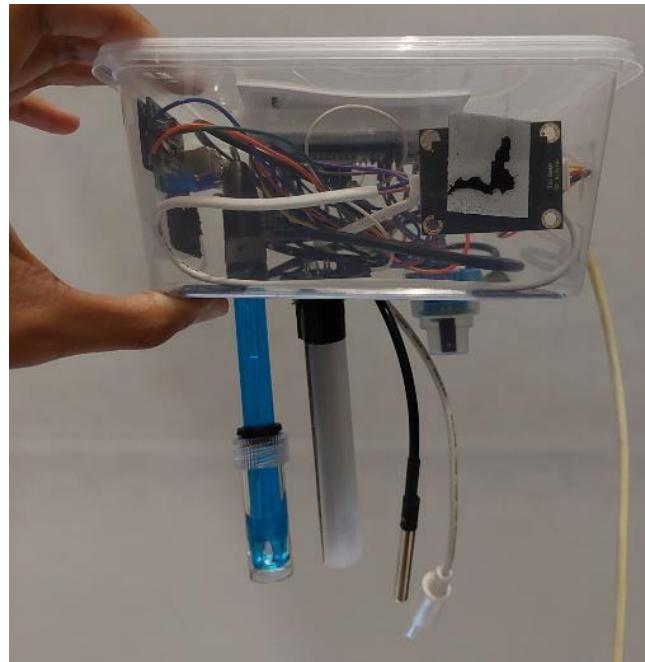
1. Sensor Suhu DS18B20
 - VDD → VIN
 - GND → GND
 - DQ → D25

2. Sensor PH 4502C
 - V+ → 3v3
 - G → GND
 - DO → D33
3. Sensor Kekeruhan SEN-0189
 - V → VIN
 - G → GND
 - A → D32
4. Sensor Gas MQ-135
 - VCC → VIN
 - GND → GND
 - AO → D34
5. Sensor TDS KS0429
 - + → VCC
 - - → GND
 - A → D35

Untuk mengalirkan daya dari mikrokontroler ke kumpulan sensor, kabel *jumper* dari port VIN dan GND pada mikrokontroler dihubungkan pada *power strips* yang tersedia pada *breadboard*. Kumpulan sensor kemudian dihubungkan pada mikrokontroler menggunakan kabel *jumper* dan disesuaikan port untuk menerima daya dan membaca masukkan data pada sensor.

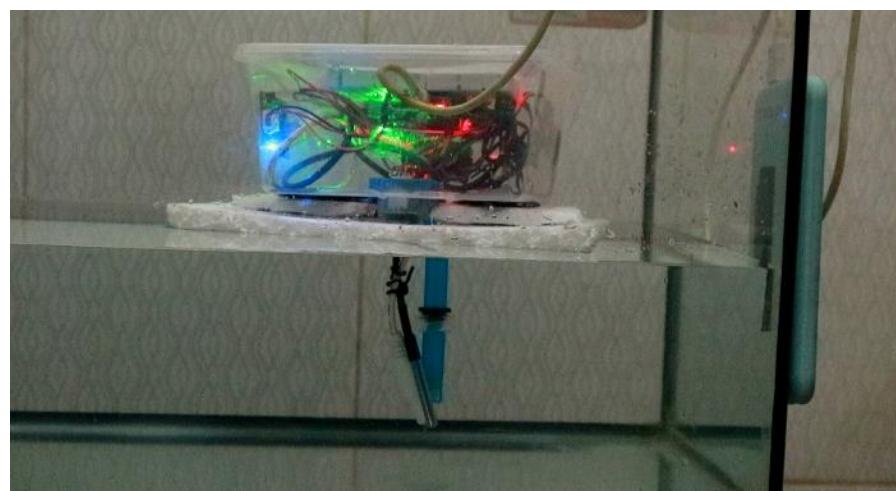
4.3.2 Kemasan Alat

Sensor yang sudah dilakukan kalibrasi kemudian digabungkan menjadi kesatuan perangkat yang ditampung menggunakan sebuah wadah plastik dengan tujuan untuk membuat rangkaian alat menjadi lebih ringkas dan dikemas dengan rapih. Wadah plastik diberi lubang di bagian bawah untuk memberikan ruang untuk *probe* dari sensor-sensor untuk menghadap ke bagian bawah wadah.



Gambar 4. 24 Sensor Dikemas Dalam Wadah Apung.

Bagian bawah wadah kemudian di pasang gabus atau *styrofoam* yang cukup tebal untuk memberi keamanan komponen di dalam wadah terlindungi dari air yang masuk dan memberi wadah kemampuan untuk mengapung di permukaan air.



Gambar 4. 25 Kondisi Wadah Mengapung di Air.

4.4 Pembuatan Aplikasi *Web*

Hasil dari pembacaan kumpulan sensor yang sudah di kalibrasi memerlukan sebuah *platform* untuk menampilkan data dari hasil pengukuran yang sudah ada.

Pembuatan aplikasi *web* menjadi pilihan dikarenakan pembuatan perangkat membutuhkan suatu wadah yang menggunakan *IoT* untuk menyajikan data dan dapat diakses dimana saja dan kapan saja menggunakan *web* dan sambungan internet. Pembuatan aplikasi *web* memerlukan beberapa program tambahan untuk membantu memudahkan dalam proses pengembangan. Berikut adalah *software* dan *hardware* yang digunakan untuk proses pengembangan *web*:

1. *Laptop* dengan spesifikasi sebagai berikut:

- *Processor Ryzen 7 5800U*
- *16GB RAM*
- *512GB NVMe*
- *1 Buah Port USB-A*

2. *Framework Laravel.*

3. *Postman.*

4. *XAMPP Control Panel.*

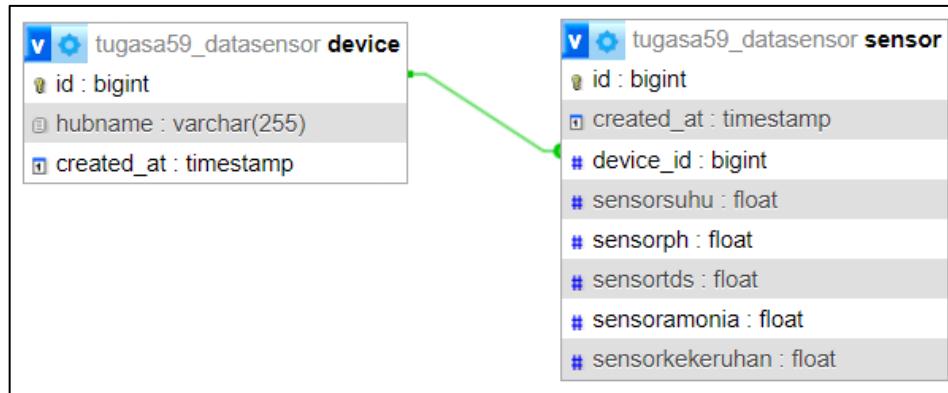
5. *Dbeaver.*

6. *Arc Browser.*

4.4.1 Pembuatan *Database*

Data-data dari sensor membutuhkan sebuah wadah untuk ditampung bila ingin ditampilkan dalam sebuah aplikasi *web*. Penyimpanan *database* dalam pengembangan penelitian ini bersifat lokal, yang artinya nilai data yang tertampung di dalam *database* disimpan di dalam penyimpanan perangkat komputer dan belum diunggah melalui jaringan internet untuk diimplementasikan ke aplikasi *web*.

Database yang diperlukan dalam penelitian ini mempunyai dua tabel yang terdiri dari tabel bernama tabel *device* yang mempunyai fungsi untuk menyimpan data kumpulan perangkat mikrokontroler dengan jumlah lebih dari satu, dan tabel bernama *sensor* yang mempunyai fungsi untuk menyimpan nilai data dari tiap sensor yang dihubungkan oleh mikrokontroler.



Gambar 4. 26 Struktur Database Untuk Perangkat Sensor.

Pada Gambar 4.22, *database* yang mempunyai tabel *device* mempunyai relasi *one to many* yang mengindikasikan data dari sensor-sensor tersimpan dalam tabel *sensor* yang terdiri dari banyaknya kolom yang akan merepresentasikan nama-nama sensor yang akan tersimpan di dalam baris dari tiap kolom.

Pada dasarnya, tabel *device* digunakan untuk menampung kumpulan data perangkat *ESP32* yang terdaftar ke dalam sistem, dan setiap perangkat *ESP32* yang terdaftar di dalam tabel *device* mempunyai tabelnya masing-masing yang bernama tabel *sensor* untuk menampung data dari sensor-sensor yang sudah terpasang di *ESP32*.

Tabel 4. 4 Contoh Isi Tabel *device*.

id	devicename	created_at
1	Mikrokontroler_A	2024-3-24 12:31:05
2	Mikrokontroler_B	2024-4-21 12:31:05
3	Mikrokontroler_C	2024-4-24 12:31:05
4	Mikrokontroler_D	2024-5-13 13:01:02

Tabel *device* mempunyai kumpulan kolom seperti ID dengan tipe data *big integer* untuk mengetahui ID dari tiap mikrokontroler yang sudah terdaftar di dalam *database* dan juga catatan waktu dengan tipe data *timestamps* untuk mengetahui kapan mikrokontroler tercatat aktif di dalam *database*.

Tabel 4. 5 Contoh Isi Tabel sensor.

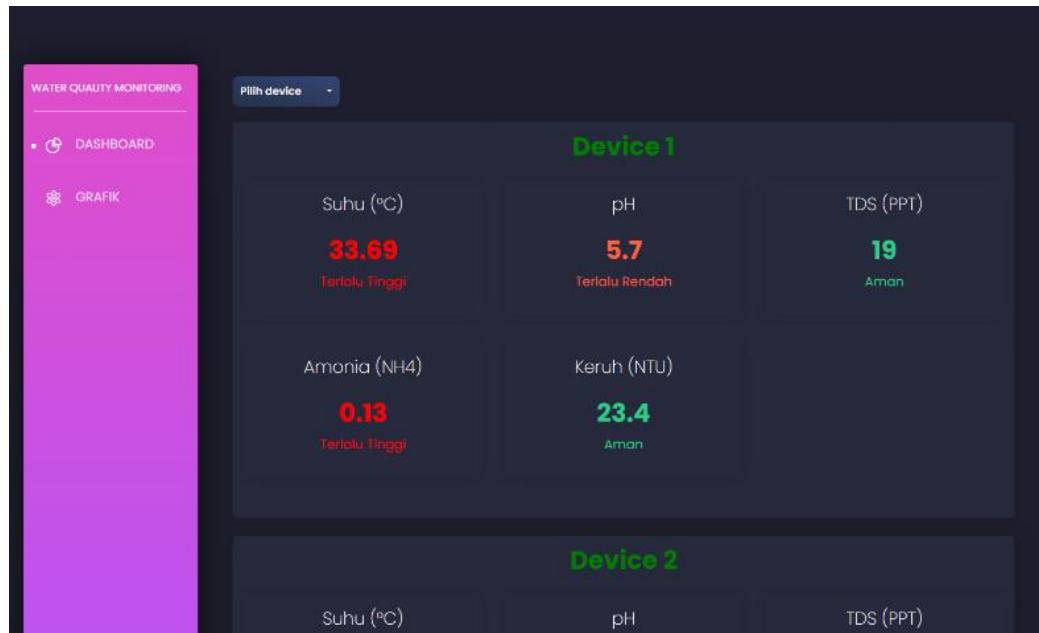
id	created_at	device_id	suhu	ph	tds	amonia	kekeruhan
1	2024-3-24 12:31:05	1	24,8	7,1	15	0,091	15
2	2024-3-24 12:31:05	1	25,8	7,2	15	0,06	15
3	2024-3-24 12:31:05	2	24,6	5,6	23	0,07	23
4	2024-3-24 12:31:05	3	29,7	8,1	21	0,08	21
5	2024-3-24 12:31:05	3	29,6	8,1	21	0,05	21
6	2024-3-24 12:31:05	1	27,8	7,2	16	0,1	16
7	2024-3-24 12:31:05	2	27,8	5,6	22	0,07	22
8	2024-3-24 12:31:05	2	27,1	5,8	22	0,092	22
9	2024-3-24 12:31:05	3	28,1	8	23	0,01	23

Kemudian untuk tabel *device* mempunyai kumpulan kolom untuk menampung nilai dari sensor yang sedang aktif melakukan pengambilan data di lapangan. Kolom dengan nama *device_id* dengan tipe data *big integer* mempunyai fungsi untuk mengetahui bahwa nilai sensor yang masuk ke dalam *database* termasuk dari mikrontroler dengan ID sekian. Catatan waktu juga diperlukan untuk mengetahui kapan data dari sensor masuk ke dalam *database*.

4.4.2 Pembuatan Aplikasi *Web*

Pengembangan aplikasi *web* menggunakan sebuah *framework* bernama *laravel* yang berbasis bahasa pemrograman PHP. *Laravel* bekerja dengan

memberikan berbagai macam *tools* yang dapat diakses dengan membuat lingkup kerja pembuatan *web* dengan bantuan dari *server laravel*.



Gambar 4. 27 Tampilan Awal Halaman Web.

Penggunaan *framework laravel* adalah memudahkan untuk pengembangan aplikasi *web* dalam penggunaan kasus *IoT* dalam penelitian yang sedang dikerjakan, seperti menyambungkan *database* dengan perangkat mikrokontroler dan pembuatan lingkup kerja *web* yang lebih ringkas.

Tampilan halaman awal aplikasi *web* pada gambar 4. 30 mempunyai komponen seperti deret kartu yang mempunyai tujuan sebagai indikator untuk menampilkan data sensor yang sudah tertampung di dalam *database* secara *realtime* saat perangkat sensor sudah terhubung dengan *database* dan sudah diaktifkan. Nilai data pada tampilan halaman *web* diberikan indikator warna untuk memberikan informasi mengenai sensor yang mempunyai nilai di ambang batas bawah dan atas pada kualitas air yang terdapat pada tabel.

```

function fetchAndDisplaydeviceData() {
  $.ajax({
    url: 'API yang ingin digunakan',
    method: 'GET',
    dataType: 'json',
    success: function(response) {
      // Memproses respons untuk mengelompokkan data berdasarkan device_id dan sensor suhu
      response.forEach(function(sensorData) {
        const deviceId = sensorData.device_id;

        // Hanya ambil data sensor terbaru untuk setiap device
        deviceData[deviceId] = sensorData;
      });

      // Menampilkan data terbaru dari setiap device di console (opsional)
      Object.keys(deviceData).forEach(function(deviceId) {
        console.log(`Data terbaru dari device ${deviceId}:`, deviceData[deviceId]);
      });

      // Memanggil fungsi untuk menampilkan data di HTML dengan device_id yang dipilih
      renderdeviceData(deviceData, selecteddeviceId);
    },
    error: function(error) {
      console.error('Error fetching data:', error);
    }
  });
}

```

Gambar 4. 28 Fungsi Menampilkan Data Sensor Bentuk Numerik.

Fungsi untuk menampilkan data sensor dalam halaman *web* menggunakan fungsi *javascript* yang terlampir pada gambar 4.26. Fungsi yang sudah dibuat digunakan untuk mengambil data dari dalam *database* dan ditampilkan ke halaman *web* secara otomatis bahkan ketika isi data di dalam *database* mengalami perubahan secara *realtime*. Data yang diambil kemudian ditampilkan ke dalam *web* merupakan data terakhir yang terekam dan *interval* yang digunakan untuk menampilkan data adalah 2 detik atau 2000 *millisecond*.

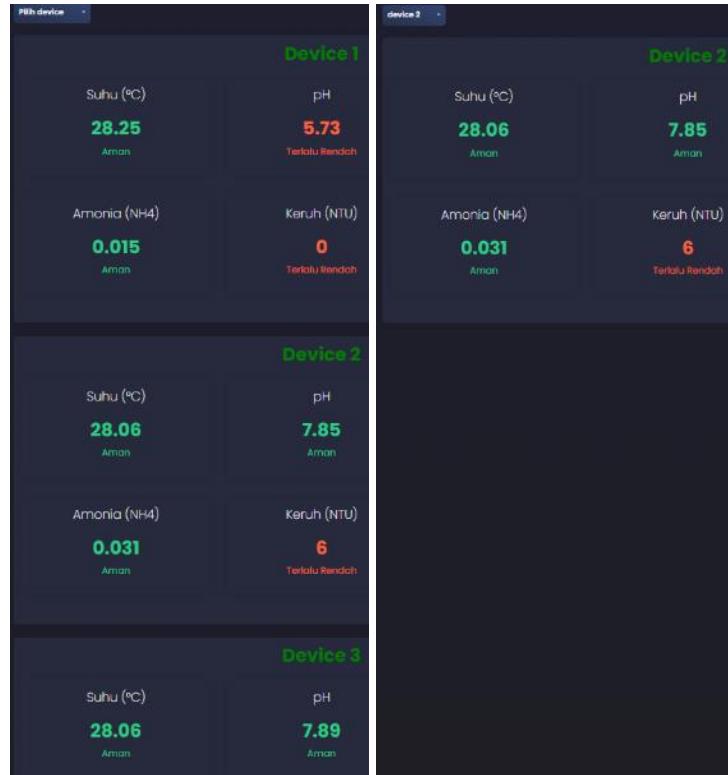
```

$(document).ready(function() {
  // loadInitialData(); // Ambil dan tampilkan data pertama kali
  setInterval(fetchAndDisplaydeviceData, 2000); // Perbarui data setiap 2 detik
});

```

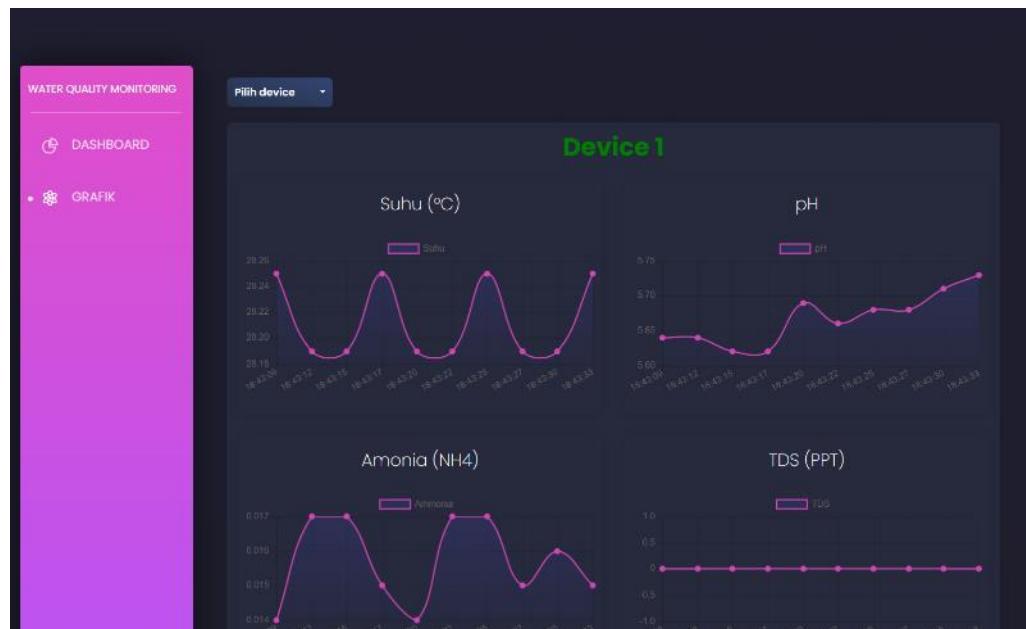
Gambar 4. 29 Fungsi Untuk Menampilkan Data Dalam Interval 2 Detik.

Menu *dropdown* pada halaman utama aplikasi *web* yang terletak di sebelah kiri atas tampilan kartu yang berisikan data-data sensor, yang berfungsi untuk memilih dari *device* atau perangkat mikrokontroler yang ingin tampil sesuai dengan yang terdata di dalam tabel *device*.



Gambar 4. 30 Halaman Web Tampil Semua Device (Kiri) dan Halaman Web Setelah Memilih Device Yang Ingin Ditampilkan (Kanan).

Aplikasi *web* juga mempunyai *sidebar* yang mempunyai komponen menu untuk melihat grafik.



Gambar 4. 31 Tampilan Awal Halaman Grafik.

Grafik pada *web* seperti gambar 4.35 mempunyai fungsi untuk melihat perkembangan naik atau turunnya data yang sudah terukur oleh sensor. Data yang sudah terekam masuk ke dalam *database* juga akan tampil secara *realtime* pada grafik.

Data yang tampil melalui grafik ditampilkan menggunakan fungsi yang terlampir pada gambar 4. 35. Fungsi yang sudah dibuat membuat kumpulan grafik yang digunakan untuk menyimpan nilai data yang sudah tersimpan di dalam *database* dan mengambil data tersebut menggunakan API. Data yang sudah tampil dalam bentuk grafik kemudian diberikan fungsi untuk memperbarui secara otomatis jika ada data baru di dalam *database* tanpa perlu memuat ulang halaman *web*.

```
function fetchDataAndDisplay1(chartId, chartOption, hubId) {
  $.ajax({
    url: '/api/sensor-get',
    method: 'GET',
    dataType: 'json',
    success: function(data) {
      // Filter data berdasarkan hub_id
      const filteredData = data.filter(item => item.hub_id === hubId);
      const latestData = filteredData.slice(-10);
      const labels = latestData.map(item => extractTimeFromCreatedAt(item.created_at));
      const chartData = chartId === 'suhuChart' ? latestData.map(item => item.sensorsuhu) :
        (chartId === 'phChart' ? latestData.map(item => item.sensorsph) :
        (chartId === 'amoniaChart' ? latestData.map(item => item.sensorsamonia) :
        (chartId === 'saltChart' ? latestData.map(item => item.sensortds) :
        latestData.map(item => item.sensorkekeruhan)));
      // Panggil fungsi createChart tanpa variabel global
      if (!charts[chartId]) {
        createChart(chartId, labels, chartData, chartOption);
      } else {
        // Jika chart sudah ada, perbarui data chart
        updateChart(chartId, labels, chartData);
      }
      console.log('Filtered Data (hub_id ' + hubId + '):', filteredData);
    },
    error: function(error) {
      console.error('Error fetching data:', error);
    }
  });
}
```

Gambar 4. 32 Fungsi Menampilkan Data Bentuk Grafik.

Menu *dropdown* pada halaman grafik aplikasi *web* disediakan sedemikian rupa, yang berfungsi untuk memilih dari *device* atau perangkat mikrokontroler yang ingin tampil sesuai dengan yang terdata di dalam tabel *device*.

Data yang terdapat pada menu *dropdown*, diambil dari *database* dengan tabel bernama *device* yang berisikan perangkat mikrokontroler yang sudah terdata, sehingga ketika ada penambahan perangkat mikrokontroler di dalam *database*,

maka menu *dropdown* dan konten pada halaman *web* akan diperbarui secara otomatis sesuai dengan penambahan mikrokontroler dan sensor-sensor baru.

4.4.3 Integrasi API Untuk *Database*

API (*Application Programming Interface*) adalah mekanisme yang memungkinkan dua komponen *software* dapat berkomunikasi menggunakan kumpulan definisi dan protokol. Arsitektur API dijelaskan dalam kaitan lingkup klien dan server, dimana aplikasi yang mengirimkan permintaan disebut klien dan aplikasi yang mengirimkan respon disebut server.

Penggunaan API dalam penelitian ini sudah didapatkan di dalam *framework laravel*, dimana *framework* tersebut sudah mempunyai *tools* API untuk mengintegrasikan *database* yang sudah dibuat dengan aplikasi *web*. Kegunaan API dalam pengembangan aplikasi *web* adalah untuk memudahkan sensor mengirimkan nilai data ke dalam *database* melalui API yang sudah dimuat dalam bentuk URL (*Uniform Resource Locator*) sehingga mikrokontroler hanya perlu menggunakan API yang sudah dibuat di dalam programnya tanpa harus mengintegrasikan *database* di dalam lingkup perangkat mikrokontroler dan sensor.

```
function storesensor(Request $r) {
    DB::table('sensor')->insert([
        'device_id' => $r->device_id,
        'sensorsuhu' => $r->sensorsuhu,
        'sensorph' => $r->sensorph,
        'sensortds' => $r->sensortds,
        'sensoramonia' => $r->sensoramonia,
        'sensorekeruhan' => $r->sensorekeruhan
    ]);
    return 'Data Terkirim ke Database';
}

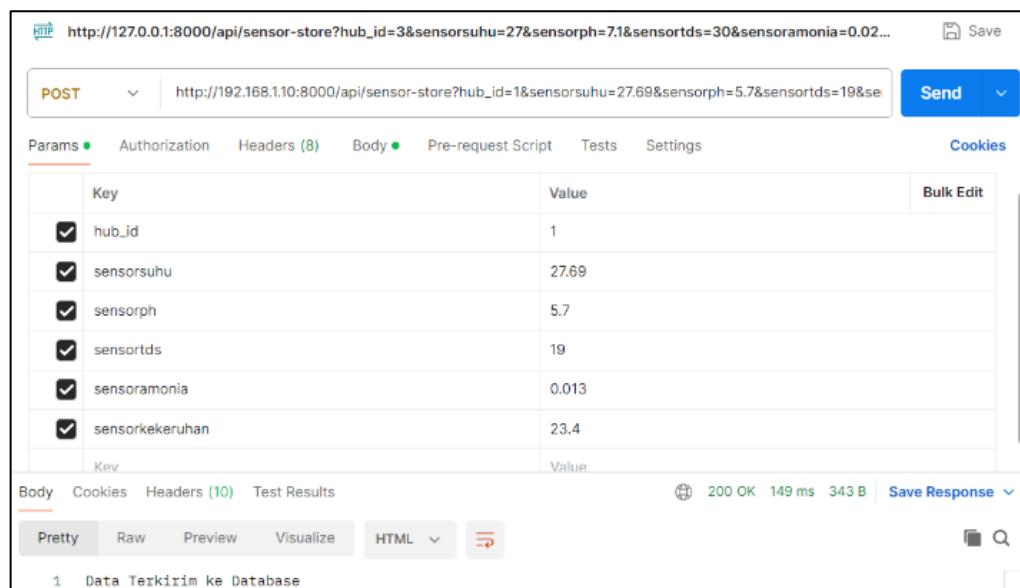
function getsensor() {
    return DB::table('sensor')->get();
}

function getdevice(){
    return DB::table('device')->get();
}
```

Gambar 4. 33 Fungsi Untuk Mengirim dan Menerima Data dari Database.

Untuk mengirimkan data sensor ke dalam *database*, diperlukan fungsi yang terlampir pada gambar 4.27. Fungsi *store* digunakan untuk memasukkan data sensor ke dalam *database* yang sudah dibuat, dan perlu disesuaikan untuk nama variabel dari aplikasi *web* dan nama kolom pada tabel *database*. Fungsi *get* digunakan untuk menerima data dan menampilkannya jika fungsi tersebut dipanggil pada kondisi yang dibutuhkan.

Simulasi pengiriman data sensor untuk dikirimkan ke dalam *database* menggunakan API dilakukan dengan penggunaan aplikasi bernama *postman*. *Postman* memudahkan untuk mensimulasikan ketika sensor sudah membaca data di lapangan dan hendak mengirimkan data tersebut ke dalam *database* melalui API.



Gambar 4. 34 Percobaan API Menggunakan Postman.

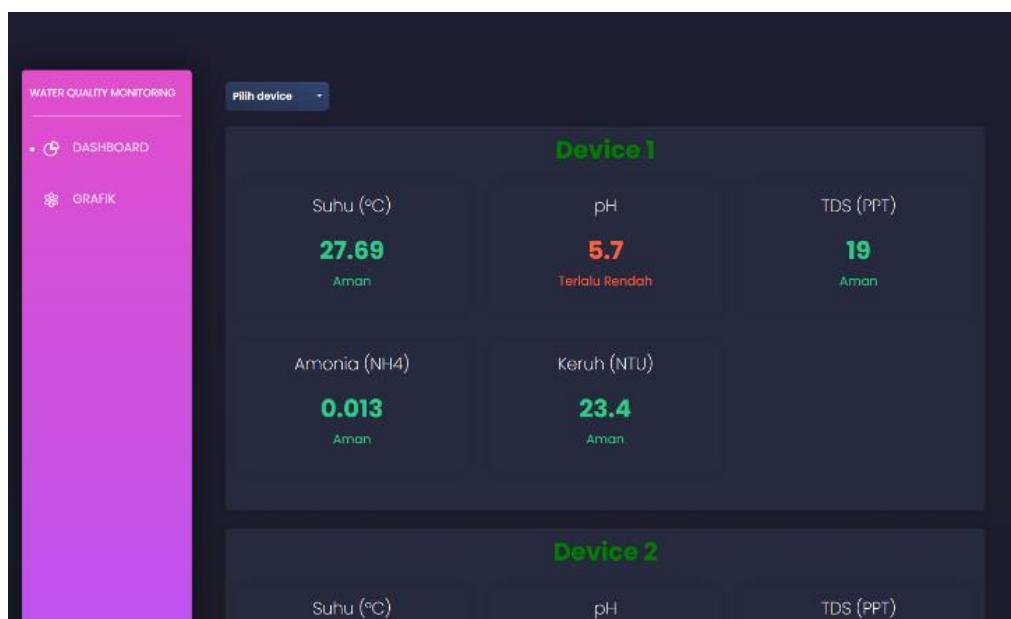
Data yang ingin dikirim maupun ditampilkan melalui *postman* dapat menggunakan fungsi yang terlampir pada gambar 4.38.

```
Route::controller(Csensor::class)->group(function () {
    Route::post('/sensor-store', 'storesensor');
    Route::get('/sensor-get', 'getsensor');
    Route::get('/device-get', 'getdevice');
});
```

Gambar 4. 35 Fungsi Untuk Memanggil Data API.

Dalam lingkup *postman* pada gambar 4.32 , di ilustrasikan data dari sensor ingin dikirmkan ke dalam *database* menggunakan API. *Postman* digunakan untuk mensimulasikan sensor-sensor di lapangan yang sudah menangkap data secara *realtime* menggunakan metode *post* ke URL yang sudah dibuat di dalam *framework laravel*.

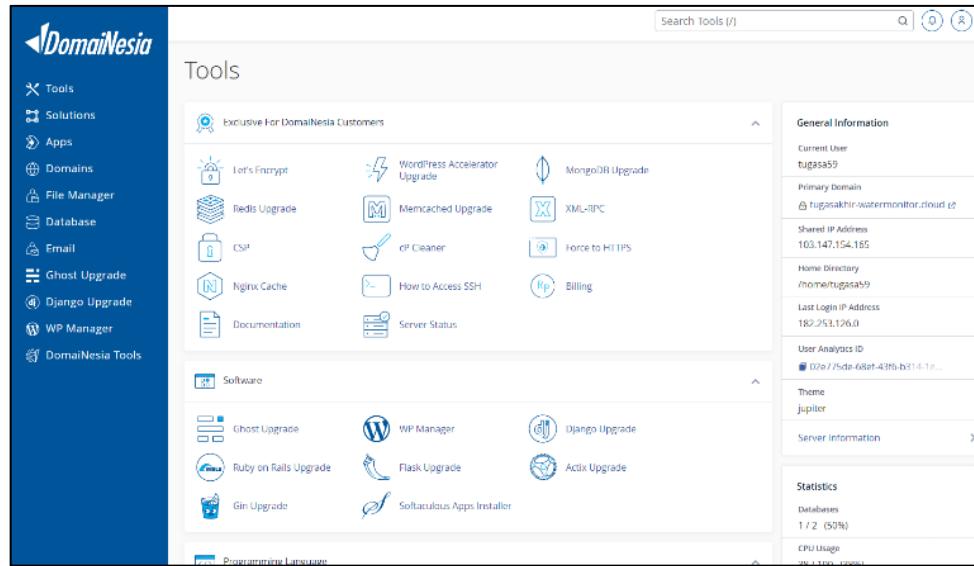
Dikarenakan perangkat mikrontroler dan sensor belum tersambung ke API, maka untuk sementara data di inputkan secara manual sesuai nama kolom yang ada pada tabel di *database* untuk menguji hubungan antara API dengan *database* dan *database* dapat menerima serta menampilkan data tersebut di dalam aplikasi *web* seperti pada gambar 4.33.



Gambar 4. 36 Tampilan Web Menerima Data Realtime Dari Postman.

4.4.4 Akses Daring Apikasi *Web*

Aplikasi *web* yang sudah dibuat masih bersifat lokal dan belum bisa diakses secara daring. Untuk permasalahan tersebut, diperlukannya sebuah *web-hosting* untuk menampung aplikasi *web* ke dalam server yang sudah berbasis daring dan dapat menghasilkan sebuah URL yang dapat diakses menggunakan jaringan internet.



Gambar 4. 37 Tampilan Dashboard DomaiNesia.

Web-hosting menggunakan layanan dari sebuah perusahaan penyedia jasa *web-hosting* yang bernama DomaiNesia. Komponen-komponen aplikasi *web* yang sudah dibuat secara lokal akan ditampung di dalam layanan *server* DomaiNesia. Aplikasi *web* yang sudah dibuat kemudian diterbitkan menggunakan layanan dari DomaiNesia untuk membuat *web* dapat diakses secara daring dan mempunyai nama domain yaitu, <https://tugasakhir-watermonitor.cloud/dashboard>.

4.5 Penggunaan Alat

Penggunaan prototipe alat cukup dihubungkan dengan daya melalui kabel daya yang sudah terpasang ke mikrokontroler dan meletakkan alat di permukaan air. Alat sudah otomatis menyala dan terhubung ke jaringan internet yang terdaftar di dalam perangkat mikrokontroler. Jaringan internet perlu di registrasi secara manual menggunakan aplikasi *arduino IDE* untuk memasukkan nama jaringan dan kata sandi dari jaringan internet baru yang akan dikenali oleh mikrokontroler.

```
13 const char* ssidList[] = {"LAUSEPA FAMILY", "hp adam"};
14 const char* passwordList[] = {"717451F4", "adam0309"};
```

Gambar 4. 38 Fungsi Untuk Menambah Jaringan Internet Baru.

Pendaftaran perangkat atau mikrokontroler baru yang ingin ditampilkan di dalam *web* dapat dilakukan dengan mengubah nilai pada gambar 4.38, dimana nilai *device_id* dirubah dengan merubah nilai yang berbeda dari perangkat lain yang sudah terdaftar di dalam *database*.

```
81 url += "&device_id=1";
```

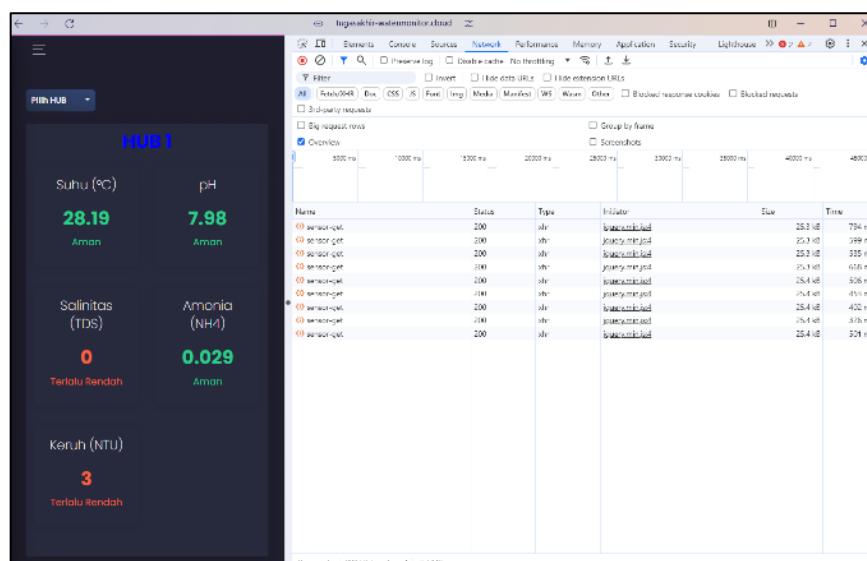
Gambar 4. 39 Fungsi Registrasi Alat Baru.

4.6 Pengujian

Pengujian pada perangkat terbagi menjadi beberapa bagian, diantaranya adalah pengujian untuk mengamati kecepatan waktu untuk *web* dalam menerima data yang diambil dari *database*, pengujian menggunakan air tambak tiruan berupa air bersih yang dicampur air garam dan subtansi serupa, dan penyebaran angket untuk menerima respon kelayakan produk yang dibuat.

4.6.1 Pengujian Transfer Data Sensor

Pengujian dilakukan untuk mengamati kecepatan API dalam menerima data melalui *database* dan ditampilkan ke dalam aplikasi *web*. Pengujian menggunakan aplikasi *web browser* Bernama *Arc Browser* dengan menggunakan *developer tools* dan menggunakan fitur *network* untuk mengamati API dalam menerima data melalui *database*.



Gambar 4. 40 Fungsi Network Pada Developer Tools.

Hasil pengujian kecepatan API dalam mengirimkan data ke dalam aplikasi *web* dilakukan dengan mengamati 10 rekam data dengan *interval* untuk menampilkan data sebesar 2 detik atau 2000 *millisecond* dari setiap komponen sensor dalam mengirimkan data ke dalam *database* dan menampilkannya ke dalam *web* menggunakan API. Pengujian dilakukan dengan menghitung rata-rata dari waktu yang terekam dan standar deviasi untuk membantu memahami variabilitas waktu respons.

Tabel 4. 6 Hasil Pengujian Kecepatan Transfer Data Pada Web Dalam Millisecond.

No.	Data Sensor Dalam Array (Millisecond)
1	795,67
2	600,39
3	536,74
4	669,27
5	507,41
6	453,82
7	406,31
8	326,78
9	501,83
10	422,34
Rata-Rata	522,06
Standar Deviasi	137,29

Pengujian kecepatan API menunjukkan bahwa waktu rata-rata respons API adalah 522,06 ms, dengan standar deviasi 137,29 ms. Berdasarkan standar industri, waktu respons ideal seharusnya kurang dari 100 ms. Oleh karena itu, diperlukan optimasi lebih lanjut pada API untuk mencapai standar performa yang lebih baik.

4.6.2 Pengujian Simulasi Pada Air Tambak Tiruan

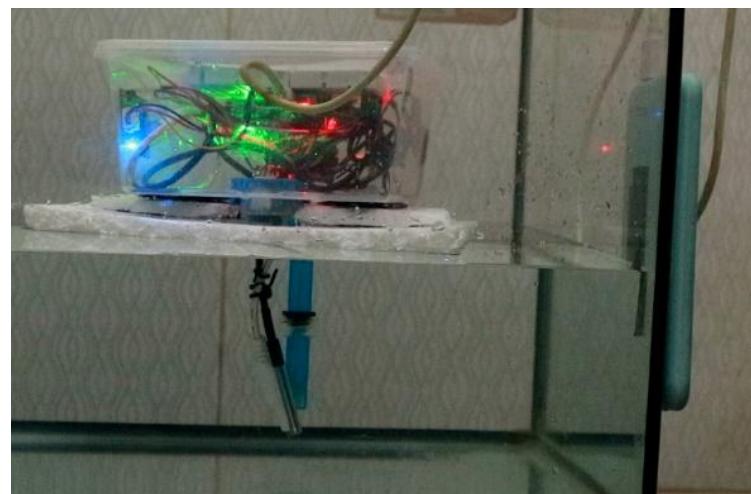
Pengujian prototipe alat dilakukan menggunakan air tambak tiruan dengan menggunakan skenario yang berbeda untuk mengamati kemampuan alat untuk mengukur kualitas air sesuai dengan pengaturan program untuk standar kualitas air tambak. Air tambak menggunakan air laut dikarenakan udang merupakan makhluk

yang hidup di perairan air laut, sehingga tiruan air tambak dibuat dengan menggunakan garam seperti garam dapur yang dilarutkan menggunakan air bersih dan dicampur sehingga tiruan air laut yang digunakan untuk melakukan pengujian pada alat.



Gambar 4. 41 Garam Dicampur Dengan Air Jernih.

Alat kemudian diletakkan di atas permukaan air dengan menggunakan wadah apung yang sudah dibuat sedemikian rupa, sehingga komponen lain tidak rusak dikarenakan tersentuh atau tergenang oleh air.



Gambar 4. 42 Alat Mengapung Di Permukaan Air Tambak Tiruan Kondisi Jernih.

Hasil Pengujian data sensor dilakukan dengan air tambak tiruan dengan kondisi bersih atau bening dan dilakukan pembacaan sebanyak 10 rekam data pengujian kemudian diperiksa rata-ratanya beserta standar deviasinya.

Tabel 4. 7 Hasil Pengujian Air Tambak Tiruan Kondisi Jernih.

No	Suhu (°C)	pH	Amonia (ppm)	TDS (ppt)	Kekeruhan (NTU)
1	25,62	7,79	0,0123	19,2	1
2	25,62	8,26	0,0128	19,4	-4
3	25,62	8,26	0,0138	19,5	-3
4	25,62	7,4	0,0126	19,6	-3
5	25,56	8,45	0,0127	19,3	-3
6	25,62	8,23	0,0128	19,3	-4
7	25,62	8,33	0,0127	19,1	-3
8	25,62	8,37	0,0127	19,2	-3
9	25,62	8,24	0,0137	19,3	-4
10	25,62	8,25	0,0143	19,3	-3
Rata-Rata	25,61	8,15	0,013	19,32	-2,9
Standar Deviasi	0,011	0,31	0,00065	0,14	1,44

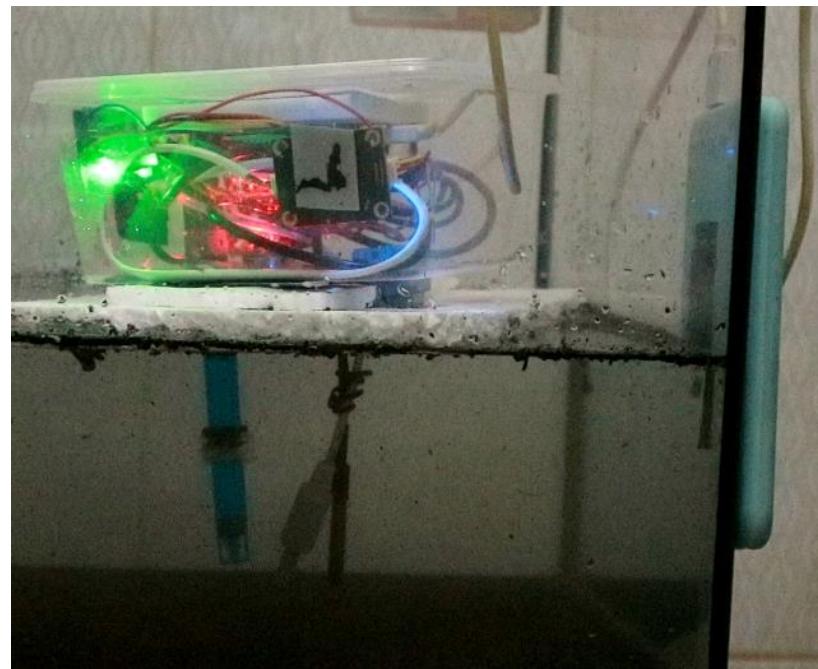
Hasil pengujian sensor untuk mengukur kualitas air tambak tiruan dalam kondisi air jernih menunjukkan akurasi yang memadai terhadap parameter kualitas air. Suhu rata-rata tercatat 25,61°C (standar deviasi 0,019°C), sedikit di bawah batas bawah 26°C. pH rata-rata 8,158 (standar deviasi 0,31), berada dalam rentang standar 7,5-8,8. Konsentrasi amonia rata-rata 0,013 ppm (standar deviasi 0,00065), jauh di bawah batas atas 0,1 ppm. Salinitas rata-rata 19,32 ppt (standar deviasi 0,14), sesuai dengan kisaran standar 10-30 ppt. Namun, pengukuran kekeruhan menunjukkan rata-rata -2,9 NTU (standar deviasi 1,44), jauh di bawah standar 15-40 NTU, yang mungkin memerlukan kalibrasi sensor lebih lanjut. Secara keseluruhan, sensor suhu, pH, amonia, dan salinitas memberikan hasil yang akurat, sedangkan sensor kekeruhan memerlukan evaluasi tambahan untuk memastikan keandalan dalam berbagai kondisi air.

Pengujian berikutnya menggunakan media tanah sebagai pembentuk skenario air tambak yang kotor atau keruh dan mengamati kondisi alat dalam mengukur data kualitas air.



Gambar 4. 43 Tanah Dicampur Dengan Air Garam Bersih.

Alat tetap diletakkan di atas permukaan air untuk mengamati kondisi air garam yang sudah dicampur oleh tanah untuk membuat skenario air tambak yang kotor atau kurang optimal.



Gambar 4. 44 Alat Mengapung Di Permukaan Air Tambak Tiruan Kondisi Kotor.

Hasil pengujian menggunakan air tambak tiruan yang terkontaminasi dengan tanah menunjukkan beberapa perbedaan signifikan dibandingkan dengan kondisi air jernih.

Tabel 4. 8 Hasil Pengujian Air Tambak Tiruan Kondisi Kotor.

No	Suhu (°C)	pH	Amonia (ppm)	TDS (ppt)	Kekeruhan (NTU)
1	25,44	8,08	0,0186	20,1	30
2	25,5	7,14	0,0185	19,7	25
3	25,56	7,21	0,02	19,7	26
4	25,56	7,21	0,0201	19,9	28
5	25,56	7,21	0,0248	19,7	26
6	25,62	7,09	0,0234	19,6	26
7	25,62	8,09	0,0295	19,7	27
8	25,62	8,06	0,0262	19,6	23
9	25,62	7,09	0,0273	19,6	23
10	25,62	7,21	0,0247	19,8	23
Rata-Rata	25,57	7,43	0,0233	19,74	25,7
Standar Deviasi	0,061	0,44	0,00384	0,15	2,31

Suhu rata-rata yang tercatat adalah 25,57°C (standar deviasi 0,062°C), masih berada di bawah batas bawah 26°C. Namun, pH rata-rata 7,43 (standar deviasi 0,44) dan konsentrasi amonia rata-rata 0,0233 ppm (standar deviasi 0,00384) menunjukkan variasi yang signifikan dari kondisi ideal. Salinitas rata-rata mencapai 19,74 ppt (standar deviasi 0,15), sesuai dengan rentang standar 10-30 ppt. Keberuhan rata-rata 25,7 NTU (standar deviasi 2,31) juga melampaui batas atas 15-40 NTU, menunjukkan kemungkinan adanya tantangan dalam mendeteksi kekeruhan pada kondisi air yang kotor. Secara keseluruhan, meskipun sensor suhu dan salinitas memberikan hasil yang sesuai dengan standar, peningkatan pH dan kekeruhan serta variasi amonia menunjukkan perlunya penyesuaian sensor untuk mendapatkan hasil yang lebih konsisten dalam simulasi air tambak yang kotor.

4.6.3 Pengujian Aplikasi Web

Untuk menguji efektivitas prototipe alat pengukur kualitas air berbasis IoT pada tambak udang, untuk pengujian produk dilakukan dengan metode survei. Proses ini diawali dengan pembuatan video demo yang menjelaskan cara kerja dari prototipe alat. Video tersebut kemudian diunggah ke *platform* daring untuk memudahkan akses oleh calon responden. Survei dilakukan dengan menyebarluaskan angket secara daring yang dilengkapi dengan tautan video demo, kepada individu-individu yang berprofesi di bidang informatika serta mereka yang terlibat dalam kegiatan budidaya tambak. Tujuannya adalah untuk mendapatkan umpan balik yang relevan dan mendalam dari para ahli dan praktisi terkait, guna mengevaluasi akurasi data, kemudahan penggunaan, dan potensi pengembangan lebih lanjut dari alat yang saya kembangkan.

Berdasarkan hasil survei yang melibatkan 7 responden dari berbagai latar belakang profesional, prototipe alat pengukur kualitas air berbasis IoT yang dikembangkan menunjukkan penerimaan yang positif.

Tabel 4. 9 Profil Responden.

Profesi Responden	Jumlah Responden
Manager Produksi	1 (14,3%)
Budidaya Udang	
Frontend Developer	1 (14,3%)
Karyawan Swasta	1 (14,3%)
Lead Backend Developer	1 (14,3%)
Mahasiswa	2 (28,6%)
Staf	1 (14,3%)

Sebagian besar responden menilai bahwa alat ini mudah digunakan, dengan antarmuka web yang intuitif dan data yang cukup akurat. Sebanyak 85,7% responden menemukan data yang disajikan alat ini akurat, sementara 57,1% merasa bahwa data tersebut sangat mudah ditemukan. Selain itu, 85,7% responden menilai antarmuka web aplikasi sangat mudah digunakan, dan 71,4% merasa bisa mulai dan

mengoperasikan alat ini dengan cepat setelah menonton demo. Responden juga menekankan pentingnya akses data kualitas air secara real-time dengan 85,8% menganggapnya sangat penting atau penting. Alat ini dianggap cukup efektif dalam membantu memahami kualitas air dibandingkan metode sebelumnya, dengan 85,7% responden menyatakan bahwa prototipe alat cukup efektif. Meskipun data disajikan dalam bentuk angka dan grafik dianggap cukup jelas dan informatif oleh sebagian besar responden (57,1%), ada ruang untuk perbaikan lebih lanjut terutama dalam meningkatkan kejelasan dan akurasi data. Temuan ini menunjukkan bahwa alat ini memiliki potensi yang baik untuk membantu pengelolaan kualitas air dan memberikan panduan untuk pengembangan lebih lanjut.

Tabel 4. 10 Hasil Survei Kelayakan Prototipe.

Aspek Penilaian	Kategori	Responden
Pengalaman	Tidak Ada	0 (0%)
Menggunakan Perangkat IoT	Sedikit Pengalaman	3 (42,9%)
	Cukup Berpengalaman	3 (42,9%)
	Sangat Berpengalaman	1 (14,3%)
Kemudahan Membaca Data	Sangat Mudah	4 (57,1%)
	Cukup Mudah	3 (42,9%)
	Cukup Sulit	0 (0%)
	Sangat Sulit	0 (0%)
Keakuratan Data	Sangat Akurat	0 (0%)
	Akurat	6 (85,7%)
	Kurang Akurat	1 (14,3%)
	Tidak Akurat	0 (0%)

Tabel 4. 10 Hasil Survei Kelayakan Prototipe (Lanjutan).

Aspek Penilaian	Kategori	Responden
Kemudahan Penggunaan	Sangat Mudah	6 (85,7%)
Antarmuka Web	Cukup Mudah	1 (14,3%)
	Cukup Sulit	0 (0%)
	Sangat Sulit	0 (0%)
Kecepatan	Sangat Cepat	2 (28,6%)
Pengoperasian Setelah	Cukup Cepat	5 (71,4%)
Menonton Demo	Lambat	0 (0%)
	Sangat Lambat	0 (0%)
Pentingnya Akses Data	Sangat Penting	3 (42,9%)
Real-time	Penting	3 (42,9%)
	Tidak Begitu Penting	1 (14,3%)
	Tidak Penting	0 (0%)
Efektivitas Alat	Sangat Efektif	1 (14,3%)
Dibandingkan Metode	Cukup Efektif	6 (85,7%)
Konvensional	Kurang Efektif	0 (0%)
	Tidak Efektif	0 (0%)

BAB V

PENUTUP

5.1 Simpulan

Prototipe alat ukur kualitas air yang dirancang dan diuji dalam tugas akhir ini telah menunjukkan kemajuan signifikan dalam kemampuan pengukuran parameter kualitas air. Hasil pengujian pada kondisi air jernih menunjukkan bahwa sensor suhu, pH, amonia, dan padatan terlarut berupa salinitas memberikan hasil yang akurat sesuai dengan standar yang ditetapkan. Meskipun demikian, sensor kekeruhan memerlukan penyesuaian lebih lanjut untuk meningkatkan konsistensi dalam mendeteksi kekeruhan pada kondisi air yang kotor. Pengujian menggunakan sampel air tambak yang terkontaminasi dengan tanah menunjukkan variasi yang signifikan dalam hasil pengukuran, menyoroti tantangan dalam adaptasi prototipe alat untuk lingkungan air tambak yang lebih kompleks. Selain itu, aplikasi *web* yang terintegrasi dengan prototipe alat telah berhasil menampilkan data pengukuran secara langsung, namun masih memerlukan optimalisasi untuk meningkatkan kecepatan penerimaan data melalui API. Dengan demikian, meskipun prototipe ini masih dalam tahap awal pengembangan dan lebih berfokus pada studi kasus serta penelitian daripada aplikasi industri, hasil keseluruhan menunjukkan kemungkinan untuk pengembangan lebih lanjut guna meningkatkan performa dan aplikabilitas alat dalam kondisi lingkungan yang lebih kompleks di masa mendatang.

5.2 Saran

Saran untuk pengembangan alat pendeksi kualitas air berbasis IoT selanjutnya adalah sebagai berikut:

1. Melakukan optimasi terhadap API sehingga kecepatan API dalam menerima data untuk dikirimkan ke antarmuka *web* dapat di bawah 100 ms (*Millisecond*).
2. Melakukan pengukuran menggunakan alat-alat laboratorium sesuai dengan sensor-sensor yang digunakan untuk keakuratan hasil.

DAFTAR PUSTAKA

- Adani, Farhan, & S. S. (2019). Internet of Things: Sejarah teknologi dan penerapannya. *Jurnal Online Sekolah Tinggi Teknologi Mandala 14*, no. 2 , 92-99.
- Anggraeni, Sri, S. W., E. W., & K. K. (2022). Pendampingan Monitoring Kondisi Air Tambak Udang Varane Berbasis WEB. *Jurnal DIANMAS 11*, no. 1.
- Ansarullah, D., & Nurwasito, H. (2022). Monitoring Kualitas Air pada Tambak Udang berbasis Internet of Things dengan Protokol Komunikasi ZigBee. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer 6*(2), 615-624.
- Budiman, F., Rivai, M., & Nugroho, M. A. (2019). International Seminar on Intelligent Technology and Its Applications (ISITIA). *Monitoring and Control System for Ammonia and pH Levels for Fish Cultivation Implemented on Raspberry Pi 3B*.
- DFRobot. (2017, January 31). *DigiKey*. Diambil kembali dari SEN0189 by DFRobot Datasheet | DigiKey: <https://www.digikey.com/en/htmldatasheets/production/2071175/0/0/1/sen0189>
- Faruq, M. M., & Hirawan, D. (2018). WATER QUALITY MONITORING SYSTEM. *1st Int. Conf. Mater. Eng. Manag.-Eng. Sect (Vol. 165)*.
- Gregersen, C. R. (2023, October 2023). *3 IoT Latency Issues and How to Fix Them*. Diambil kembali dari Built In: <https://builtin.com/articles/how-to-fix-iot-latency>
- Hooda, P. (2024, April 17). *Introduction to Postman for API Development - GeeksforGeeks*. Diambil kembali dari GeeksforGeeks: <https://www.geeksforgeeks.org/introduction-postman-api-development/>
- Imrana, I., Nursati, E., Ramadhani, F., Angriawan, F., Illijas, M. I., & Saleh, L. (2023). Aplikasi Bakteri Heterotrof untuk Mengatasi Fluktuasi pH Air pada Pembesaran Udang Vaname Secara Intensif. *Lutjanus 28*, no. 1, 74-79.
- Inayah, Z. N., Musa, M., & Arfiati, D. (2023). Growth of Vannamei Shrimp (*Litopenaeus vannamei*) in Intensive Cultivation Systems. *Jurnal Penelitian Pendidikan IPA*, 8821–8829.
- Intan, I., Nurlina, Pangerang, F., & Mulyawan, A. (2020). Sistem Monitoring Sirkulasi Air pada Budidaya Udang Vaname berbasis. *Pekommas*, 203-214.

- Lawi, A., Panggabean, B. L., & Yoshida, T. (2021). Evaluating graphql and rest api services performance in a massive and intensive accessible information system. *Computers 10(11)*, 128.
- Lusiana, Ritonga, Sudrajat, M. A., & Arifin, M. Z. (2021). Manajemen pakan pada pembesaran udang vannamei (*Litopenaeus vannamei*) di tambak intensif CV. Bilangan Sejahtera Bersama. *Jurnal Penelitian Chanos Chanos 19*, no. 2, 187-189.
- Nanda Rizky, F. J., Hari Saputra, R. S., Supendar, H., & Budiawan, I. (2020). Tambak-Ku: Sarana Penunjang Dalam Industri Udang Untuk Mengkuti Pekembangan Era Industri 4.0. *Jurnal Infortech 2(2)*, 145-152.
- Noviana, & Rina. (2022). Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan MYSQL. *Jurnal Teknik dan Science 1*, no. 2, 112-124.
- Pauzi, G. A., Syafira, M. A., Surtono, A., & Supriyanto, A. (2017). Aplikasi IoT Sistem Monitoring Kualitas Air Tambak Udang . *Jurusan Fisika, Universitas Lampung, Bandar Lampung, Indonesia, 35141*.
- Psathwara, J. (2024, March 20). *Introduction to Laravel and MVC Framework - GeeksforGeeks*. Diambil kembali dari GeeksforGeeks: <https://www.geeksforgeeks.org/introduction-to-laravel-and-mvc-framework/>
- Purnamasari, I., Purnama, D., & Utami, M. (2017). Pertumbuhan udang vaname (*Litopenaeus vannamei*) di tambak intensif. *Jurnal enggano, 2(1)*, 58-67.
- Rafiquzzaman, M. (2018). *Microcontroller Theory and Applications with the PIC18F Second Edition*. John Wiley & Sons.
- Samura, Ayu, W. K., & Setyawan, G. E. (2018). Sistem Kontrol dan Monitoring Kualitas Air Tambak Udang Windu Dengan Metode Fuzzy Logic Control Menggunakan Mikrokontroler NI myRIO. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(9)*.
- Sanjaya, A. (2020, Februari 4). *NEVERMIND TECH* . Diambil kembali dari Kalibrasi Sensor MQ135 Untuk Deteksi NH3(Amonia) Dengan Arduino: <https://www.youtube.com/watch?v=x1rUew7S5LA>
- Suwarsih, Marsoedi, N. H., & M. M. (2016). Kondisi kualitas air pada budidaya udang di tambak wilayah pesisir kecamatan Palang kabupaten Tuban. *Prosiding Seminar Nasional Kelautan, vol. 1, no. 1*, 138-143.
- Usina. (2018, June 26). *How to use a PH probe and sensor*. Diambil kembali dari How to use a PH probe and sensor: <https://cdn.awsli.com.br/969/969921/arquivos/ph-sensor-ph-4502c.pdf>

Wagyana, A. (2019). Prototipe Modul Praktik untuk Pengembangan Aplikasi Internet of Things (IoT). *Setrum: Sistem Kendali-Tenaga-elektronika-telekomunikasi-komputer* 8, no. 2, 238-247.

Wardhana, E. K. (2022). *Rancang Bangun Prototipe Sistem Rekomendasi Kualitas Air Untuk Budidaya Ikan Nila*. Malang: Ernanda Kusuma Wardhana.