

**Perancangan Sistem Otomatisasi Pada Peternakan Ikan  
Berbasis *Microcontroller* Arduino IDE Untuk Meningkatkan Hasil Panen Ikan  
(Studi Kasus: Peternakan Ikan Minaverse Company)**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1  
Pada Jurusan Teknik Industri Fakultas Teknologi Industri**



Disusun Oleh:

Nama : Muhammad Abdullah Salam  
No. Mahasiswa : 19522254

**PROGRAM STUDI TEKNIK INDUSTRI  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2023**

## SURAT PERNYATAAN KEASLIAN

Saya mengakui bahwa pelaksanaan dan hasil dari pelaksanaan tugas akhir ini adalah hasil karya sendiri kecuali kutipan dan ringkasan yang seluruhnya sudah saya jelaskan sumber-sumbernya. Jika kemudian hari ternyata pengakuan saya terbukti tidak benar dan melanggar peraturan yang sah, maka saya sangat bersedia menerima konsekuensi apabila ijazah yang sama terima ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 01 Juli 2023



(Muhammad Abdullah Salam)

19522254

## SURAT KETERANGAN PELAKSANAAN TA



FAKULTAS  
TEKNOLOGI INDUSTRI

Gedung KH. Mas Mansur  
Kampus Terpadu Universitas Islam Indonesia  
Jl. Kaliurang km 14,5 Yogyakarta 55584  
T. (0274) 898444 ext. 4110, 4100  
F. (0274) 895007  
E. [fti@uii.ac.id](mailto:fti@uii.ac.id)  
W. [fti.uui.ac.id](http://fti.uui.ac.id)

### SURAT KETERANGAN PENELITIAN

Nomor : 143/Ka.lab SIMANTI/20/Lab.SIMANTI/VII/2023

*Assalamu 'alaikum Warohmatullahi Wabarokaatuh*

Dengan hormat,

Yang bertanda tangan dibawah ini, menerangkan bahwa:

Nama : Muhammad Abdullah Salam

Nim : 19522254

Jurusan : Teknik Industri

Dosen Pembimbing : Ir. Muchamad Sugarindra, S.T., M.T., IPM

Menyatakan bahwa mahasiswa tersebut diatas telah melaksanakan penelitian tugas akhir dengan judul " **Perancangan Sistem Otomatisasi Pada Peternakan Ikan Berbasis *Microcontroller* Arduino IDE Untuk Meningkatkan Hasil Panen Ikan**" mulai pelaksanaan penelitian tanggal 1 Mei 2023 sampai tanggal 3 Juli 2023.

Demikian surat keterangan penelitian ini kami buat. Atas perhatiannya dan kerja samanya kami mengucapkan terima kasih.

*Wassalamu 'alaikum Warohmatullahi Wabarokaatuh*

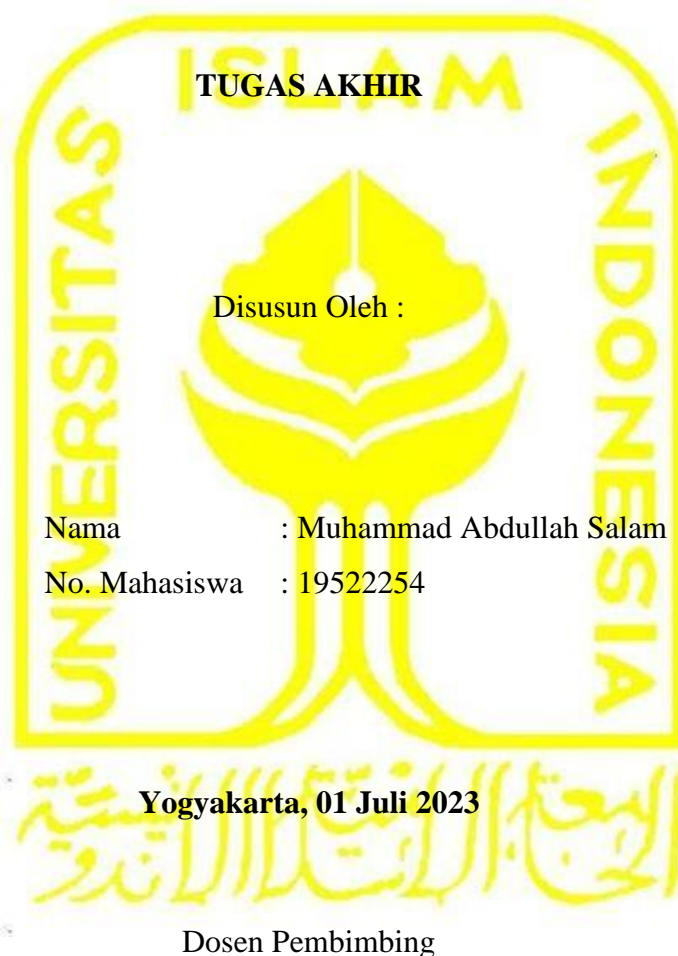
Yogyakarta, 10 Juli 2023

Kepala Laboratorium  
Sistem Manufaktur Terintegrasi

Abdullah 'Azzam, S.T., M.T.

**LEMBAR PENGESAHAN PEMBIMBING**

**Perancangan Sistem Otomatisasi Pada Peternakan Ikan  
Berbasis *Microcontroller* Arduino IDE Untuk Meningkatkan Hasil Panen  
Ikan**



**(Ir. Muchamad Sugarindra, S.T., M.T., IPM)**

**LEMBAR PENGESAHAN DOSEN PENGUJI**

**Perancangan Sistem Otomatisasi Pada Peternakan Ikan  
Berbasis *Microcontroller* Arduino IDE Untuk Meningkatkan Hasil Panen Ikan**

**TUGAS AKHIR**

Disusun Oleh :

Nama : Muhammad Abdullah Salam

No. Mahasiswa : 19522254

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata-1 Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 04 Agustus 2023

**Tim Penguji**Ir. Muchamad Sugarindra, S.T., M.T., IPM

Ketua

Amarria Dila Sari, S.T., M.Sc.

Anggota I

Bambang Suratno, S.T., M.T., Ph.D.

Anggota II

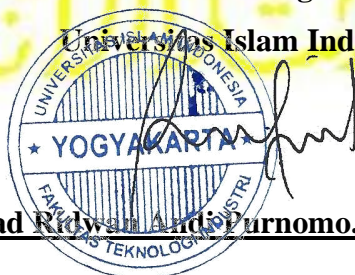


Mengetahui,

Ketua Program Studi Teknik Industri

Fakultas Teknologi Industri

Universitas Islam Indonesia

Ir. Muhammad Ridwan Andio Purnomo, S.T., M.Sc., Ph.D., IPM

015220101

## **HALAMAN PERSEMBAHAN**

Puji dan syukur kepada Zat yang Maha Agung,

Allah SWT atas segala berkat, rahmat, nikmat, hikmah, dan perlindunganNya penulis dapat menyelesaikan penulisan skripsi ini.

Sholawat dan salam selalu tercurahkan kepada Nabi besar Muhammad SAW, keluarga, sahabat, dan para pengikutnya.

Skripsi penulis persembahkan kepada:

**Abah dan Mama**

Terima kasih atas semua kasih sayang yang tulus, perjuangan, pengorbanan, dan untaian setiap doa dan sujud yang tersampaikan kepada Penulis

## MOTTO

“Dan carilah pada apa yang telah dianugerahkan Allah kepadamu (kebahagiaan) negeri akhirat, dan janganlah kamu melupakan bahagiamu dari (kenikmatan) duniawi dan berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik, kepadamu, dan janganlah kamu berbuat kerusakan di (muka) bumi.

Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan”(QS. Al-Qashas ayat 77).

“Ilmu tanpa amal adalah kegilaan, dan amal tanpa ilmu adalah kesia-siaan.”

Imam Ghazali

## KATA PENGANTAR

*Assalamualaikum Warahmatullahi Wabarakatuh*

*Alhamdulillahillobbil 'Alamin Assholatu Wassalamu Ala Asyrofil Ambiya Wal Mursalin, Wa 'Ala Wasohbihi Ajmain Amma Ba'du.*

Segala puji bagi Allah SWT. Yang Maha Pengasih lagi Maha Penyayang atas kehadiran dan berkat kemudahan yang telah diberikan oleh-Nya, sehingga penulis dapat menyelesaikan kegiatan Tugas Akhir yang berjudul “**Perancangan Sistem Otomatisasi Pada Peternakan Ikan Berbasis Microcontroller Arduino IDE Untuk Meningkatkan Hasil Panen Ikan**” sesuai dengan tenggat waktu yang telah diberikan meskipun dengan adanya hambatan selama proses pengerjaannya. Tanpa pertolongan-Nya penulis tidak dapat menyelesaikan Tugas Akhir ini dengan baik yang disusun sebagai memenuhi syarat kelulusan pada Teknik Industri. Maka dari itu, penulis telah berusaha memberikan yang terbaik dalam pengerjaan laporan ini semaksimal mungkin.

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu selama penyusunan Tugas Akhir ini, baik waktu dan bantuan lainnya. Untuk itu, padakeempatan ini penulis ingin menyampaikan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU. selaku dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM. Selaku Ketua Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Ir. Muchamad Sugarindra, S.T., M.T.I., IPM. Selaku dosen pembimbing Tugas Akhir yang selalu sabar memberikan motivasi, arahan dan bimbingan dalam perjalanan penelitian saya.
4. Bapak Dyogi Miyosa Trianzie selaku pemilik peternakan dan pembudidayaan ikan nila Minaverse Company yang telah bersedia meluangkan waktu, pemikiran dan tempat untuk proses penelitian ini.
5. Bapak (Alwie) dan Ibu (Anisah) selaku orang tua penulis yang selalu mendoakan dan mendukung penulis sehingga dapat berjuang dan



menyelesaikan Tugas Akhir. Juga untuk saudara kandung penulis Lia Fadhilah, Fitriyah, dan Surayah yang senantiasa memberikan dukungan moril dan motivasi yang diberikan.

6. Masyarakat Desa Sambilegi lor, Maguwoharjo, Depok, Sleman, DI Yogyakarta selaku daerah tempat penelitian yang senantiasa memberikan selamat serta pengalaman baru dalam dunia peternakan ikan.
7. Teman-teman Teknik Industri Angkatan 2019 yang telah memberikan semangat dalam pelaksanaan Tugas Akhir
8. Jajaran pengurus HMTI LEM FTI UII periode 2021/2022 yang selalu membantu dan memberikan pengalaman terbaik selama jenjang kuliah ini.
9. Hibrizi Fatih Ardiwal, Rayhan Rahmanda Putra, Muhammad Arfin S, Okyadi Fatir, dan Rillo Pambudi selaku sahabat penulis yang selalu meluangkan waktu untuk bertukar pikiran dan menghabiskan waktu dalam penyelesaian penelitian Tugas Akhir
10. Raiful Mu'afiq, Akmal Maulana, Farhan Hafizh, Alief Furqon, dan Yundri Nugraha selaku sahabat penulis dari semester satu hingga sekarang dan membantu penulis melewati masa-masa bersama.
11. Pihak lain yang tidak bisa disebutkan namanya satu-persatu yang telah ikut berperan dan mendoakan saya dalam penyelesaian tugas akhir ini, Terima kasih banyak, semoga Allah SWT memberikan balasan yang setimpal.

Harapannya semoga laporan Tugas Akhir ini dapat menambah pengetahuan, baik bagi pembaca maupun penulis sendiri. Dalam penulisan laporan Tugas Akhir ini, tentunya masih jauh dari kata sempurna. Penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan tugas ini. Akhir kata semoga laporan Tugas Akhir ini dapat digunakan sebagaimana mestinya serta bermanfaat bagi penulis khususnya dan bagi para pembaca yang berminat. Aamiin Yaa Robbal 'Aalamiin.

***Wassalamu'alaikum Warrahmatullahi Wabarakatuh.***

Yogyakarta, 01 Juli 2023



**Muhammad Abdullah Salam**

**19522254**

## ABSTRAK

Ikan adalah organisme tertinggi dalam suatu perairan, kelangsungan hidup ikan dipengaruhi oleh berbagai faktor penting salah satunya adalah air tempat ikan hidup dan berkembang biak. Ikan hidup di lingkungan perairan baik di perairan tawar, payau, ataupun asin. Kondisi lingkungan yang tidak stabil sangat berpengaruh terhadap perubahan organisme akuatik baik secara langsung maupun tidak langsung. Peternakan ikan ataupun budidaya ikan belum terlalu memperhatikan kondisi lingkungan yang dapat berakibat terhadap produksi ikan ataupun pertumbuhan dari ikan. Dalam hal tersebut seperti kondisi pH air, kadar oksigen, dan pengecekan suhu yang masih manual dan berdasarkan intuisi pembudidaya. Metode yang digunakan adalah *prototyping* untuk membuat alat pemantauan kualitas air. Pemantauan pH air berkisar 7-8, suhu berkisar 28-30°C dan kondisi oksigen terlarut 6-8,5 ppm dapat meningkatkan hasil panen ikan. Penulisan ini bertujuan untuk membantu proses pengelolaan peternakan ikan dengan memanfaatkan *microcontroller* Arduino IDE dan dapat memperbesar tingkat efektivitas, efisiensi dan membuat pertumbuhan ikan menjadi produktif.

**Kunci:** Ikan, *Prototyping*, pH air, Kadar Oksigen, suhu, *Microcontroller*, Otomatisasi, Arduino IDE

## DAFTAR ISI

SURAT PERNYATAAN KEASLIAN.....	ii
SURAT KETERANGAN PELAKSANAAN TA.....	iii
LEMBAR PENGESAHAN PEMBIMBING.....	iv
LEMBAR PENGESAHAN DOSEN PENGUJI.....	v
HALAMAN PERSEMBAHAN.....	vi
MOTTO.....	vii
KATA PENGANTAR.....	viii
ABSTRAK.....	xi
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR.....	xvi
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	4
1.5 Batasan Penelitian.....	4
1.6 Sistematika Penulisan.....	4
BAB II.....	6
TINJAUAN LITERATUR.....	6
2.1 Kajian Induktif.....	6
2.2 Landasan Teori.....	21
2.2.1 Ikan nila.....	21
2.2.2 Pengolahan kualitas air.....	22
2.2.3 Mikrokontroler ESP-32.....	24
2.2.4 Sensor Analog pH KB2004.....	26
2.2.5 Sensor suhu DS18B20.....	27
2.2.6 Sensor oksigen.....	29

2.2.7	Modul relay .....	30
2.2.8	Bahasa pemrograman C pada Arduino.....	32
2.2.9	Arduino IDE.....	35
2.2.10	Prototyping .....	35
BAB III.....		37
METODE PENELITIAN .....		37
3.1	Objek Penelitian .....	37
3.2	Subjek Penelitian .....	37
3.3	Jenis Data Penelitian.....	37
3.4	Alur Penelitian.....	38
BAB IV .....		41
PEMBANGUNAN SISTEM.....		41
4.1	Pengumpulan Data.....	41
4.2	Profil Minaverse Company.....	41
4.3	Requirements Gathering .....	42
4.3.1	Kebutuhan pengguna .....	42
4.4	<i>Quick Design</i> .....	43
4.4.1	Peralatan penelitian.....	43
4.4.2	Diagram blok perancangan sistem .....	50
4.5	Build Prototype.....	52
4.5.1	Persiapan awal Blynk Cloud .....	52
4.5.2	Pembangunan Dissolved Oxygen Sensor.....	54
4.4.3	Pembangunan Suhu ( <i>Temperature</i> ).....	59
4.4.4	Pembangunan sistem pH .....	62
4.4.5	Pembangunan Sistem <i>Relay pump</i> dan <i>Relay</i> kincir .....	66
4.6	Evaluasi .....	71
4.6.1	Evaluasi <i>prototype</i> .....	71
4.6.2	Perbaikan <i>prototype</i> .....	72
4.7	Pembuatan dan Perakitan Produk .....	74
4.7.1	Desain skematik keseluruhan .....	74
4.7.2	<i>Script</i> Keseluruhan .....	75
4.7.3	Hasil Pemasangan <i>Prototype</i> .....	89

BAB V.....	92
PENGUJIAN DAN PEMBAHASAN SISTEM .....	92
5.1    Pengujian Sistem .....	92
5.1.1    Uji Fungsionalitas .....	92
5.2    Pembahasan Sistem .....	94
5.2.1    Sistem sensor pH dan air <i>pump</i> .....	94
5.2.2    Sistem sensor <i>dissolved oxygen</i> dan kincir .....	95
5.2.3    Sistem sensor suhu .....	96
5.3    Tata Letak Sensor .....	97
BAB VI .....	101
PENUTUP.....	101
6.1    Kesimpulan.....	101
6.2    Saran .....	102
DAFTAR PUSTAKA .....	104
LAMPIRAN .....	A-1
A-Hasil Perancangan Skematik .....	A-1
B-Hasil Tampilan Blynk <i>Cloud</i> .....	A-1
C-Hasil Pembacaan LCD.....	A-2
D-Pengambilan Data Lokasi Peletakan Sensor .....	A-3
E-Pengkodean Arduino IDE.....	A-4
F-Diskusi Dengan Pemilik Minaverse Company .....	A-7

## DAFTAR TABEL

Tabel 2. 1 Kajian Induktif .....	10
Tabel 2. 2 Parameter Kualitas Air .....	22
Tabel 2. 3 Spesifikasi ESP-32 .....	24
Tabel 2. 4 Spesifikasi Sensor Analog pH KB2004 .....	26
Tabel 2. 5 Spesifikasi Sensor Suhu DS18B20 .....	28
Tabel 2. 6 Spesifikasi Sensor DO.....	30
Tabel 4. 1 Kebutuhan Pengguna .....	42
Tabel 4. 2 <i>Hardware</i> .....	43
Tabel 4. 3 <i>Software</i> .....	49
Tabel 4. 4 Peralatan Sensor DO .....	54
Tabel 4. 5 Sambungan Skematik Sensor DO .....	55
Tabel 4. 6 <i>Script Dissolved Oxygen Sensor</i> .....	56
Tabel 4. 7 Peralatan Suhu.....	59
Tabel 4. 8 Sambungan Skematik Sensor Suhu.....	61
Tabel 4. 9 <i>Script Sensor Suhu</i> .....	61
Tabel 4. 10 Peralatan Sistem pH .....	62
Tabel 4. 11 Sambungan Skematik Sistem pH.....	63
Tabel 4. 12 <i>Script Sensor pH</i> .....	64
Tabel 4. 13 Peralatan <i>Relay</i> .....	67
Tabel 4. 14 Sambungan Sistem <i>Relay</i> .....	67
Tabel 4. 15 <i>Script Sistem Relay</i> .....	69
Tabel 4. 16 Sambungan Keseluruhan.....	75
Tabel 4. 17 <i>Script Keseluruhan</i> .....	76

## DAFTAR GAMBAR

Gambar 2. 1 Ikan Nila .....	21
Gambar 2. 2 Mikrokontroler ESP-32 .....	24
Gambar 2. 3 Pin ESP-32 .....	25
Gambar 2. 4 Sensor pH air .....	27
Gambar 2. 5 Sensor Suhu DS18B20 .....	28
Gambar 2. 6 Sensor <i>Dissolved Oxygen</i> .....	29
Gambar 2. 7 Modul Relay .....	30
Gambar 2. 8 Struktur Komponen Relay .....	31
Gambar 2. 9 Tahapan <i>Prototyping</i> .....	36
Gambar 3. 1 Alur Penelitian.....	38
Gambar 4. 1 Minaverse Company.....	41
Gambar 4. 2 ESP-32.....	44
Gambar 4. 3 Laptop.....	44
Gambar 4. 4 USB Kabel.....	44
Gambar 4. 5 Kabel Jumper.....	45
Gambar 4. 6 Adaptor.....	45
Gambar 4. 7 LCD I2C .....	45
Gambar 4. 8 Analog DO .....	46
Gambar 4. 9 Relay Modul .....	46
Gambar 4. 10 Sensor DS18B20 .....	46
Gambar 4. 11 I2C Modul .....	47
Gambar 4. 12 pH Sensor .....	47
Gambar 4. 13 Analog <i>Signal Isolator</i> .....	48
Gambar 4. 14 <i>Air Pump</i> .....	48
Gambar 4. 15 <i>Stop Kontak</i> .....	49
Gambar 4. 16 Arduino IDE.....	49
Gambar 4. 17 Blynk Cloud .....	50
Gambar 4. 18 Diagram Blok Awal.....	50
Gambar 4. 19 <i>Log In</i> .....	52
Gambar 4. 20 <i>Create New Template</i> .....	53



Gambar 4. 21 <i>New Device</i> .....	53
Gambar 4. 22 <i>Templates Menu</i> .....	54
Gambar 4. 23 Skematik Sensor DO .....	55
Gambar 4. 24 <i>Virtual Pin Datastream</i> .....	55
Gambar 4. 25 <i>Widget Box</i> .....	56
Gambar 4. 26 <i>Slider Settings</i> .....	56
Gambar 4. 27 Skematik Suhu .....	60
Gambar 4. 28 Skematik sistem pH .....	63
Gambar 4. 29 <i>Virtual Pin Datastream</i> .....	64
Gambar 4. 30 <i>Widget</i> .....	64
Gambar 4. 31 Skematik Sistem <i>Relay</i> .....	67
Gambar 4. 32 <i>Virtual Pin Datastream Kincir</i> .....	68
Gambar 4. 33 <i>Virtual Pin Datasream pH</i> .....	68
Gambar 4. 34 <i>Widget</i> .....	69
Gambar 4. 35 Evaluasi <i>Prototype</i> .....	72
Gambar 4. 36 Unduh Blynk IoT .....	73
Gambar 4. 37 <i>Login</i> .....	73
Gambar 4. 38 Tampilan Aplikasi .....	73
Gambar 4. 39 <i>Watermeter</i> .....	74
Gambar 4. 40 Skematik Keseluruhan .....	74
Gambar 4. 41 Hasil <i>Prototype</i> .....	90
Gambar 4. 42 Indikator Alat .....	90
Gambar 4. 43 Indikator Blynk <i>Cloud</i> .....	91
Gambar 5. 1 Ukuran Kolam Ikan .....	97
Gambar 5. 2 Skematik Pengambilan Air di Permukaan .....	98
Gambar 5. 3 Skema peletakan sensor .....	99

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Ikan Nila (*Oreochromis Niloticus*) merupakan ikan air tawar, ikan yang dapat hidup di daerah dataran rendah yang bersifat (*eurihaline*) hidup di perairan bersalinitas tinggi. Ikan nila juga termasuk ikan yang dikenal ikan konsumsi untuk masyarakat dan tingkat permintaannya sangat tinggi sehingga konsumen juga mudah didapatkan (Suriana et al., 2021). Ikan nila merupakan salah satu jenis ikan yang banyak dibudidayakan oleh pembudidaya karena sangat diminati masyarakat untuk dikonsumsi atau sebagai bahan pangan. Menurut Direktorat Jenderal Perikanan budidaya ikan nila terus mengalami peningkatan, produksi tahun 2016 sebesar 1.114.156 ton, sedangkan tahun 2017 meningkat menjadi 1.265.201 ton. Produksi hingga triwulan III tahun 2018 tercatat 579.688 ton. Produksi ikan nila yang meningkat sejalan dengan permintaan pasar ikan nila yang mengalami kenaikan di setiap tahunnya (Salsabila, 2018). Kualitas air menjadi faktor utama yang mempengaruhi pertumbuhan ikan yang dibudidayakan (A. Widodo et al., 2021).

Air merupakan sumber daya alam yang sangat penting bagi kelangsungan hidup ikan. Sebagian besar ikan sangat peka terhadap perubahan lingkungan perairan, sehingga kualitas air yang digunakan sebagai tempat hidup sangatlah penting. Berbagai proses metabolisme yang terjadi di dalam tubuh ikan yang berperan penting dalam peningkatan dan kelangsungan hidup dipengaruhi oleh berbagai faktor fisik kualitas air (E. Efendi, 2015). Parameter kualitas air yang sering diamati antara lain suhu, pH (*power of Hydrogen*), DO (*Dissolve Oxygen*) (Marlina et al., 2016).

Kualitas air sangat berpengaruh dengan memperhatikan parameter diatas sehingga dapat memberikan peningkatan dari jumlah produksi ikan, dapat mempercepat pertumbuhan yang optimal dan kelangsungan hidup yang tinggi (Azhari & Tomaso, 2018). Suhu optimal dalam budidaya ikan air tawar adalah 28-32°C (Mas'ud, 2014). Kadar oksigen terlarut pada kolam yang apabila berkisar antara 1-5 ppm mengakibatkan pertumbuhan ikan menjadi lambat sedangkan oksigen terlarut yang kurang dari 1 ppm dapat bersifat toksik bagi sebagian besar

spesies ikan (Ratannanda, 2011). Oksigen terlarut untuk ikan nila optimum 7 ppm sesuai yang diungkapkan oleh (Effendi, 2003). Menurut (Tebbut, 1998), bila kadar oksigen terlarut rendah menyebabkan meningkatnya toksisitas pada hewan, namun bila kadar oksigen terlarut tinggi atau optimal konsentrasi amonia tidak terlalu besar. Selain kadar oksigen terlarut dan suhu, pH juga mempengaruhi toksisitas suatu senyawa. Menurut (Mas'ud, 2014) pH stabil berkisar pada 6-7 yang dapat menunjang kelangsungan hidup ikan.

Permasalahan yang harus dihadapi oleh peternak ikan secara tradisional yakni dari faktor alam yang tidak menentu sehingga ikan tidak dapat beradaptasi secara maksimal dan bersiko kematian. Cuaca yang tidak menentu sangat mempengaruhi kualitas air. Dengan perubahan cuaca yang signifikan dapat mengakibatkan kadar pH, suhu, dan oksigen menjadi tidak stabil atau dalam kondisi tidak normal. Pada prosesnya, para pembudidaya menggunakan cara manual untuk melakukan pengecekan kualitas air. Sehingga suhu air yang tidak stabil mengakibatkan bibit ikan nila banyak yang mati. Selain itu, pemberian probiotik yang bertujuan untuk memperbaiki dan menjaga kualitas air secara berlebihan atau kurang akan berdampak kepada pH air di dalam kolam. Penggunaan kincir oksigen yang berlebihan mengakibatkan pemborosan pemakaian listrik.

Seiring dengan perkembangan teknologi, penggunaan sistem automasi pada peternakan ikan nila menjadi semakin populer. Pentingnya menjaga kualitas air agar permasalahan yang terjadi dikarenakan faktor alam dan kelalaian manusia dapat teratasi. Sistem automasi dapat membantu peternak dalam memantau dan mengontrol kualitas air secara terus-menerus dan akurat, sehingga dapat meningkatkan efisiensi dalam pengelolaan peternakan dan kualitas ikan yang dihasilkan.

Jika para pembudidaya ikan bisa melakukan monitoring terhadap perubahan suhu dari jarak jauh dan dapat memberikan cairan probiotik sesuai dengan kadar pH dan suhu yang sesuai maka akan meminimalisir kematian bibit ikan dan mengurangi biaya dari probiotik tersebut, dan begitu juga dengan kincir oksigen yang dapat hidup sesuai kadar oksigen yang dibutuhkan bukan secara berkala. Sistem monitoring jarak jauh ini menggunakan jaringan internet (IoT) *Internet of Things*.

Pada budidaya ikan nila pada Minaverse Company pada bulan januari 2023 memiliki hasil panen sejumlah 2 kwintal dan memiliki perkembangan pada bulan april 2023 memiliki hasil

panen sejumlah 3 kwintal. Faktor utama peningkatan tersebut adalah pemakaian kincir oksigen untuk membuat oksigen tetap terkontrol. Menurut wawancara langsung dengan pemilik Minaverse Company, kualitas air sangat berpengaruh terhadap kelangsungan hidup ikan baik suhu, pH, maupun DO. Ikan akan mengalami gangguan seperti ikan tidak sehat, tidak berkembang dengan baik dan kematian jika kekurangan 3 faktor tersebut.

*Internet of things* (IoT) merupakan perangkat atau alat *hardware* dapat terkoneksi jaringan internet dalam tujuan memperluas koneksi jaringan dengan perangkat keras secara menyeluruh (Muzawi et al., 2018). *Internet of things* (IoT) digunakan sebagai mengendalikan peralatan yang bisa dioperasikan melalui jaringan dan dapat diterapkan melalui mobile dari jarak jauh sehingga dapat mempermudah untuk memantau perangkat elektronik tersebut (Y. Efendi, 2018). Komponen elektronik sebagai pendukung dan dapat bekerja sesuai dengan kegunaannya.

Perkembangan sistem monitoring sudah sangat pesat dengan menggunakan perkembangan informasi dari monitor berupa gambar atau tulisan, pengiriman informasi melalui pesan singkat dengan memanfaatkan interface yang telah dibuat. Dengan memanfaatkan IoT sangat berguna dalam hal monitoring dan aplikasi automasi untuk meringankan pekerja dan memangkas biaya operasional.

Berdasarkan masalah tentang perubahan suhu, pH, dan kincir oksigen pada kolam ikan yang mengakibatkan matinya bibit ikan, berkurangnya tingkat produksi ikan, dan munculnya bakteri pada kolam ikan, maka peneliti ingin merancang sistem automasi untuk melakukan monitoring jarak jauh. Jika terjadi perubahan seperti di atas maka dapat terdeteksi secara langsung dan diberikan penanganan secara otomatis.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas maka rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana Perancangan Sistem Otomatisasi Pada Peternak Ikan Berbasis Mikrokontroler Arduino IDE Untuk Meningkatkan Hasil Panen Ikan?
2. Bagaimana sistem otomatisasi dapat melakukan *controlling* dan *monitoring* pH, suhu dan oksigen terlarut pada peternakan ikan?

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Melakukan desain sistem otomatisasi untuk pendeteksi oksigen terlarut, pH dan suhu air
2. Melakukan desain sistem pengendalian kadar oksigen terlarut, pH dan suhu air

#### **1.4 Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi Peneliti
  - a. Mampu membuat sistem automasi IoT dalam dunia peternakan ikan
  - b. Mampu meminimalisir pengeluaran biaya terkait operasional pada peternakan ikan
  - c. Menambah pengetahuan dalam dunia IoT dan dapat mengimplementasikan menjadi sistem nyata
2. Bagi Pembaca

Dapat menjadi sumber baca dan referensi terkait IoT khususnya pada peternakan ikan untuk penelitian selanjutnya.
3. Bagi Perusahaan
  - a. Dapat membantu peternak ikan dalam mendapatkan hasil produksi yang maksimal
  - b. Dapat memudahkan para pekerja dalam melakukan monitoring dari jarak jauh
  - c. Sebagai bahan pertimbangan peternak ikan apakah dengan menggunakan IoT dapat menghasilkan output yang lebih baik

#### **1.5 Batasan Penelitian**

Batasan dalam melakukan penelitian ini adalah sebagai berikut:

1. Peningkatan pertumbuhan ikan belum tervalidasi
2. Melakukan penelitian langsung terhadap peternakan ikan untuk mendapatkan data dan melakukan sistem automasi mengenai pH air, suhu air dan automasi kincir oksigen pada kolam.
3. Kondisi kolam ikan yang diteliti adalah kolam ikan nila.
4. Tidak melakukan perhitungan hasil panen secara langsung dikarenakan kendala waktu.

#### **1.6 Sistematika Penulisan**

Dalam penelitian ini terdapat sistematika penulisan yang tersusun dalam enam bab, berikut merupakan isi dari tiap bab:

### **BAB I PENDAHULUAN**

Bab 1 berisikan mengenai latar belakang penelitian yaitu terkait pengelolaan kualitas air kolam pada peternakan nila menggunakan sistem IoT. Melakukan monitoring terkait suhu, pH dan oksigen di dalam air menggunakan mikrokontroler ESP-32 yang telah dihubungkan dengan sensor-sensor yang ada. Selanjutnya terdapat rumusan masalah, tujuan penelitian, manfaat dari penelitian, serta batasan dalam penelitian ini.

## **BAB II KAJIAN LITERATUR**

Bab 2 berisikan landasan teori yang dijadikan sebagai referensi dan landasan dalam melakukan penelitian ini. Pada kajian literatur berisikan dua pembahasan yaitu kajian induktif yang berisikan penelitian terdahulu yang memiliki kesamaan penelitian dengan peneliti lakukan seperti penggunaan mikrokontroler ESP-32 dan sistem automasi. Kajian deduktif yang berisikan teori-teori yang berkaitan dengan penelitian ini berdasarkan penjelasan para ahli atau penelitian sebelumnya.

## **BAB III METODE PENELITIAN**

Pada bab 3 memuat kerangka rencana penelitian, penjelasan mengenai objek penelitian, subjek penelitian, jenis data penelitian yang meliputi data primer dan sekunder, dan yang terakhir adalah alur penelitian sebagai acuan rangkaian penelitian dari awal hingga selesai.

## **BAB IV PEMBANGUNAN SISTEM**

Pada bab 4 berisikan rancangan pembangunan sistem dimulai dari pengenalan alat yang digunakan, spesifikasi alat yang digunakan dan skematik rangkaian sistem automasi. Selain itu pada bab 4 juga menjelaskan bagian *pseudo code* atau bahasa pemrograman yang digunakan dan dijelaskan maksud dari bahasa tersebut.

## **BAB V PENGUJIAN SISTEM DAN PEMBAHASAN**

Pada bab 5 dilakukan pengujian sistem yang telah dibahas pada bab 4 meliputi sensor yang digunakan apakah sudah sesuai dan dapat berjalan sesuai dengan bahasa pemrograman yang telah diatur. Setelah itu dilakukan pembahasan mengenai hasil pengujian sistem tersebut.

## **BAB VI KESIMPULAN DAN SARAN**

Pada bab 6 ini berisikan kesimpulan yang menjawab rumusan masalah yang ada pada bab 1 selain itu pada bab 6 berisikan saran serta rekomendasi untuk penelitian selanjutnya agar dapat menjadi acuan dan bahan evaluasi bagi peneliti selanjutnya.

## BAB II

### TINJAUAN LITERATUR

Tinjauan pustaka ini mencakup landasan teori yang digunakan dalam penelitian. Dalam tinjauan pustaka ini dibagi menjadi dua bagian, yaitu kajian induktif dan kajian deduktif. Penelitian induktif menjelaskan penelitian terdahulu yang telah dilakukan oleh para peneliti terdahulu yang berkaitan dengan penelitian tersebut. Sedangkan untuk penelitian deduktif merupakan penjelasan dari berbagai teori yang digunakan dalam penelitian tersebut.

#### 2.1 Kajian Induktif

Pada penelitian yang telah dilakukan oleh (T. Widodo et al., 2020) yaitu penggunaan mikrokontroler arduino uno R3 pada sirkulasi air. Pada budidaya sirkulasi bioflok, biasanya pembudidaya akan menghidupkan mesin aerator sebagai penyuplai oksigen selama 24 jam. Karena tidak adanya mesin pompa air untuk melakukan sirkulasi dan pembersihan kotoran pada kolam maka hal ini mengakibatkan pembudidaya harus melakukan pembuangan air kolam secara manual. Dengan adanya sensor dan mikrokontroler, mempermudah peneliti untuk membuat alat sirkulasi air otomatis. Dengan menggunakan sensor turbidity sebagai indikator kekeruhan air dan Pompa air yang berfungsi untuk mengaliri sirkulasi air akan bekerja sesuai dengan tingkat kekeruhan air.

Penelitian selanjutnya adalah perancangan sistem kendali pada budidaya ikan nila berbasis mikrokontroler dan teknologi data *logger* (Saputra et al., 2022) dengan melakukan pencatatan secara otomatis kualitas air kolam berupa nilai pH, suhu air, kekeruhan air, dan ketinggian air, serta memberi pakan ikan secara otomatis. Hasil dari penelitian ini adalah sistem mampu melakukan kendali pada parameter kualitas air dengan baik sesuai standar yang telah ditentukan.

Menurut (Sampebatu & Kamolan, 2021) untuk kondisi lingkungan seiring meningkatnya permintaan ikan maka harus diiringi dengan penyediaan benih ikan yang berkualitas, di mana hal ini sangat ditentukan oleh kondisi air kolam seperti suhu, kejernihan dan ketinggian air. Penelitian ini membuat rancang bangun suatu sistem yang mengatur kondisi suhu optimal untuk kolam pembibitan ikan yaitu 25°-30° celcius. Apabila suhu air kolam dibawah 25°

celcius maka sistem mengaktifkan heater sampai pada suhu 28° celcius. Untuk ketinggian air yang baik untuk kolam pembibitan adalah berkisar antara 40-60 centimeter.

Penelitian yang dilakukan oleh (Barus et al., 2018) mengenai kadar pH di ukur dengan menggunakan sensor pH E-201-C dan suhu diukur dengan sensor DS18B20. Proses kontrol pH di lakukan dengan menambahkan cairan pH *up* dan pH *down* dan direalisasikan dengan katupsolenoid. Tujuan pembuatan sistem ini adalah untuk mengontrol nilai pH air dalam akuarium dan memberi informasi tentang suhu air. Ke dalam sistem ini telah diinput nilai standar pH dan suhu pada masing-masing jenis ikan. Sistem ini akan bekerja secara otomatis untuk menyesuaikan lingkungan hidup ikan hias sesuai dengan kebutuhannya masing-masing.

Penelitian yang dilakukan oleh (Sholikin et al., 2021) ada beberapa faktor yang harus diperhatikan dalam pemeliharaan ikan nila, antara lain kondisi lingkungan dan kualitas air yang baik. Untuk menjaga kondisi kualitas air yang baik, air dilakukan pemantauan dan pengendalian mutu. Seiring dengan perkembangan teknologi, hal ini penelitian bertujuan untuk membuat prototipe otomatisasi untuk tingkat pH dan tingkat air kolam yang dipantau *online* melalui Thinger.io.

Menurut (Kustija & Andika, 2021) tentang sistem pengelolaan ikan memiliki peran penting dalam budidaya ikan. Salah satu aspek dari pengelolaan ikan adalah kualitas air yang meliputi beberapa hal seperti suhu, pH, kadar oksigen dan juga pakan. Sejauh ini, pemantauan air kualitas dan pemberian pakan ikan selama ini dilakukan secara manual. Penelitian ini bertujuan untuk merancang sistem kontrol pemantauan untuk kadar oksigen, pH, suhu dan otomatis pemberian makan berdasarkan IoT. Pembacaan data dari sensor dan juga RTC akan dilakukan diteruskan oleh mikrokontroler ke server untuk ditampilkan kepada pengguna. Ini sistem diotomatisasi dengan aktuator berupa aerator dan motor, sehingga pemberian pakan dan penambahan kadar oksigen pada kolam akan dilakukan secara otomatis oleh mikrokontroler.

Makalah yang ditulis oleh (Gokulanathan et al., 2019) membahas tentang sistem pemantauan kualitas air melalui sensor nirkabel jaringan. Karena pesatnya pembangunan dan urbanisasi, kualitas air semakin menurun dari tahun ke tahun, dan itu mengarah pada penyakit yang terbawa air, dan itu menimbulkan dampak yang buruk. Air memiliki peran penting dalam masyarakat manusia kita dan India 65% air minum berasal dari bawah tanah sumber,



sehingga wajib untuk memeriksa kualitas air. Pada model ini digunakan untuk menguji air sampel dan melalui data itu menganalisis kualitas air. Kertas ini memberikan kekuatan solusi yang efisien dan efektif dalam domain pemantauan kualitas air juga menyediakan alarm ke remote user, jika terjadi penyimpangan parameter kualitas air.

Menurut (Susanti et al., 2022) mengenai sistem IoT murah dapat memantau parameter sistem aquaponik seperti suhu, pH dan TDS melalui platform android. Desain perangkat keras menggunakan beberapa sensor yang terintegrasi yaitu sensor suhu air tipe DS18B20, tipe pH SKU: SEN0161, sensor tipe TDS SKU: SEN0244. Perancangan perangkat lunak Android menggunakan Bahasa C untuk pemrograman. Penulis menggunakan tambahan NodeMCU ESP8266 yang dapat mengirimkan data ke *smartphone* menggunakan koneksi internet. Data dapat diakses menggunakan Android dengan aplikasi Blynk. Biaya yang dibutuhkan untuk mengembangkan perangkat ini tergolong murah yaitu 83,79 USD. Komponen pada perangkat mudah tersedia di pasaran sehingga penggantian suku cadang dapat dilakukan dengan mudah setiap 2-3 tahun sekali.

Menurut (Araneta, 2022) pengujian kualitas air merupakan bagian penting dari pemantauan lingkungan. Ketika kualitas air tidak memadai, itu berdampak pada kehidupan pertanian serta sistem biologis secara keseluruhan. Sistem Pemantauan Air berbasis Arduino adalah sepenuhnya fungsional dan dapat mendeteksi pembacaan sensor dari empat parameter, termasuk suhu, pH, kekeruhan, dan konduktivitas listrik. Berdasarkan hasil percobaan dan uji kegunaan sistem, sekarang dapat dikerahkan. *Gadget* ini juga memiliki ESP32 yang disertakan di dalamnya, yang dapat mentransfer data WIFI dan Bluetooth. Alat tersebut praktis dan dapat digunakan karena memenuhi spesifikasi. Saat dipasangkan dengan Aplikasi Seluler Kualitas Air, perangkat ini kompatibel dengan daya 5 volt perbankan dan bekerja dengan baik.

Penelitian yang dilakukan oleh (Pasika & Gandla, 2020) mengenai perkembangan komunikasi nirkabel menciptakan kemampuan sensor baru. Perkembangan terkini di lapangan jaringan sensor sangat penting untuk aplikasi lingkungan. *Internet of Things* (IoT) memungkinkan koneksi antar berbagai perangkat dengan kemampuan untuk bertukar dan mengumpulkan data. Karena air merupakan salah satu kebutuhan pokok manusia kelangsungan hidup, perlu memasukkan beberapa mekanisme untuk memantau kualitas air dari waktu ke waktu. Oleh karena itu, ada kebutuhan untuk memastikan pasokan dimurnikan

air minum bagi masyarakat baik di kota maupun desa. Pemantauan Kualitas Air (WQM) adalah cara yang hemat biaya dan sistem efisien yang dirancang untuk memantau kualitas air minum yang memanfaatkan teknologi *Internet of Things* (IoT).

Menurut (Manoj et al., 2022) salah satu isu utama yang dihadapi dunia adalah sumber air bersih yang semakin berkurang cepat karena perubahan iklim, kontaminasi, dan polusi. Makhluk hidup yang paling terpengaruh adalah bentuk kehidupan bawah air karena mereka akhirnya mengambil racun ini dan dengan demikian rentan terhadap kematian terus memeriksa kualitas air. Salah satu teknologi baru, *Internet of Things* (IoT) menunjukkan janji besar yang terkait dengan bidang ini.

Menurut (Md. Mahbubur Rahman, 2020) untuk pengamatan kualitas air yang tepat, kami membangun sistem pengukuran kualitas air berbasis IoT yang berbiaya rendah dan berkelanjutan. Dalam pekerjaan ini kami mengusulkan perangkat antarmuka sensor pintar yang dapat merasakan parameter kualitas air dan menghasilkan data secara efektif dalam sistem *online* untuk menunjukkan ukuran parameter kualitas air secara *real-time*. Ini mengintegrasikan pengukuran kualitas air minum dengan berbagai jenis sensor. Sensor-sensor ini berhubungan dengan Arduino dengan tujuan memantau parameter kualitas air. Untuk mentransmisikan nilai, kami membuat komunikasi serial antara Arduino dan NodeMCU yang akan menampilkan data pada sistem *online* (antarmuka web).

Penelitian yang dilakukan oleh (Endut et al., 2019) diantara parameter yang dipantau untuk menentukan kualitas air adalah tingkat PH, suhu dan kekeruhan. Parameter ini penting untuk memastikan air berada pada tingkat optimal untuk budidaya ikan. Oleh karena itu, Sistem Pemantauan Air *Real-Time* diusulkan untuk mengatasi masalah tersebut dengan memanfaatkan teknologi *Internet of Things* (IoT) untuk mengotomatisasi proses pemantauan air. Sensor digital digunakan untuk mengukur kadar PH air, suhu dan juga kekeruhan dan sensor ini dikontrol menggunakan papan Arduino.

Menurut (Aziz et al., 2020) diperlukan tindakan identifikasi dan pemantauan yang membutuhkan media menggunakan alat yang meliputi sensor yang berisi pH air yang berasal dari kandungan zat asam dalam air, kekeruhan atau kandungan benda padat di dalam air, dan air suhu. Pengujian untuk melihat hasil monitoring menggunakan aplikasi Android yang terhubung dengan server. Dari hasil tersebut dapat disimpulkan bahwa data yang ditampilkan

pada aplikasi dapat berjalan dengan lancar, nilai yang ditampilkan dapat dilihat secara *realtime*.

Menurut (Ismail et al., 2020) saat membudidayakan organisme air, kualitas air merupakan faktor penting. Itu tergantung terutama pada berbagai parameter seperti oksigen terlarut, amonia, pH, suhu, garam, nitrat, dan karbonat. Pengujian ini dilakukan oleh pembudidaya tambak secara manual atau oleh peneliti melalui pengujian laboratorium, di mana tidak ada pengujian nyata. pemantauan waktu. Saat ini, dengan kemajuan *Internet of Things* (IoT) dan sensor, teknologi ini mencapai tingkat dasar dengan penerapannya di bidang pertanian dan akuakultur. Sangat diinginkan untuk memiliki sistem otomatis berbasis IoT yang dapat terus mengukur dan memantau tambak dengan pemanfaatan sumber daya yang optimal. Makalah ini mengusulkan sebuah model sistem pengukuran dan pemantauan *real-time* berbasis teknologi IoT untuk mengukur dan memantau parameter air di tambak. Kami menggunakan metode kualitatif untuk menyelidiki persepsi, pengalaman, dan gagasan petani tambak untuk mengusulkan model. Dengan bantuan model yang diusulkan, kualitas air dapat terus diukur dan dipantau untuk memastikan pertumbuhan dan kelangsungan hidup ikan di dalam kolam. Akibatnya, tindakan pencegahan dapat diambil tepat waktu untuk meminimalkan kerugian dan meningkatkan hasil panen ikan.

Berikut pada tabel 2.1 merupakan kajian induktif dari penelitian sebelumnya:

Tabel 2. 1 Kajian Induktif

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
1	Sistem Sirkulasi Air Pada Teknik Budidaya Bioflok Menggunakan Mikrokontroler Arduino Uno R3	Tri Widodo, Bambang Irawan, Agung Prastowo, Ade Surahman (2020)	Sirkulasi, Teknologi Bioflok	Sistem otomatis pada bioflok dapat mengatur kejernihan air dengan menggunakan sensor <i>turbidity</i> . Metode pengaturan sirkulasi otomatis pada bioflok terbukti lebih efektif dan

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
				<p>hemat air, dibandingkan dengan metode sirkulasi yang belum otomatis dalam mengatur hidup dan mati pompa. Sistem otomasi bioflok dapat memberikan kemudahan dalam menjalankan suatu sistem serta dapat menjadi alternatif dalam budidaya ikan di lahan terbatas.</p>
2	<p>Perancangan Sistem Kendali Pada Budidaya Ikan Nila Berbasis Mikrokontroler Dan Teknologi Data Logger</p>	<p>Yogi Alfian Saputra, Ulinnuha Latifa (2022)</p>	<p>Kuantitatif dan deskriptif</p>	<p>Sistem mampu melakukan kendali pada parameter kualitas air dengan baik sesuai standar yang telah ditentukan, yaitu nilai pH 6-8, nilai ketinggian air 50 -60 cm, nilai suhu air 25-32°C, nilai kekeruhan air 1-10 NTU, sistem mampu memberikan pakan ikan sesuai jadwal yang telah</p>

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
				<p>ditentukan yaitu pukul 08:00:00, 12:00:00, dan 15:00:00, dan sistem juga mampu melakukan pencatatan data parameter kualitas air secara otomatis dengan baik melalui fitur data <i>logger</i>.</p>
3	<p>Rancang Bangun Sistem Otomatisasi dan Telekontrol Pengkondisian Air kolam Pembibitan Ikan Nila</p>	<p>Limbran Sampebatu, Aries Kamolan (2021)</p>	<p>Arduino UNO</p>	<p>Hasil simulasi penelitian ini memperlihatkan bahwa rancang bangun <i>prototipe</i> otomatisasi kontrol pengkondisian air, sudah dapat melakukan pengurusan, pengisian, penjernihan dan penghangatan otomatis serta mampu mengirim pesan pada pengguna.</p>
4	<p>Otomatisasi Sistem Kontrol pH dan Informasi Suhu Pada Akuarium Menggunakan</p>	<p>Eltra E. Barus, An dreas Ch. Louk,</p>	<p>Arduino UNO dan Raspberry PI 3</p>	<p>Sistem pengontrolan pH dapat bekerja apabila nilai</p>

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
PI 3	Arduino Uno dan Raspberry	Redi K. Pinggak (2018)		<p>pengukuran pH menyatakan kondisi pH tinggi atau pH rendah dan pada kondisi tersebut air akan mengalir ke tabung penyesuaian pH untuk melakukan kontrol. Apabila nilai pengukuran pH menyatakan bahwa pH tinggi maka cairan Kontrol pH <i>down</i> akan mengalir dan apabila nilai pengukuran pH menyatakan bahwa pH rendah maka cairan kontrol pH <i>up</i> akan mengalir. Sistem ini telah teruji mampu mempertahankan kondisi lingkungan hidup ikan hias dengan hasil pengukuran pH dan suhu adalah 7.48 - 7.8 dan 28.87 –29.55 0C.</p>

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
5	Kontrol Kadar pH dan Ketinggian Air Pada Kolam Ikan Nila Berbasis IOT	Nur Sholikin, Imam Abdul Rozaq, Mohammad Iqbal, Noor Yulita, Dwi Setyaningsih (2021)	IoT dan Thinger.Io	Hasil pengujian sensor <i>error</i> 0,07% dengan akurasi 99,93% untuk ketinggian air, dan <i>error</i> 0,02% dengan akurasi 99,98% untuk pH air. Menguji tampilan LCD antara tampilan web Thinger.io mendapatkan kesalahan 0% dengan akurasi 100%. Untuk menguji seluruh sistem, diperoleh hasil linier dimana sistem mampu menstabilkan pH dan ketinggian air sesuai dengan parameter yang ditentukan yaitu dengan pH antara 6,5 - 8,6 dan ketinggian air minimum 12 cm.
6	<i>Control - Monitoring System Of Oxygen Level, Ph,</i>	JAJA KUSTIJA, FURQON	IoT	Hasil penelitian ini menunjukkan sistem

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
	<i>Temperature And Feeding in Pond Based on Iot</i>	ANDIKA (2021)		dapat bekerja, data suhu, kadar oksigen, pH dapat dipantau melalui server dan makan juga bisa dilakukan
7	<i>A GSM Based Water Quality Monitoring System using Arduino</i>	S. Gokulanathan, P.Manivasagam, N.Prabu, T.Venkatesh (2019)	IoT Arduino	Sistem yang kami usulkan menggunakan penggunaan beberapa sensor untuk mengukur parameter, mengukur kualitas air secara <i>real-time</i> untuk tindakan yang efektif, dan ekonomis, akurat, dan membutuhkan lebih sedikit tenaga kerja.
8	<i>Design and Implementation of Water Quality Monitoring System (Temperature, pH, TDS) in Aquaculture Using IoT at Low Cost</i>	Novita Dwi Susanti, Diang Sagita, Ignatius Fajar Apriyanto, Cahya Edi Wahyu Anggara, Doddy Andy Darmajana, Ari Rahayuningtyas (2022)	IoT	Sensor terintegrasi dapat memantau dari jarak jauh menggunakan android sehingga kualitas air dapat dipantau tanpa di situs. Keuntungan dari



No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
				waktu nyata sistem pemantauan melalui smartphone membawa kesuksesan bagi ikan pertanian. Rekam data parameter air dan simpan di kartu memori sehingga dapat menjadi referensi dan umpan balik untuk pemilik usaha untuk menentukan kualitas air yang sesuai.
9	<i>Design of an Arduino-Based Water Quality Monitoring System</i>	Arvin Anthony S. Araneta, DMT (2022)	IoT Arduino	Perangkat pemantauan Air berbasis Arduino dapat digunakan dan memenuhi persyaratan yang telah ditentukan. Perangkat ini kompatibel dengan 5 <i>volt power bank</i> dan bekerja dengan baik jika dipasangkan dengan <i>Water Quality Mobile</i>
10	<i>Smart water quality</i>	Sathish Pasika,	IoT	Algoritma yang

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
	<i>monitoring system with cost-effective using IoT</i>	Sai Teja Gandla (2020)		efisien dikembangkan secara <i>real-time</i> , untuk melacak kualitas air. Nilai pH yang diukur berkisar antara 6,5 hingga 7,5 untuk air pasokan kota Metropolitan Hyderabad dan 7 hingga 8,5 untuk air tanah. Nilai kekeruhan yang diukur berkisar antara 600 hingga 2000 NTU untuk pasokan air dan tanah kota Metropolitan Hyderabad air.
11	<i>State of the Art Techniques for Water Quality Monitoring Systems for Fish Ponds Using IoT and Underwater Sensors: A Review</i>	M. Manoj, V. Dhilip Kumar, Muhammad Arif, Elena-Raluca Bulai, Petru Bulai, Oana Geman (2022)	IoT	Tinjauan ini mengakui pencapaian utama terkait ukuran kualitas dan indikator keberhasilan mengenai pengukuran

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
12	<i>Internet of Things (IoT) Based Water Quality Monitoring System</i>	Md. Mahbubur Rahman, Chinmay Bapery, Mohammad Jamal Hossain, Zahid Hassan, G.M. Jamil Hossain, Md. Muzahidul Islam (2020)	IoT	kualitatif dan kuantitatif  Pemantauan kualitas air telah menjadi pekerjaan yang diperlukan dalam perlindungan lingkungan. Otomasi pemantauan dan telemetri merupakan tren untuk meningkatkan kemampuan sistem pemantauan kualitas air. Dengan bantuan sensor, kita bisa mengecek kualitas air dengan menggunakan modul Wi-Fi. Karena sistemnya otomatis maka biayanya rendah dan tidak memerlukan tenaga manusia sehingga waktu dan tenaga keduanya hemat. Ini memiliki aplikasi luas dan nilai ekstensi.

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
13	<i>Real-Time Water Monitoring System for Fish Farmers Using Arduino</i>	Nor Adora Endut, Mohammad Fahmi Mohd Fo'ad, Nor Azylia Ahmad Azam, Nor Ashitah Abu Othman, Siti Rahayu Abdul Aziz, Anis Shobirin Abdullah Sani (2019)	IoT Arduino	Tiga sensor digital digunakan untuk mengilustrasikan kecepatan dan akurasi di mana data sensor dapat dibaca. Penggunaan teknologi IoT untuk menyederhanakan dan membuat industri akuakultur menjadi lebih akurat merupakan eksplorasi yang menarik untuk mengumpulkan data yang lebih akurat dan <i>real time</i> dengan biaya yang jauh lebih murah.
14	<i>Monitoring system water pH rate, turbidity, and temperature of river water</i>	F A Aziz, M Sarosa, E Rohadi (2020)	IoT mikrokontroler nodeMCU	Pada suhu mencapai 28,37 OC kriteria sedang, pH air mencapai 9,2 kriteria buruk, sedangkan TDS atau kekeruhan mencapai 192 ppm kriteria buruk. Nilai yang ditampilkan dapat dilihat secara <i>real-time</i> pada

No	Judul Penelitian	Peneliti	Metode	Hasil Penelitian
15	<i>A Proposed Model of Fishpond Water Quality Measurement and Monitoring System based on Internet of Things (IoT)</i>	Rosli Ismail, IoT Kamarudin Shafinah, Kamil Latif (2020)		<p>aplikasi android. Jika ada kendala terhadap akurasi hasil sensor maka diperlukan kalibrasi dan jika terjadi delay antara tampilan data di aplikasi maka yang terjadi pada koneksi <i>wireless</i> terputus</p> <p>Data yang dikumpulkan dapat dianalisis menggunakan analitik data besar, dan tindakan pencegahan dapat dilakukan sebelum kisaran ambang batas dilampaui oleh parameter kualitas air. Dimungkinkan untuk mengotomatisasi sistem tambak menggunakan internet hal-hal yang mengurangi konsumsi energi dan biaya tenaga kerja</p>

## 2.2 Landasan Teori

### 2.2.1 Ikan nila

Tilapia (*Oreochromis niloticus*) merupakan salah satu spesies ikan air tawar yang dapat dipelihara di air tawar, air payau, laut, kolam biasa, kolam air tenang (KAT), Keramba Jaring Apung (KJA), Kolam Air Hujan (KAH), keramba dan kolam. Tilapia berasal dari Sungai Nil dan danau di sekitarnya. Pada tahun 1969, Balai Penelitian Perikanan Air Tawar membawa ikan nila dari Taiwan ke Bogor untuk pertama kalinya. Satu tahun kemudian, Ikan nila itu mulai menyebar ke beberapa daerah. nama nila berdasarkan keputusan Dirjen Perikanan tahun 1969.

Berikut pada gambar 2.1 merupakan gambar ikan nila:



Gambar 2. 1 Ikan Nila

Tilapia merupakan ikan air tawar yang digemari oleh masyarakat Indonesia untuk konsumsi. Secara morfologi, ikan nila biasanya memiliki bentuk tubuh tinggi, ramping, sisik besar dan mata cukup besar, yang terlihat menonjol dan tepi putih. Ikan nila biasanya berwarna merah, hitam dan hasil campuran (hitam-merah). tangani Sedangkan menurut Khairuman dan Amri (2013), ciri lainnya adalah: gurat sisi (*linea lateralis*) kemudian putus di tengah melangkah lebih jauh, tetapi lebih rendah dari garis atas sirip dada Ikan nila memiliki lima sirip yaitu sirip punggung. Sirip dada, sirip perut (*pelvic fins*), ekor dan sirip ekor (sirip ekor). Sirip punggung memanjang dari ujung penutup insang bagian atas sirip ekor. Ia memiliki sepasang sirip dada kecil dan sirip perut.

Berikut ini adalah klasifikasi ikan nila menurut (Khairuman et al., 2013):

Filum	: Chordata
Subfilium	: Vertebrata
Kelas	: Pisces
Subkelas	: Acanthopterigii
Suku	: Cichlidae
Marga	: Oreochromis
Spesies	: Oreochromis sp.

### 2.2.2 Pengolahan kualitas air

Pengaruh pertumbuhan, pemeliharaan dan pembibitan budidaya ikan nila yaitu kualitas air. Kualitas air adalah sifat air dan konsentrasi makhluk hidup, zat energi atau komponen lain di dalam air. Kualitas air dinyatakan dengan beberapa parameter, yaitu parameter fisik (suhu, kekeruhan, TDS, dll), parameter kimia (pH, oksigen terlarut, BOD, kandungan logam, dll.) dan parameter biologis (keberadaan plankton, bakteri). Berdasarkan standar nasional Indonesia nomor 7550 2009 bahwa persyaratan kualitas air layak untuk budidaya tilapia sebagai berikut:

Tabel 2. 2 Parameter Kualitas Air

No	Parameter	Nilai
1	Suhu	25-32 °C
2	pH	6.5-8.5
3	Oksigen terlarut	≤ 3 mg/l
4	Amoniak (NH <sub>3</sub> )	< 0.02 mg/l
5	Kecerahan	30-40 cm

Melalui tabel 2. 2, dapat diketahui bahwa parameter dan nilainya adalah sebagai berikut:

#### a. Suhu

Suhu adalah faktor pengontrol dan berperan dalam sistem daur ulang. Berdasarkan SNI nomor 7550:2009 bahwa pengukuran suhu dapat dilakukan pada permukaan

air dan dasar tangki untuk memungkinkan pertumbuhan dan perkembangan nila sesuai dengan suhu sekitar 25-32°C. Jika nilai ikan nila tumbuh pada suhu 14 °C atau suhu tinggi 38 °C pada suhu 6 °C atau 42 °C nila mati.

Menurut (Siegers et al., 2019) bahwa pada suhu di bawah 25°C pergerakan dan nafsu makan ikan dimulai mengurangi Ikan mati membeku di bawah 12 °C. Di atas 35 °C ikan akan menderita stres dan kesulitan bernapas karena konsumsi oksigen yang meningkat.

b. Derajat kesamaan pH

Keasaman air terbagi menjadi tiga yaitu pH rendah (keasaman), pH netral dan pH tinggi (basa). Air menjadi asam ketika  $pH < 7$ . Biasanya, perubahan pH air disebabkan oleh residu pakan dan pupuk ikan. Berdasarkan SNI nomor 7550:2009 adalah 6,5 - 8.5. Ketika pH air  $< 6$ , ikan mati, mudah sakit dan pertumbuhan melambat.

c. Oksigen terlarut

Oksigen terlarut atau dissolved oxygen adalah jumlah oksigen larut dalam air kolam. Hewan membutuhkan oksigen untuk hidup seperti ikan dan organisme air lainnya untuk bernapas. Berhubungan dengan SNI nomor 7550:2009 bahwa oksigen terlarut dalam budidaya ikan nila adalah  $\leq 3$  mg/l. Efek kekurangan oksigen bisa diubah mengakibatkan ikan mati, pincang dan mobilitas berkurang. Jika Saat kondisi tersebut terjadi, ikan mencari oksigen atau air di permukaan air.

d. Amoniak

Senyawa beracun pada ikan nila adalah amonia. Hal ini dapat terjadi karena proses metabolisme ikan dan proses penguraian bahan organik yang dibawa oleh bakteri Berdasarkan SNI nomor 7550:2009, amoniak yang sesuai dengan pembudidayaan ikan nila yaitu  $< 0.02$  mg/l.

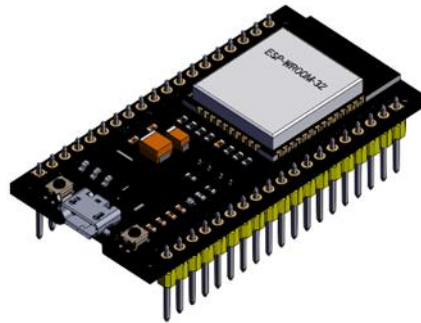
e. Kekeruhan

Penyebab kekeruhan adalah beberapa partikel seperti lumpur, material, puing-puing atau plankton. Kekeruhan yang disebabkan oleh plankton dinilai baik karena mengandung banyak nutrisi alami. Kekeruhan yang direkomendasikan oleh (Oktafiadi, 2016) yaitu tidak lebih dari 50 NTU sehingga pertumbuhan ikan berkembang dengan baik.



### 2.2.3 Mikrokontroler ESP-32

Gambar 2.2 merupakan mikrokontroler yang digunakan dalam penelitian ini:



Gambar 2. 2 Mikrokontroler ESP-32

ESP 32 adalah mikrokontroler yang dikenalkan oleh Espressif System merupakan penerus dari mikrokontroler ESP8266. Menurut (Imran & Rasul, 2020) pada mikrokontroler ini sudah tersedia modul WiFi dalam *chip* sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things*. Pin dapat dijadikan *input* atau *output* untuk menyalakan LCD, lampu, bahkan untuk menggerakkan motor DC.

#### 4.2.3.1 Spesifikasi ESP-32

Berikut ini adalah spesifikasi untuk memperjelas pemahaman mengenai ESP32 pada tabel dibawah ini:

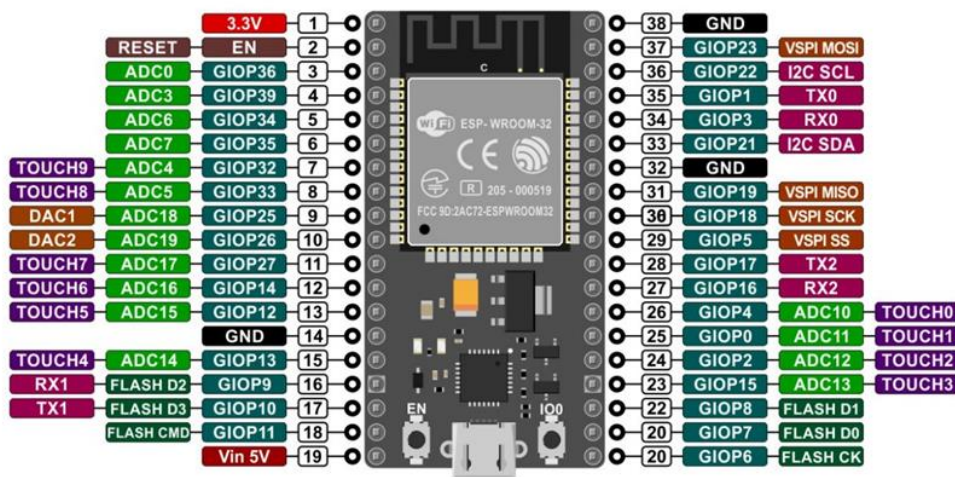
Tabel 2. 3 Spesifikasi ESP-32

Atribut	Detail
CPU	Tensilica Xtensa LX6 32bit Dual-Core di 160/240MHz
SRAM	520 KB
FLASH	2MB (max 64MB)
Tegangan	2.2V sampai 3.6V
Arus Kerja	Rata-rata 80mA
Dapat diprogram	Ya (C,C++,Python,Lua,dll)
Open Source	Ya
<b>Konektivitas</b>	
Wi-Fi	802.11 b/g/n

Atribut	Detail
Bluetooth	4.2BR/EDR + BLE
UART	3
<b>I/O</b>	
GPIO	32
SPI	4
I2C	2
PWM	8
ADC	18 (12-bit)
DAC	2 (8-bit)

#### 4.2.3.2 Konfigurasi ESP-32

Gambar 2.3 merupakan penjelasan pin pada ESP-32:



Gambar 2. 3 Pin ESP-32

Sumber: <https://www.ardutech.com/mengenal-esp32-development-kit-untuk-iot-internet-of-things/>

Pada *board* ESP32 DevKit terdapat 25 pin GPIO (*General Purpose Input Output*) dengan masing – masing pin mempunyai karakteristik sendiri – sendiri.

Pin hanya sebagai *INPUT*:

- GPIO 34
- GPIO 35
- GPIO 36
- GPIO 39

Pin dengan internal *pull up*, dapat diseting melalui program:

- GPIO14
- GPIO16
- GPIO17
- GPIO18
- GPIO19
- GPIO21
- GPIO22
- GPIO23

Pin tanpa internal *pull up* (dapat ditambahkan *pull up eksternal* sendiri):

- GPIO13
- GPIO25
- GPIO26
- GPIO27
- GPIO32
- GPIO33

#### 2.2.4 Sensor Analog pH KB2004

Sensor analog pH KB2004 adalah sensor yang digunakan untuk mendeteksi kadar pH keasaman dan kebasaan cairan. Nilai ukur pH air adalah antara 0 hingga 14. Nilai kurang dari 7 bersifat asam, jika nilai sama dengan 7 bersifat netral, dan nilai lebih besar dari 7 bersifat basa. Sensor pH air adalah jenis sensor kimia karena sensor itu berubah berdasarkan reaksi kimia dalam besaran listrik.

Berikut merupakan spesifikasi Sensor Analog pH KB2004 pada tabel 2.4:

Tabel 2. 4 Spesifikasi Sensor Analog pH KB2004

<i>Supply Voltage</i>	3,3-5,5V
-----------------------	----------

<i>Output Voltage</i>	0-3.0V
<i>Probe Connector</i>	BNC
<i>Signal Connector</i>	PH2.0-3P
<i>Measurement Accuracy</i>	$\pm 0.1 @ 25^{\circ}\text{C}$
<i>Dimension</i>	42mm*32mm
<i>Detection Range</i>	0-14
<i>Temperature Range</i>	5-60 $^{\circ}\text{C}$
<i>Response Time</i>	<2min

Sensor pH air memiliki 2 jenis elemen elektroda yaitu elektroda kaca dan elektroda referensi. Prinsip kerja sensor pH air yaitu, elektroda kaca mengukur jumlah ion dalam larutan sehingga elektroda referensi mengubah nilai jumlah ion yang dibaca oleh elektroda referensi adalah nilai tegangan. Sehingga semakin banyak elektron yang terdeteksi elektroda kaca dan elektroda referensi maka larutannya bersifat asam dan jika semakin sedikit elektron yang terdeteksi, semakin basa larutan tersebut.

Berikut merupakan gambar sensor pH air pada gambar 2.4:



Gambar 2. 4 Sensor pH air

#### 2.2.5 Sensor suhu DS18B20

Sensor suhu DS18B20 merupakan sensor yang dapat mendeteksi suhu dengan menggunakan 1 kabel (membutuhkan 1 pin untuk jalur data) yang terhubung ke mikrokontroler. Sensor ini tahan air dan bisa mengukur suhu antara  $-55^{\circ}\text{C}$  dan  $+125^{\circ}\text{C}$ . Spesifikasi sensor suhu DS18B20 sebagai berikut:

Tabel 2. 5 Spesifikasi Sensor Suhu DS18B20

Power Supply	3V – 5,5 V
Konsumsi arus	1 mA
Range suhu	-55 sampai 125°C
Akurasi	±0,5%
Resolusi	9 – 12 bit
Waktu konversi	< 750 ms

Sensor suhu DS18B20 juga menyediakan pembacaan 9-12 bit dan dapat dikonfigurasi. Hasil baca dikirim ke alamat atau tujuan DS18B20 lebih dari 1 - koneksi kabel. Power dibutuhkan untuk membaca, menulis dan melakukan konversi suhu dapat diturunkan dari jalur data itu sendiri tanpa sumber daya eksternal. Bentuk dari sensor suhu DS18B20 ditunjukkan pada gambar sebagai berikut:



Gambar 2. 5 Sensor Suhu DS18B20

### 2.2.6 Sensor oksigen

Sensor DO atau *dissolved oxygen* merupakan perangkat sensor terlarut yang kompatibel dengan papan Arduino. Sensor ini digunakan untuk mengukur oksigen terlarut dalam air dan untuk mengetahui kualitas air. Sensor ini dapat diterapkan dalam banyak aplikasi kualitas air, seperti akuakultur, pemantauan lingkungan, dan kualitas air.

Berikut merupakan gambar sensor DO pada gambar 2.6:



Gambar 2. 6 Sensor *Dissolved Oxygen*

Oksigen terlarut adalah salah satu parameter penting untuk mencerminkan kualitas air. Oksigen terlarut yang rendah dalam air akan menyebabkan organisme akuatik kesulitan bernafas, hal itu dapat mengancam kehidupan organisme tersebut. Sensor ini sangat cocok untuk proyek dalam mendeteksi konsentrasi oksigen terlarut.

Berikut merupakan spesifikasi sensor DO pada tabel 2.6:

Tabel 2. 6 Spesifikasi Sensor DO

<i>Type</i>	Galvanic Probe
<i>Detection Range</i>	0~20 mg/L
<i>Temperature Range</i>	0~40 °C
<i>Response Time</i>	Up to 98% full response, within 90 seconds (25°C)
<i>Pressure Range</i>	0~50 PSI
<i>Cable Length</i>	2 meters
<i>Probe Connector</i>	BNC

### 2.2.7 Modul relay

Relay adalah sakelar elektronik yang menggunakan arus listrik. Prinsip relai bekerja dengan induksi elektromagnetik menggunakan tuas pemindah dengan kawat (solenoid) yang dililitkan pada batang besi kemudian diberi energi dan tuas ditarik oleh gaya magnet yang ada pada elektromagnet sehingga kontak saklar akan menutup. Ketika elektromagnet kehilangan energi maka tuas kembali keposisi semula dan kontak saklar terbuka kembali.

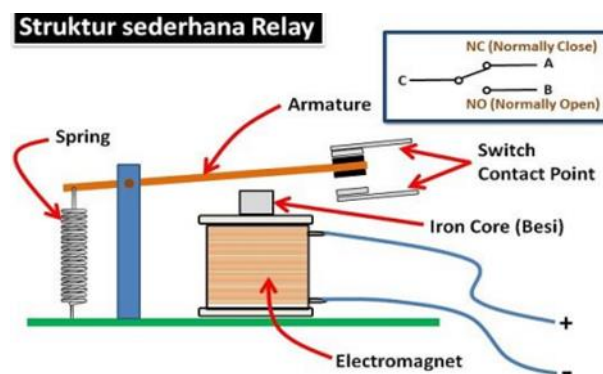
Berikut gambar modul relay pada gambar 2.7:



Gambar 2. 7 Modul Relay

Dua bagian utama adalah elektromagnet (koil) dan mekanik (set kontak). Relai ini mengontrol magnet dengan arus atau tegangan kecil, tetapi kontak relai dapat beralih atau mengontrol listrik atau tegangan tinggi. Terdapat kontak relay yaitu *Normally Open* (NO) dan *Normally Close* (NC). *Normal Open* (NO) adalah kontak yang terhubung/tertutup saat kumparan solenoid aktif sedangkan *Normally Closed* (NC) adalah kontak yang terbuka ketika solenoid aktif. Dalam sistem ini, relai berfungsi sebagai fungsi utama yang dapat mengaktifkan output adalah pompa DC dan katup solenoida DC. Relay biasanya digunakan untuk memindahkan atau mengaktifkan perangkat elektronik dengan arus tinggi atau tegangan tinggi di seluruh kontak relai. Magnet tersebut diaktifkan dengan tegangan DC sehingga dapat mengarahkan kontak relay dengan menggunakan tegangan AC. Dengan arus yang relatif kecil untuk mengendalikan peralatan dengan arus besar.

Berikut merupakan gambar struktur komponen relay pada gambar 2.8:



Gambar 2. 8 Struktur Komponen Relay

Penjelasan bagian – bagian dari relay sebagai berikut :

- *Electromagnet* / koil electromagnet yang berfungsi sebagai pembentuk medan magnet Ketika mendapatkan tegangan listrik yang sesuai dengan tegangan kerja relay.
- *Iron core* / inti besi yaitu bagian yang berperan menjadi perantara medan magnet untuk menarik bagian armature sehingga dapat mengubah posisi *switch* kontak *point*.
- *Armature* yaitu lempengan logam yang digunakan sebagai tuas kontak yang bergerak mengubah posisi kontak berdasarkan sifat magnet dari komponen inti besi yang mempengaruhinya.
- *Switch contact point* yaitu bagian relay yang berfungsi sebagai *output* relay. *Switch* kontak terdapat 2 yaitu *Normally Open* (NO) dan *Normally Close* (NC).
- *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di



posisi tertutup

- *Normally Open (NO)* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi terbuka
- *Spring / per* yaitu bagian yang digunakan untuk mengembalikan posisi *switch contact point relay* saat koil electromagnet dialiri arus/tegangan.

### 2.2.8 Bahasa pemrograman C pada Arduino

Bahasa pemrograman C dapat diterapkan untuk Windows, Unix, Linux atau sistem operasi lain tanpa mengalami perubahan pada *source code*, dimana ketika terjadi perubahan biasanya sangat minor atau sedikit. Karena Arduino menggunakan bahasa C lintas platform, perangkat lunak Arduino bekerja pada sistem operasi umum seperti Windows, Linux dan MacOS.

Berikut penjelasan dasar-dasar pemrograman mikrokontroler Arduino menggunakan bahasa C, yang meliputi: Struktur, sintaks, variabel, operator matematik, operator relasional, struktur bersyarat, inisialisasi I/O digital dan analog sebagai berikut:

#### a. Struktur

Dalam setiap program arduino harus memiliki dua buah fungsi yang harus ada yaitu *Void Setup* dan *Void Loop*.

- *Void Setup () {}*

Kode yang berada pada kurung kurawal *void setup* akan dijalankan hanya satu kali ketika program arduino dijalankan untuk pertama kalinya.

- *Void Loop () {}*

*Void loop* dijalankan ketika void setup sudah dijalankan. Fungsi ini akan terus-menerus dijalankan sampai dengan daya dimatikan.

#### b. Syntax

- Komentar

Komentar digunakan untuk menjelaskan setiap baris instruksi yang diberikan agar komentar sangat berguna bagi pengembang karena mempermudah untuk menemukan temuan jalur data yang diperlukan. Simbol `"/"` (dua garis miring) digunakan untuk menulis komentar satu baris sementara jika ingin menulis komentar multibaris, gunakan simbol di setiap kasus dimulai dengan `"/*` (garis miring dan tanda bintang) dan diakhiri dengan simbol `*/"` (bintang dan garis).

- *Curly bracket*

Simbol “{” dan “}” kurung kurawal (*curly bracket*) digunakan untuk menentukan awal dan akhir sebuah blok kode program. Penggunaan simbol ini digunakan pada fungsi dan *loop*.

- *Semicolon*

Simbol ";" disebut juga titik koma (titik koma). Tujuan penulisan titik koma adalah sehingga tidak terjadi kesalahan pemrograman atau kesalahan prosedur selama kompilasi program.

c. Variabel

Variabel digunakan untuk menyimpan atau mentransfer nomor karakter dalam program. Variabel harus dideklarasikan terlebih dahulu dan di *sketch* (program arduino) sesuai dengan keperluan. Nama variabel dimulai dengan angka dan kata kunci. Beberapa contoh tipe data bahasa pemrograman yang digunakan di Arduino adalah sebagai berikut:

- *int* (integer) : Tidak mempunyai angka *decimal* dan memiliki rentang nilai antara – 32.748 sampai dengan +32.768.
- *long* : Tipe data ini digunakan ketika terdapat variabel yang tersimpan di *int* tidak mencukupi lagi. Tipe data ini memiliki rentang nilai antara 2.147.483.648 sampai dengan +2.147.483.648 .
- *Booelan* : Tipe data sederhana yang digunakan untuk menyimpan nilai *True* dan *False*.
- *Float* : Tipe data ini digunakan untuk bilangan desimal (*floating point*). Memiliki rentang nilai antara -3,4028235E+38 sampai dengan 3,4028235E+38.
- *char* (karakter) : Tipe data karakter yang menggunakan kode ASCII (misalnya ‘A’ = 65) dan memakai 1 byte dari RAM.
- *String* : tipe data ini digunakan untuk menyimpan informasi teks, dengan karakter ASCII dan menggunakan string untuk mengirim pesan *via serial port*.

d. Operator Matematik

Operator matematik digunakan untuk memanipulasi angka. Berikut merupakan jenis – jenis operator matematik :

- = : Untuk sama dengan.

- % : Digunakan sebagai modulo, untuk memberikan nilai sisa dari hasil pembagian suatu angka dengan angka yang lain.
- + : Untuk penjumlahan
- - : Untuk pengurangan
- / : Untuk pembagian

#### e. Operator Perbandingan

Operator perbandingan digunakan untuk operasi perbandingan logis antara dua variabel yang berbeda. Operator perbandingan sebagai berikut:

- == : sama dengan (*equal to*)
- != : tidak sama dengan (*not equal to*)
- < : lebih kecil dari (*less than*)
- > : lebih besar dari (*greater than*)

#### f. Bitwise Operator

Operator ini digunakan untuk melakukan perhitungan variabel *bit level*. Berikut ini adalah jenis-jenis operator *bitwise*:

- & : *bitwise and*
- | : *bitwise or*
- ^ : *bitwise xor*
- ~ : *bitwise not*
- << : *bitwise left*
- >> : *bitwise right*

#### g. Operator Boolean

Operator boolean digunakan dalam *statement* percabangan *IF*. Berikut ini jenis – jenis operator *boolean* :

- && : *AND*
- || : *OR*
- ! : *NOT*

### 2.2.9 Arduino IDE

Arduino IDE (*Intergrated Development Environment*) adalah program perangkat lunak yang memiliki berbagai fungsi peran seperti editor program atau menulis sketsa (program Arduino), menerjemahkan ke dalam kode biner dan memuat ke dalam mikrokontroler di papan Arduino. Sketsa di Arduino IDE juga mendukung fungsi nomor baris, *syntax highlighting*, yaitu memeriksa sintaks dari kode proyek. *Sketch* adalah program yang dibuat menggunakan Arduino IDE dan ditulis atau disimpan dalam sebuah ekstensi atau format.ino.

Cara penggunaannya harus memiliki dua fungsi untuk menjalankannya yaitu:

- *Void Setup ( ) { }* : fungsi ini akan dijalankan sekali pada awal program yang dapat menginisialisasi pengaturan.
- *Void Loop ( ) { }* : fungsi ini dijalankan secara berulang-ulang sampai mikrokontroler di non-aktifkan.

Tampilan *toolbar* pada Arduino IDE yaitu *File, Edit, Sketch, Tools, dan Help*. Arduino IDE terdiri dari:

1. *Verify*: Digunakan untuk melihat *sketch* (program Arduino) pada editor program.
2. *Upload*: Digunakan untuk mengirimkan ke dalam memori mikrokontroler yang ada di papan Arduino
3. *New*: Digunakan untuk membuat *sketch* (program Arduino) yang baru.
4. *Open*: Digunakan untuk membuka daftar *sketch* pada *sketchbook* arduino.
5. *Save*: Digunakan untuk kode *sketch* pada *sketchbook* Arduino.
6. *Serial Monitor*: Digunakan untuk menampilkan informasi yang dikirim melalui komunikasi serial.
7. *Editor Program*: Digunakan untuk menulis atau mengedit *sketch* (program Arduino).

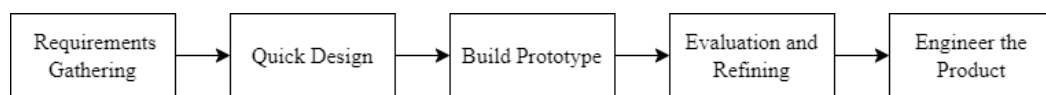
### 2.2.10 Prototyping

Menurut (Ogedebe & Jacob, 2012) menyampaikan bahwa *prototyping* merupakan metode pengembangan perangkat lunak, yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. *Prototype* dibangun untuk mendefinisikan kebutuhan awal dan akan dihilangkan atau ditambahkan pada bagiannya sehingga sesuai dengan perencanaan dan analisis yang dilakukan oleh pengembang (Purnomo, 2017).

Kelebihan metode prototyping menurut (Supandi et al., 2019) :

1. Dapat mengakomodir (menyediakan sesuatu untuk memenuhi kebutuhan) perangkat lunak yang spesifikasi kebutuhannya belum rinci.
2. Dapat diimplementasikan di dalam konteks setiap model perangkat lunak.
3. Lebih menghemat waktu dalam pengembangannya.
4. Pelanggan berperan aktif dalam pengembangan sistem.

Menurut (Agustia & Kiki H, 2018) metode *prototyping* memiliki proses tahapan yang berurutan sehingga dapat lebih mudah dalam mengidentifikasi kecacatan dalam pengembangan serta pengujian performa yang lebih baik. Berikut merupakan urutan dalam proses *prototyping*:



Gambar 2. 9 Tahapan *Prototyping*

1. *Requirements Gathering*

Pada tahapan ini dilakukan analisis kebutuhan sistem yang akan dibangun dengan melakukan proses pengumpulan data menggunakan teknik diskusi antara *developer* dengan pengguna untuk mendapatkan keinginan pengguna.

2. *Quick Design*

Tahapan kedua dalam *prototype* adalah membuat desain awal serta memberikan gambaran umum tentang sistem yang akan dikembangkan.

3. *Build Prototype*

Langkah ketiga adalah membangun *prototype* sebenarnya yang akan digunakan sebagai acuan dalam pembuatan alat.

4. *Evaluation and Refining*

Setelah pembuatan *prototype* selanjutnya dilakukan presentasi kepada pengguna untuk dilakukan evaluasi. Pengguna akan memberikan saran mengenai hasil *prototype* yang telah dibuat.

5. *Engineer the Product*

Pada tahap ini, produk segera dibuat dan diserahkan kepada pengguna sesuai dengan hasil evaluasi yang telah dilakukan.

## BAB III

### METODE PENELITIAN

#### 3.1 Objek Penelitian

Objek pada penelitian ini adalah perancangan sistem automasi pada budidaya ikan nila Minaverse Company yang terletak di sambilegi lor, Maguwoharjo, Depok, Sleman, DI Yogyakarta. Dalam penelitian ini memiliki fokus untuk membuat sistem automasi berbasis IoT (*Internet of Things*) untuk mengukur pH air dan kadar oksigen sehingga dapat memaksimalkan hasil panen ikan.

#### 3.2 Subjek Penelitian

Subjek pada penelitian ini adalah pemilik kolam ikan nila Minaverse Company. Subjek penelitian ini memiliki peran dalam proses identifikasi keberhasilan sistem yang telah di rancang. Untuk melihat keberhasilan tersebut maka lingkungan hidup ikan nila atau kolam harus memiliki pH dan oksigen yang baik. Jika kedua aspek tersebut berjalan dengan baik maka hasil yang didapatkan adalah hasil panen ikan akan meningkat.

#### 3.3 Jenis Data Penelitian

Pada penelitian ini digunakan beberapa jenis sumber data. Jenis sumber data yang digunakan yaitu data primer dan data sekunder. Berikut merupakan penjelasan lebih detail terkait kedua jenis sumber data yang digunakan

##### 1. Data Primer

Data primer merupakan sumber data yang memberikan data kepada pengumpul data. Data tersebut dikumpulkan langsung oleh peneliti yang didapatkan langsung dari sumber pertama atau lokasi objek penelitian (Sugiyono, 2018). Berikut merupakan metode yang digunakan dalam pengambilan data primer:

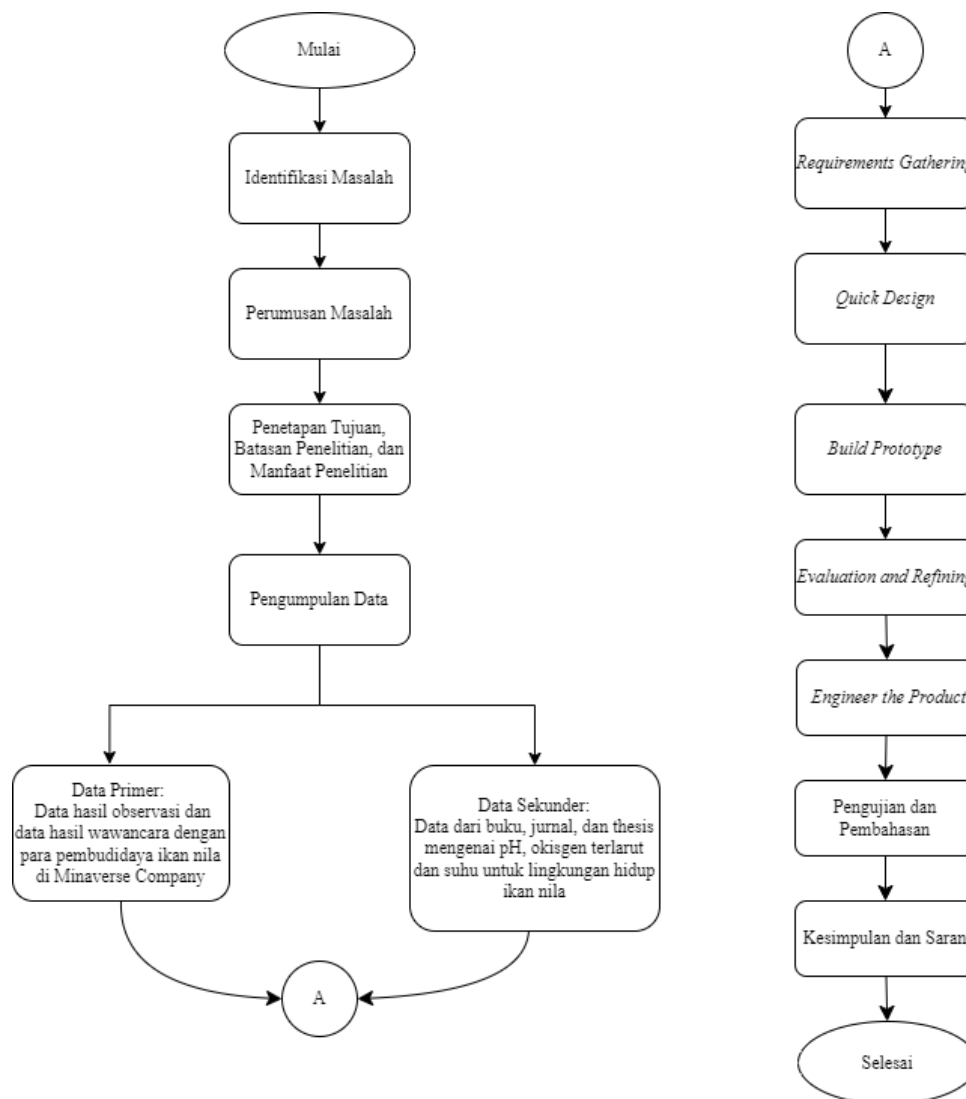
- a. Observasi, adalah pengamatan yang dilakukan secara langsung untuk memperoleh data-data yang dibutuhkan seperti data produksi yang digunakan dalam membangun simulasi.
- b. Wawancara, adalah kegiatan tanya jawab guna mendapatkan informasi yang dibutuhkan dalam penelitian. Kegiatan ini dilakukan dengan pemilik budidaya ikan nila Minaverse Company.

## 2. Data Sekunder

Data sekunder merupakan data pendukung yang didapatkan secara tidak langsung. Berikut merupakan metode yang digunakan pada penelitian ini dalam pengumpulan data sekunder. Data sekunder yang digunakan pada penelitian ini adalah data yang berasal dari beberapa literatur seperti jurnal dan buku yang memiliki korelasi dengan metode ataupun topik pada penelitian ini.

### 3.4 Alur Penelitian

Pada penelitian ini terdapat beberapa tahap yang dilalui, sebagai berikut :



Gambar 3. 1 Alur Penelitian

Alur di atas merupakan gambaran tahapan yang dilakukan dalam penelitian ini, berikut adalah penjelasan untuk setiap tahapan tersebut :

1. Mulai

Peneliti mempersiapkan hal-hal yang dibutuhkan untuk melakukan penelitian

2. Identifikasi Masalah

Peneliti melakukan identifikasi dari masalah yang ada pada kolam ikan terkait pH air, oksigen, dan suhu. Permasalahan ini akan menjadi rumusan masalah dari penelitian kali ini

3. Perumusan Masalah

Setelah melakukan identifikasi masalah, maka dilakukan perumusan masalah. Pada tahap ini masalah-masalah yang telah ditemukan sebelumnya akan dikerucutkan agar lebih spesifik dan rumusan masalah tersebut akan diselesaikan pada penelitian ini.

4. Penentuan Tujuan, Batasan Masalah, dan Manfaat Penelitian

Menentukan tujuan yang ingin dicapai pada penelitian ini, selanjutnya menentukan batasan masalah yang ditujukan untuk membatasi cakupan dari penelitian agar tetap berfokus pada tujuan yang hendak dicapai, dan yang terakhir menentukan manfaat penelitian bagi peneliti maupun tempat yang diteliti.

5. Pengumpulan Data

Pengumpulan data dibagi menjadi dua metode, yaitu:

a. Data primer

Berupa hasil observasi yang dilakukan secara langsung pada pembudidaya ikan nila bertempat di Minaverse Company yaitu kegiatan pemberian cairan pH, monitoring suhu dan kincir oksigen. Selain observasi peneliti juga melakukan wawancara langsung dengan pemilik Minaverse Company terkait pengelolaan budidaya ikan nila terkait kesulitan yang dihadapi pada saat melakukan budidaya dan faktor untuk meningkatkan hasil panen ikan.

b. Data sekunder

Data sekunder yaitu data dari berbagai jurnal, buku dan thesis yang memiliki kaitan dengan penelitian ini terkait pengelolaan kualitas air untuk budidaya ikan nila, baik internasional maupun nasional dengan tujuan memperkaya dasaran ilmu dalam mendukung jalannya penelitian.

6. *Requirements Gathering*



Berdasarkan hasil wawancara dan observasi, maka dapat dilakukan analisis kebutuhan terkait pengembangan sistem kedepannya. Pada tahap ini juga ditentukan *output* yang ditargetkan secara garis besar yakni berfokus pada pengelolaan, *monitoring*, dan *controlling*.

#### 7. *Quick Design*

Perancangan awal meliputi *hardware dan software* yang digunakan berdasarkan analisis kebutuhan. *Hardware* yang digunakan meliputi mikrokontroler ESP-32, sensor suhu DS18B20, sensor pH air KB2004, dan sensor *dissolved oxygen*. *Software* yang digunakan adalah Arduino IDE dan Blynk *Cloud*.

#### 8. *Build Prototype*

Membuat *prototype* menggunakan mikrokontroler yang telah diberi *input, proses*, dan *output* setelah itu dilakukan proses penyetulan pada Blynk *Cloud* yang menjadi tempat untuk melakukan *controlling* dan *monitoring*.

#### 9. *Evaluation and Refining*

Pada tahap ini peneliti membahas terkait hasil *prototype* dan dilakukan *testing* kepada calon pengguna yaitu peternakan ikan nila Maguwoharjo, setelah itu didapatkan *feedback* terkait produk yang dirancang.

#### 10. *Engineer the Product*

Tahap terakhir pada penelitian ini adalah dimana peneliti akan membuat produk *final* berdasarkan saran dari calon pengguna dan dapat diartikan sebagai produk yang telah selesai dibuat.

#### 11. Pengujian dan Pembahasan

Pada tahap ini peneliti membahas terkait *output* yang dihasilkan dalam penelitian dan melakukan perbandingan antara kolam yang telah dilakukan proses automasi untuk mengetahui alat akan diletakkan di posisi yang membutuhkan.

#### 12. Kesimpulan dan Saran

Tahap terakhir pada penelitian ini adalah dimana peneliti akan menarik kesimpulan dan menentukan saran terbaik yang akan diberikan sesuai dengan analisis yang telah dilakukan.

## BAB IV

### PEMBANGUNAN SISTEM

#### 4.1 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan di Minaverse Company, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Data yang diambil berupa hasil dari alat automasi yang telah dirangkai dan diaplikasikan di dalam kolam. Hasil dari alat automasi yaitu pengukuran pH air dan suhu air untuk menjaga kualitas air dan meningkatkan hasil panen. Selain itu penggunaan rangkaian automasi oksigen untuk membuat kincir oksigen menyala pada saat oksigen didalam kolam menurun, sehingga dapat meminimalisir pengeluaran biaya listrik.

#### 4.2 Profil Minaverse Company

Minaverse Company Kabupaten Sleman adalah perusahaan yang bergerak dalam bidang *agritechnology* salah satunya dalam hal perikanan. Minaverse Company berdiri pada tahun 2019 dan bergerak dalam pembudidayaan ikan nila, yang diolah dan dikelola oleh Bapak Dyogi M.T sebagai pendiri dari Minaverse Company didampingi oleh Bapak F. Adhi sebagai wakil pendiri dan Bapak Adhi Khusna sebagai pengelola keuangan, selain itu para pekerja dilakukan oleh warga setempat atau masyarakat yang berada di sambilegi lor, Maguwoharjo, Depok, Sleman, DI Yogyakarta.

Berikut merupakan gambar logo Minaverse Company pada gambar 4.1:



Gambar 4. 1 Minaverse Company

Minaverse hadir sebagai pendamping dan pembudidaya ikan dari Yogyakarta dengan sistem “*zero waste*” yaitu mengintegrasikan antara pertanian, perikanan dan peternakan sehingga petani bisa memiliki pendapatan harian, mingguan dan bulanan yang berkelanjutan. Menciptakan lapangan kerja dan melibatkan semua lapisan masyarakat baik orang tua, laki-laki atau perempuan sesuai dengan prinsip tahapan pembangunan berkelanjutan “*no one left behind*”.

#### A. Visi

Mewujudkan sektor perikanan yang mandiri, maju dan terintegrasi antara pertanian, peternakan, dan perikanan berbasis pada kearifan lokal yang berkelanjutan.

#### B. Misi

Menciptakan lapangan kerja bagi semua kalangan dan bekerja sama dengan berbagai pihak untuk menciptakan ekosistem usaha perikanan.

### 4.3 Requirements Gathering

#### 4.3.1 Kebutuhan pengguna

Pada tahap ini dilakukan analisis terhadap kebutuhan pengguna yaitu pembudidaya ikan nila. Dalam proses pengelolaannya dapat mengetahui apa saja yang perlu dikembangkan peneliti menggunakan metode wawancara kepada pihak pengguna. Selain itu, dilakukan juga observasi terkait kebutuhan meliputi sistem ataupun mekanisme pada struktur pembudidaya ikan nila secara langsung sebagai acuan dalam proses pengembangan sistem automasi.

Berikut merupakan tabel kebutuhan pengguna pada tabel 4.1:

Tabel 4. 1 Kebutuhan Pengguna

No.	Kegiatan	Kondisi	Solusi
1.	Pengelolaan	Kegiatan pengelolaan budidaya ikan nila masih secara tradisional (manual). Tidak ada data yang digunakan hanya berdasarkan pengalaman dan intuisi para pembudidaya ikan.	Memudahkan proses kegiatan budidaya ikan nila tanpa harus menggunakan banyak tenaga manusia. Kegiatan pengelolaan yang perlu dilakukan automasi yaitu: pemberian probiotik untuk mengatur pH air secara stabil, mengatur oksigen air,

No.	Kegiatan	Kondisi	Solusi
2.	<i>Monitoring and controlling</i>	Kegiatan pemantauan dan kontrol kolam ikan masih dilakukan secara tradisional dengan cara para pembudidaya rutin melihat kolam secara langsung	dan suhu air. Memudahkan proses pemantauan dan kontrol sistem kolam ikan nila sehingga peternak tidak perlu rutin ke kolam untuk mengecek kondisi kolam dan pengelolaan melalui sistem <i>cloud</i> dengan bantuan IoT.

#### 4.4 Quick Design

##### 4.4.1 Peralatan penelitian


Dalam proses pembangunan *prototype* pada penelitian ini dibutuhkan peralatan yang mendukung perancangan sistem. Peneliti menggunakan dua macam alat yang digunakan, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Berikut merupakan keterangan dari masing-masing perangkat yang digunakan:

##### 1. Perangkat keras (*hardware*)

Perangkat keras atau *hardware* merupakan komponen yang dapat dipegang secara fisik dan juga dapat dilihat menggunakan mata secara langsung. Dalam penggunaannya biasanya dibantu atau dioperasikan menggunakan perangkat lunak (*software*).




Berikut merupakan tabel *hardware* pada tabel 4.2:

Tabel 4. 2 *Hardware*



No.	Nama dan Gambar	Keterangan
1.	Mikrokontroler ESP-32 	ESP-32 adalah salah satu jenis mikrokontroler yang telah dilengkapi dengan modul WiFi dan ditambah dengan BLE ( <i>Bluetooth Low Energy</i> ) dalam chip sehingga

No.	Nama dan Gambar	Keterangan
	Gambar 4. 2 ESP-32  (sumber: <a href="https://raharja.ac.id/2021/11/16/mikrokontroler-esp32-3/">https://raharja.ac.id/2021/11/16/mikrokontroler-esp32-3/</a> )	sangat mendukung dan dapat menjadi pilihan bagus untuk membuat sistem aplikasi <i>Internet of Things</i> .
2.	Laptop    Gambar 4. 3 Laptop (sumber: <a href="https://laptopmedia.com/series/asus-rog-gl552jx/">https://laptopmedia.com/series/asus-rog-gl552jx/</a> )	Perangkat laptop merupakan alat yang digunakan dalam melakukan pengecekan, pengkodean ( <i>coding</i> ) serta melakukan <i>monitoring</i> dan <i>controlling</i> sistem yang dibuat.
3.	USB kabel    Gambar 4. 4 USB Kabel (sumber: <a href="https://www.tokopedia.com/find/kabel-usb-esia">https://www.tokopedia.com/find/kabel-usb-esia</a> )	Kabel USB berguna sebagai alat penghubung laptop dengan mikrokontroler ESP-32, kabel USB juga berfungsi sebagai alat bertukar informasi dari kedua perangkat untuk mengirim kode ke dalam mikrokontroler, selain itu kabel USB juga digunakan sebagai sumber daya untuk mikrokontroler.
4.	Kabel Jumper	Kabel jumper berguna sebagai alat penghubung antar



No.	Nama dan Gambar	Keterangan
	 <p data-bbox="547 645 868 712">Gambar 4. 5 Kabel Jumper (sumber: <a href="https://www.arduinoindonesia.id/2022/11/pengertian-jenis-dan-cara-kerja-kabel-jumper-arduino.html">https://www.arduinoindonesia.id/2022/11/pengertian-jenis-dan-cara-kerja-kabel-jumper-arduino.html</a>)</p>	<p data-bbox="1082 282 1367 568">komponen pada arduino, kabel ini memungkinkan kondisi listrik dengan aman dikarenakan memiliki pin disetiap ujungnya.</p>
5.	<p data-bbox="360 831 501 864">Adaptor 9V</p>  <p data-bbox="419 1267 995 1335">Gambar 4. 6 Adaptor (sumber: <a href="https://ecadio.com/jual-adaptor-9v-uno">https://ecadio.com/jual-adaptor-9v-uno</a>)</p>	<p data-bbox="1082 831 1367 1518">Adaptor merupakan sumber daya yang digunakan dalam rangkaian perangkat, penggunaan adaptor dengan kapasitas sebesar 9V. Dapat digunakan sebagai <i>power supply</i> pada mikrokontroler ESP-32 sehingga hantaran listrik tidak kekurangan dengan banyaknya sensor yang digunakan.</p>
6.	<p data-bbox="360 1538 544 1572">LCD 12C 16x2</p>  <p data-bbox="403 1910 1011 1977">Gambar 4. 7 LCD I2C (sumber: <a href="https://www.waveshare.com/lcd1602-rgb-">https://www.waveshare.com/lcd1602-rgb-</a></p>	<p data-bbox="1082 1538 1367 1921">LCD atau <i>Liquid Crystal Display</i> merupakan media penampil <i>interface</i> pada mikrokontroler. LCD berfungsi sebagai alat pemantau jalannya program.</p>

No.	Nama dan Gambar module.htm)	Keterangan
7. <i>Analog Dissolved Oxygen</i>		<p>DO meter atau <i>Dissolved Oxygen</i> merupakan sensor yang digunakan untuk menunjukkan kadar oksigen terlarut dalam air yang selanjutnya dapat menjadi acuan dasar untuk menyimpulkan baik atau tidaknya kualitas air di dalam kolam.</p>
8. <i>Relay Module</i>		<p>Relay berguna untuk menggerakkan arus/tegangan yang besar. Selain itu modul relay berfungsi sebagai pemutus sumber tegangan apabila terjadi konsleting pada rangkaian eletronik.</p>
9. Sensor suhu DS18B20 <i>waterproof</i>		<p>Sensor suhu DS18B20 <i>waterproof</i> berguna sebagai pengukur temperatur suhu yang dapat dihubungkan dengan mikrokontroler. Sensor ini memiliki tingkat akurasi yang tinggi dan</p>

Gambar 4. 10 Sensor DS18B20

No.	Nama dan Gambar	Keterangan
	(sumber: <a href="https://indobot.co.id/blog/pemantau-suhu-akuarium-menggunakan-sensor-ds18b20-dengan-iot-telegram/">https://indobot.co.id/blog/pemantau-suhu-akuarium-menggunakan-sensor-ds18b20-dengan-iot-telegram/</a> )	dapat mengukur kestabilan suhu lebih baik dari sensor lainnya.
10.	12C/TWI LCD 1602 Module 	Modul 12C merupakan standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data.
	Gambar 4. 11 I2C Modul (sumber: <a href="https://saptaji.com/2016/06/27/bekerja-dengan-i2c-lcd-dan-arduino/">https://saptaji.com/2016/06/27/bekerja-dengan-i2c-lcd-dan-arduino/</a> )	Analog pH sensor merupakan alat pengukur pH baik keasaman maupun kebasahan yang dimiliki oleh cairan/larutan
11.	Analog pH Sensor 	
	Gambar 4. 12 pH Sensor (sumber: <a href="https://www.indiamart.com/proddetail/ph-sensor-kit-20479334630.html">https://www.indiamart.com/proddetail/ph-sensor-kit-20479334630.html</a> )	<i>Signal Isolator</i> berguna untuk menghilangkan <i>noise</i> dengan cara sinyal yang dikirim ke mikrokontroler akan
12.	<i>Arduino Analog Signal Isolator</i>	



No.	Nama dan Gambar	Keterangan
	 <p data-bbox="480 748 932 913">Gambar 4. 13 Analog <i>Signal Isolator</i> (sumber: <a href="https://botland.store/gravity-converters/9529-dfrobot-gravity-analog-signal-isolator-5904422313579.html">https://botland.store/gravity-converters/9529-dfrobot-gravity-analog-signal-isolator-5904422313579.html</a>)</p>	<p data-bbox="1082 284 1378 734">dipisahkan dari sinyal rangkaian sensor. <i>Signal Isolator</i> juga dapat mengatasi masalah <i>ground loop</i>, dapat memproteksi mikrokontroler jika mengalami <i>short circuit</i>.</p>
13.	<p data-bbox="363 920 475 965"><i>Air Pump</i></p>  <p data-bbox="480 1447 932 1570">Gambar 4. 14 <i>Air Pump</i> (sumber: <a href="https://www.instructables.com/How-to-Control-Vacuum-Pump-Air-Pump-Suction-for-Ro/">https://www.instructables.com/How-to-Control-Vacuum-Pump-Air-Pump-Suction-for-Ro/</a>)</p>	<p data-bbox="1082 920 1378 1435"><i>Air pump</i> berguna untuk mengalirkan air dari suatu tempat ke tempat lainnya. Dalam penelitian ini <i>air pump</i> berguna untuk memberikan probiotik secara automasi dengan perintah yang telah dibuat menggunakan arduino.</p>
14.	<p data-bbox="363 1576 475 1621"><i>Stop kontak</i></p>	<p data-bbox="1082 1576 1378 1975"><i>Stop kontak</i> berguna untuk menghubungkan satu aliran listrik dengan aliran lainnya. Untuk penelitian ini <i>stop kontak</i> berguna untuk menghubungkan kincir oksigen dengan</p>

No.	Nama dan Gambar	Keterangan
		<i>relay module.</i>


Gambar 4. 15 Stop Kontak  
 (sumber: <https://www.blibli.com/jual/stop-kontak-uticon-1>)

## 2. Perangkat lunak (*Software*)

Perangkat lunak atau software merupakan sebuah program yang terdapat pada perangkat komputer yang menjembatani antara user atau pengguna dengan perangkat keras (hardware) yang akan digunakan.

Berikut merupakan *software* yang digunakan pada tabel 4.3:

Tabel 4. 3 *Software*

No.	Nama dan Gambar	Keterangan
1.	Arduino IDE 	Arduino IDE atau <i>Integrated Development Environment</i> adalah suatu perangkat lunak yang berguna untuk membuat sketsa pemrograman. Perangkat lunak ini dapat digunakan sebagai pembuatan program, editing, serta dapat mengunggah program ke papan ESP-32. Bahasa yang digunakan adalah java yang dilengkapi dengan <i>library</i> C++ sehingga dapat membuat pengaturan operasi <i>input</i> dan <i>output</i> menjadi lebih mudah

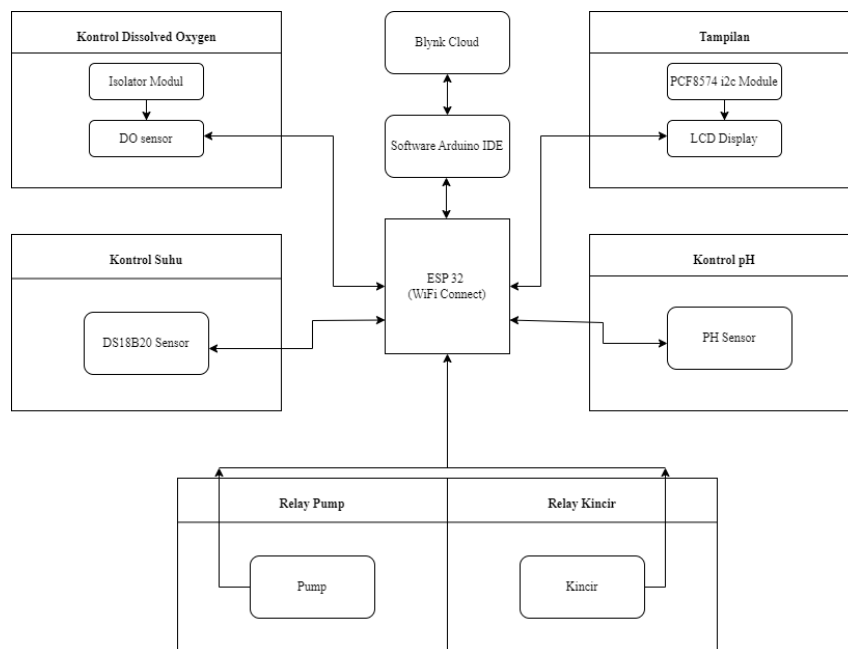
Gambar 4. 16 Arduino IDE  
 (sumber: <https://www.kmtech.id/post/mengenal-perangkat-lunak-arduino-ide>)

No.	Nama dan Gambar	Keterangan
2.	Blynk Cloud	<p>Blynk adalah sebuah perangkat lunak <i>open-source</i> yang mendukung penggunaan IoT (<i>Internet of Things</i>). Perangkat ini memungkinkan <i>user</i> untuk memonitor serta memberikan kontrol langsung terhadap perangkat keras yang telah dirangkai dengan menghubungkan pada mikrokontroler yang telah terhubung dengan WiFi dari jarak jauh menggunakan <i>widget</i> yang tersedia. Perangkat ini juga dapat diakses melalui laman web maupun aplikasi secara <i>real time</i>.</p>

Gambar 4. 17 Blynk Cloud  
(sumber: <https://shahmishafie.wordpress.com/197-2/>)

4.4.2 Diagram blok perancangan sistem

Berikut merupakan diagram blok awal pada gambar 4.18:



Gambar 4. 18 Diagram Blok Awal

Pada gambar diatas terdapat beberapa bagian yang memiliki fungsi berbeda satu sama lain. Berikut merupakan keterangan dari masing-masing bagian:

1. Kontrol *Dissolved Oxygen* (DO)

Sensor analog *dissolved oxygen* mengirimkan data oksigen terlarut pada kolam ke mikrokontroler ESP-32, kemudian data tersebut diolah berdasarkan kode yang telah dibuat. Mikrokontroler akan mengirimkan hasil oksigen dan diolah, jika melebihi batas atau kurang dari batas yang telah diprogram maka akan membuat kincir hidup secara otomatis menggunakan relay. Seluruh pertukaran informasi dalam sistem akan di unggah ke server *Blynk Cloud*.

2. Kontrol Suhu

Sensor suhu DS18B20 mengirimkan data suhu didalam kolam ke mikrokontroler ESP-32, kemudian data tersebut diolah berdasarkan program yang telah dibuat menggunakan arduino IDE . Mikrokontroler akan mengirimkan hasil suhu di dalam kolam dan akan menampilkan hasil suhu secara otomatis pada *Blynk Cloud* dan LCD *display*.

3. Tampilan

*Display* menggunakan I2C LCD display yang telah diprogram menggunakan arduino IDE dan mengirim hasil ke mikrokontroler ESP-32. Display akan menunjukkan hasil dari setiap sensor yang telah berjalan. Ketiga sensor akan mengirim sinyal hasil ke mikrokontroler dan display akan memberikan *output* berupa hasil dari pengukuran yang telah dibuat.

4. Kontrol pH

Analog pH akan mengukur pH air di dalam kolam sesuai dengan program yang telah dibuat pada arduino IDE dan akan mengirim hasil ke mikrokontroler ESP-32, setelah itu hasil pH yang telah dibaca akan mengirim sinyal ke mikrokontroler, jika kurang atau lebih dari batas yang telah ditentukan maka mikrokontroler akan mengirim sinyal ke relay *pump* untuk menghidupkan pompa sampai batas pH normal.

5. Relay *pump* dan Relay kincir

Relay *pump* akan hidup jika sensor pH memiliki hasil kurang atau lebih dari batas normal yang ditentukan, sensor akan mengirim sinyal ke mikrokontroler dan akan menghidupkan relay secara otomatis dan mati secara otomatis, begitu juga dengan relay kincir untuk menghidupkan kincir sesuai dengan hasil dari sensor DO, jika DO

tidak berada dalam kondisi normal maka relay akan hidup sesuai dengan perintah program yang diolah oleh mikrokontroler ESP-32.

## 4.5 Build Prototype

### 4.5.1 Persiapan awal Blynk Cloud

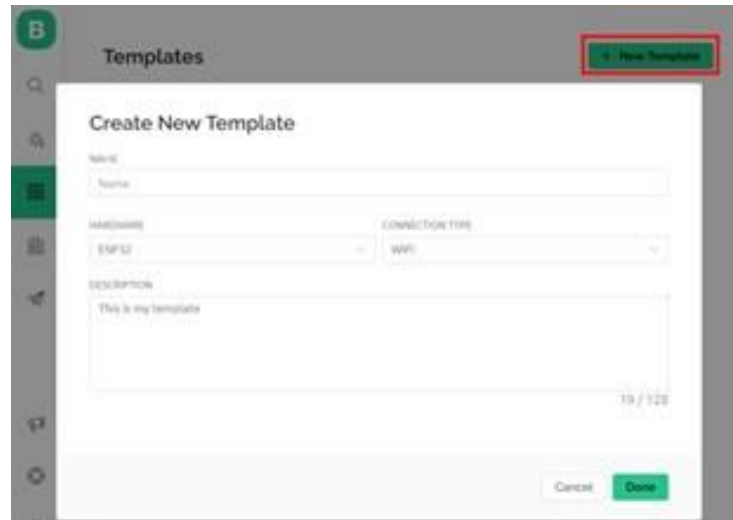
Persiapan awal yang dilakukan adalah melakukan penyetelan Blynk *Cloud* yang dapat digunakan melalui laman web <https://sgp1.blynk.cloud/dashboard/login> untuk mengendalikan mikrokontroler arduino yang telah terhubung dengan jaringan internet. *Platform* ini berguna untuk memudahkan *user* dalam melakukan kontrol terhadap mikrokontroler yang telah diprogram. Tahapan penyetelan awal dalam penggunaan Blynk *Cloud* sebagai berikut:

1. Tahapan awal adalah masuk ke laman web <https://sgp1.blynk.cloud/dashboard/login> dan melakukan pendaftaran dengan mengguna *email*, jika belum memiliki harap membuat *email* terlebih dahulu.



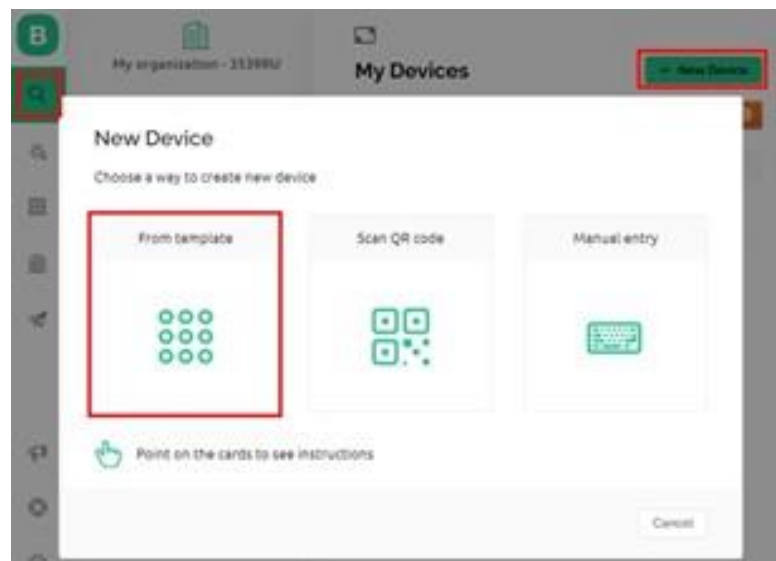
Gambar 4. 19 Log In

2. Setelah memiliki akun Blynk *Cloud*, klik pada bagian *Template* untuk membuat sistem baru “*Templates*” dan “+*New Template*”. Selanjutnya mengisikan nama proyek, *hardware* ESP32, dan *connection type* yaitu WiFi. Klik *done* apabila data telah dilengkapi.



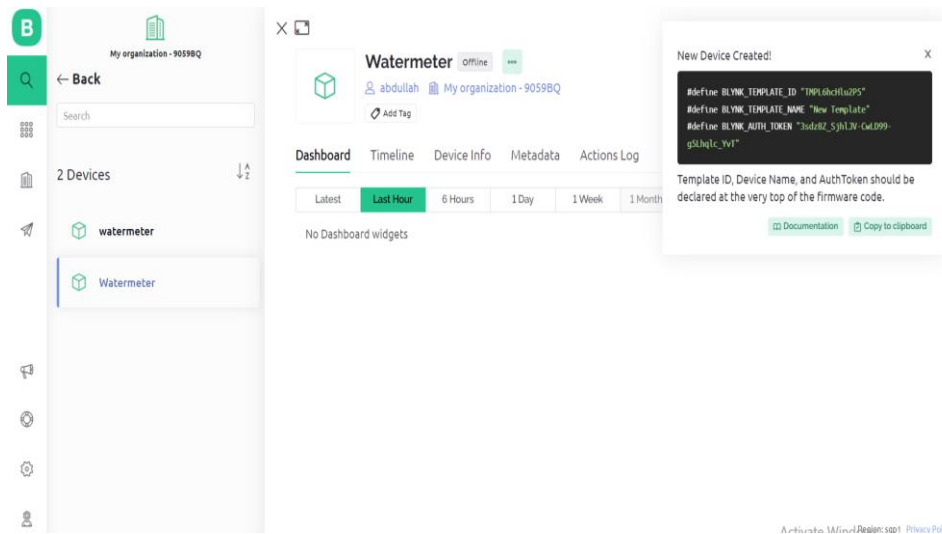
Gambar 4. 20 *Create New Template*

- Selanjutnya adalah membuat perangkat baru dengan cara klik lama *search* dan “+ *New Device*”. Pilih sistem yang telah dibuat pada tahap sebelumnya dengan mengklik *from template* dan pilih sesuai nama *template* yang telah dibuat.



Gambar 4. 21 *New Device*

- Setelah itu kembali pada menu “*Templates*” untuk mendapatkan *template\_id* dan *template\_name* yang akan digunakan untuk pengkodean pada Arduino IDE.



Gambar 4. 22 Templates Menu

#### 4.5.2 Pembangunan Dissolved Oxygen Sensor

Untuk melakukan pembangunan sensor DO atau *dissolved oxygen* diperlukan peralatan, skematik perancangan, pengaturan Blynk Cloud dan *script pseudo code* pada Arduino IDE.

##### 4.4.2.1 Peralatan *Dissolved Oxygen Sensor*

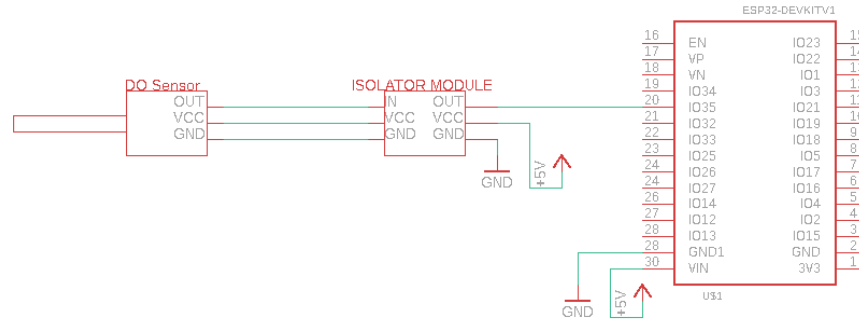
Dalam pembangunan rangkaian sistem DO sensor, diperlukan peralatan sebagai berikut:

Tabel 4. 4 Peralatan Sensor DO

No	Nama Alat	Jumlah
1	Mikrokontroler ESP-32	1
2	Isolator Modul	1
3	<i>Dissolved Oxygen Sensor</i>	1

##### 4.4.2.2 Skematik *Dissolved Oxygen Sensor*

Skematik *Dissolved Oxygen Sensor* dirancang menggunakan *software* Eagle. *Software* ini digunakan untuk membuat rancangan elektronik sebelum proses perangkaian dilakukan. Berikut merupakan skematik pada DO sensor:



Gambar 4. 23 Skematik Sensor DO

Keterangan:

Tabel 4. 5 Sambungan Skematik Sensor DO

Awalan	Sambungan
Pin IO35 pada ESP-32	Pin Out pada modul isolator
Pin GND1 pada ESP-32	<i>Ground</i>
Pin IN pada isolator modul	Pin <i>Out</i> pada DO sensor
Pin VCC pada isolator modul	Pin VCC pada DO sensor
Pin GND pada isolator modul	Pin GND pada DO sensor
Pin GND pada isolator modul	<i>Ground</i>

#### 4.4.2.3 Pengaturan Blynk Cloud *Dissolved Oxygen Sensor*

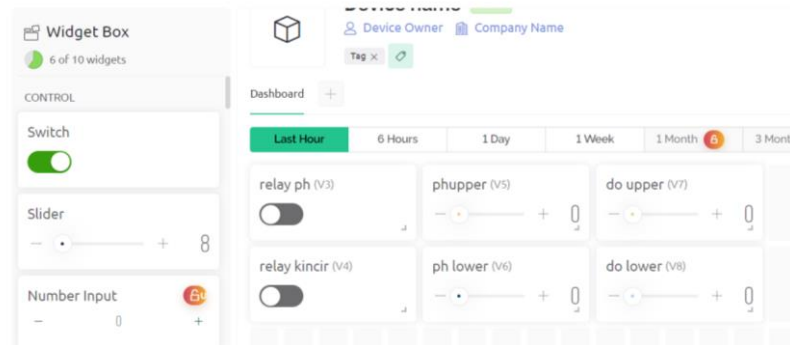
Sebelum masuk proses penerapan kode pada sistem, perlu dilakukan penyetelan *input* dan *output* pada Blynk Cloud. Berikut merupakan tahapan penyetelannya:

1. Klik menu “*Templates*”, kemudian *datastreams*. Klik “*+New Datasteam*”.

Gambar 4. 24 *Virtual Pin Datastream*

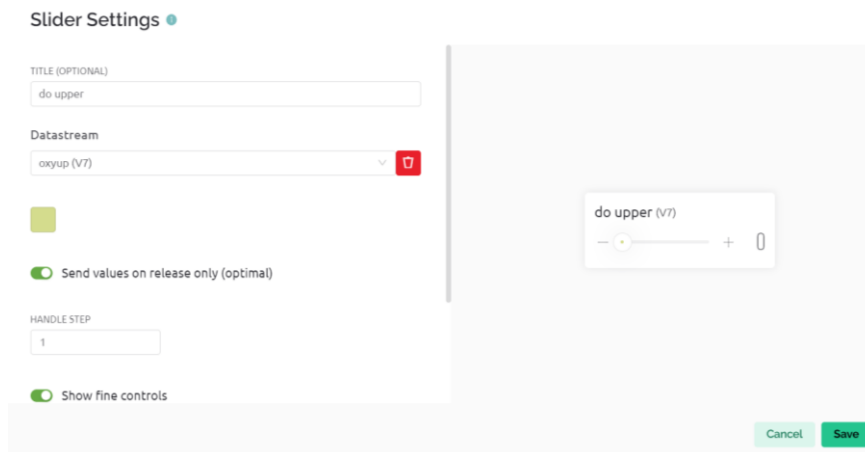


2. Selanjutnya masuk pada *web dashboard* dan klik *edit*. Pilih “*switch*” pada *widget* yang tersedia di menu sebelah kiri. Beri nama dan klik *save*.



Gambar 4. 25 *Widget Box*

3. Selanjutnya pilih kembali *widget* “*Slider*” pada menu di sebelah kiri. Isikan nama judul, datastream yang digunakan yaitu *V0* dan klik *save*.



Gambar 4. 26 *Slider Settings*

#### 4.4.2.4 *Script Dissolved Oxygen Sensor*

Program yang dibuat menggunakan aplikasi Arduino IDE yang dihubungkan menggunakan mikrokontroler ESP-32. Perinciannya sebagai berikut:

Tabel 4. 6 *Script Dissolved Oxygen Sensor*

<i>Pseudo Code</i>	<b>Fungsi</b>
#define BLYNK_TEMPLATE_ID "TMPL6p_M1CwOe"	Mendefinisikan <i>template id</i> Blynk yang telah didapat saat penyetelan awal

<i>Pseudo Code</i>	<b>Fungsi</b>
#define BLYNK_TEMPLATE_NAME "watermeter"	Mendefinisikan <i>name template</i>
#define BLYNK_AUTH_TOKEN "g4bu4y6oM7lZU42AqoqTURJ4TTs19UXZ"	Mendefinisikan token Blynk
#define BLYNK_PRINT Serial // tampilkan informasi blynk di serial	Mendefinisikan <i>serial print</i> Blynk sehingga dilakukan pemantauan melalui <i>serial monitor</i>
// input wifi dan password char auth[] = BLYNK_AUTH_TOKEN; char ssid[] = "mywifi"; // type your wifi name char pass[] = "354354354"; // type your wifi password BlynkTimer timers;	Melakukan input jaringan internet WiFi dari Id, ssid, hingga <i>password</i> WiFi yang digunakan
#include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27, 16, 2);	Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i>
#define led 2 #define pin_ph 34 #define pin_do 35 #define pin_ds 5 #define pin_relph 19 #define pin_relkr 18 #include <EEPROM.h> int doup,dodown;	<i>Input library</i> LCD Melakukan definisi LCD I2C pada <i>library</i> Melakukan pembacaan sensor dengan pin yang terdapat pada mikrokontroler ESP-32 sesuai dengan nomor pin yang telah dipilih
float temperature;	<i>Input library</i> Memberikan tipe data untuk Doup dan Dodown
int16_t readDO(uint32_t voltage_mv, uint8_t temperature_c)	Memberikan tipe data untuk menunjukkan temperatur Memasukan jenis-jenis tipe data yang akan diolah dan dapat dibaca oleh mikrokontroler.
	Melakukan kalibrasi = 0, dan

<i>Pseudo Code</i>	<b>Fungsi</b>
<pre> { #if TWO_POINT_CALIBRATION == 0   uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35;   return (voltage_mv * DO_Table[temperature_c] / V_saturation); #else   uint16_t          V_saturation          = (int16_t)((int8_t)temperature_c  -  CAL2_T) * ((uint16_t)CAL1_V      -      CAL2_V)      / ((uint8_t)CAL1_T - CAL2_T) + CAL2_V;   return (voltage_mv * DO_Table[temperature_c] / V_saturation); #endif } BLYNK_WRITE(V7){ int pinValue = param.asInt(); doup = pinValue; Serial.print("do upper:"); Serial.println(pinValue); EEPROM.write(22,pinValue); EEPROM.commit(); } BLYNK_WRITE(V8){ int pinValue = param.asInt(); dodown = pinValue; Serial.print("do lower:"); Serial.println(pinValue); EEPROM.write(23,pinValue); EEPROM.commit(); }  if((int)do_val&gt;doup) { </pre>	<p>menghasilkan <i>output</i> persentase oksigen, temperatur, dan saturasi.</p> <p>Membaca <i>input</i> dari Blynk Cloud untuk DO upper dengan pin 22</p> <p>Sebaliknya untuk membaca <i>input</i> dari Blynk Cloud pada DO Lower pada pin 23</p> <p>Jika DO lebih besar dari do <i>value</i> maka kincir oksigen akan menyala</p>

<i>Pseudo Code</i>	<b>Fungsi</b>
<pre> if(lock2!=1){ lock2=1; digitalWrite(pin_relkr, 1); } } if((int)do_val&lt;dodown){ if(lock2!=2){ lock2=2; digitalWrite(pin_relkr, 0); }} void loop() { Blynk.run(); timer.run(); </pre>	<p>Jika DO lebih kecil dari <i>do value</i> maka kincir akan mati</p> <p>Menjalankan <i>library</i> Blynk serta <i>timer</i> Blynk agar dapat berkomunikasi dengan server.</p>

#### 4.4.3 Pembangunan Suhu (*Temperature*)

Untuk melakukan pembangunan sensor suhu diperlukan peralatan, skematik perancangan, pengaturan Blynk Cloud dan *script pseudo code* pada Arduino IDE.

##### 4.4.3.1 Peralatan suhu (*temperature*)

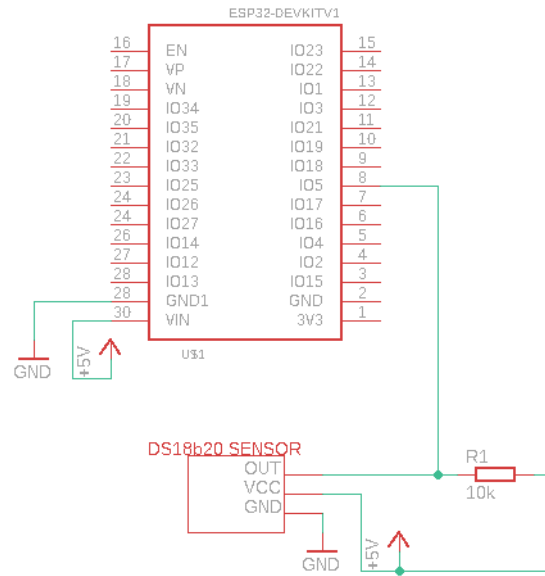
Dalam pembangunan sistem kontrol temperatur, diperlukan peralatan sebagai berikut:

Tabel 4. 7 Peralatan Suhu

<b>No</b>	<b>Nama Alat</b>	<b>Jumlah</b>
1	Mikrokontroler ESP-32	1
2	DS18B20 Sensor	1

##### 4.4.3.2 Skematik suhu (*temperature*)

Skematik kontrol temperatur dirancang menggunakan *software* Eagle. Software ini berguna untuk mendesain rangkaian elektronik sebelum proses perangkaian sebenarnya dilakukan. Berikut merupakan skematik pada kontrol temperatur:



Gambar 4. 27 Skematik Suhu

Keterangan:

Tabel 4. 8 Sambungan Skematik Sensor Suhu

Awalan	Sambungan
Pin IO5 pada ESP-32	Out R1
Input R1	Output DS18B20 Sensor
VCC pada DS18B20 Sensor	Pin 5v
GND pada DS18B20 Sensor	Ground
Pin GND1 pada ESP-32	Ground

#### 4.4.3.3 *Script* kontrol temperatur

Berikut merupakan tabel *script* yang berfungsi sebagai penjelasan dari pengkodean sistem yang dilakukan pada *software* Arduino IDE dan diinput ke mikrokontroler ESP-32:

Tabel 4. 9 *Script* Sensor Suhu

<i>Pseudo Code</i>	Fungsi
// input wifi dan password	Menghubungkan jaringan WiFi
char auth[] = BLYNK_AUTH_TOKEN;	
char ssid[] = "mywifi"; // type your wifi name	Melakukan penyetelan WiFi dimulai dari ssid, nama WiFi, dan <i>password</i> .
char pass[] = "354354354"; // type your wifi password	
#include <LiquidCrystal_I2C.h>	
LiquidCrystal_I2C lcd(0x27, 16, 2);	<i>Input library</i> LCD Memasukkan jenis LCD (0x27, 16, 2)
#include <OneWire.h>	
	<i>Input library</i> OneWire.h untuk protokol master
#include <DallasTemperature.h>	
OneWire oneWire(pin_ds);	<i>Input library</i> DallasTemperature.h untuk mengatur suhu
DallasTemperature sensors(&oneWire);	<i>Input sensor</i> suhu melalui pin D3
float temperature;	Memberikan tipe data untuk menunjukkan

<i>Pseudo Code</i>	<b>Fungsi</b>
void tasktimer(){	temperatur
sensors.requestTemperatures();	Untuk menunjukkan fungsi yang tidak dikembalikan
temperature	= Untuk menjalankan sensor dan meminta sensor berjalan
sensors.getTempCByIndex(0);	
lcd.setCursor(9,0);	Tempat awal kursor LCD
lcd.print(temperature);	LCD menunjukkan tulisan <i>temperature</i>
lcd.write(0xdf);	LCD menunjukkan <i>output</i> "C"
lcd.print("C");	
void loop() {	Menjalankan <i>library</i> Blynk serta <i>timer</i>
Blynk.run();	Blynk agar dapat berkomunikasi dengan server.
timer.run();	

#### 4.4.4 Pembangunan sistem pH

Untuk melakukan pembangunan sensor pH atau *potential hydrogen* diperlukan peralatan, skematik perancangan, pengaturan Blynk Cloud dan *script pseudo code* pada Arduino IDE.

##### 4.4.4.1 Peralatan sistem pH

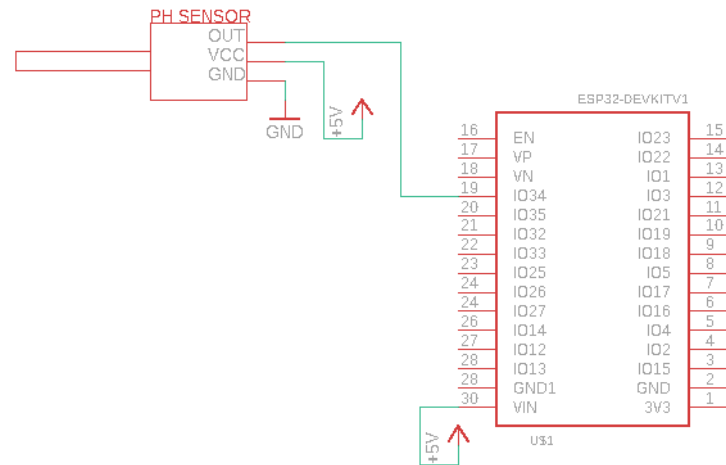
Dalam pembangunan sistem pH, diperlukan peralatan sebagai berikut:

Tabel 4. 10 Peralatan Sistem pH

<b>No</b>	<b>Nama Alat</b>	<b>Jumlah</b>
1	Mikrokontroler ESP-32	1
2	pH Sensor Analog	1

##### 4.4.4.2 Skematik sistem pH

Skematik sistem pH dirancang menggunakan *software* Eagle. *Software* ini berguna dalam mendesain rangkaian elektronik sebelum proses perangkaian sebenarnya dilakukan. Berikut merupakan skematik pada sistem pencahayaan.



Gambar 4. 28 Skematik sistem pH

Keterangan:

Tabel 4. 11 Sambungan Skematik Sistem pH

Awalan	Sambungan
Pin IO34 pada ESP-32	Pin Out PH sensor analog
PIN VIN pada ESP-32	Ground
PIN VCC pada pH sensor analog	5v
Pin GND pada pH sensor	Ground

#### 4.4.4.3 Pengaturan Blynk Cloud pH Sensor Analog

Sebelum masuk ke proses pengkodean sistem, perlu dilakukan penyetelan Blynk Cloud. Berikut merupakan tahapannya:

1. Masuk pada menu “*Templates*”, kemudian *datastreams*. Pada *datastreams* klik “+ *New Datastream*” berupa *Virtual Pin*.



Virtual Pin Datastream

**View mode**  
You are in a "View" mode. Tap on "Edit" button in the top right corner to make changes.

NAME: ph ALIAS: ph

PIN: V1 DATA TYPE: Double

UNITS: None

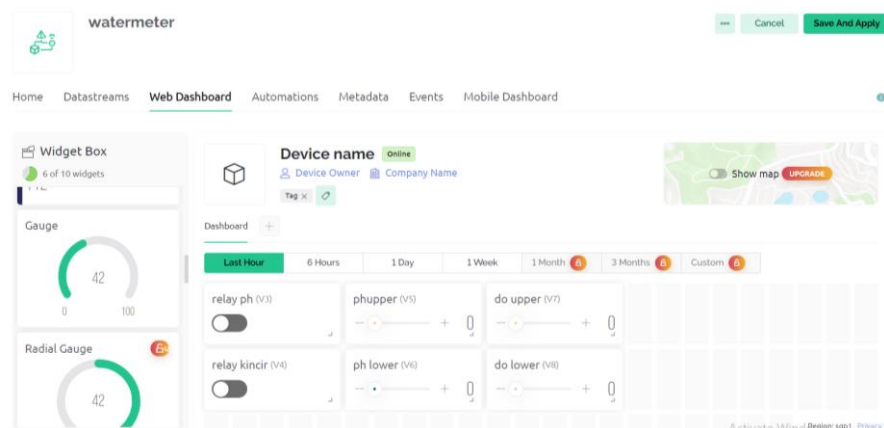
MIN: 0 MAX: 15 DECIMALS: #.00 DEFAULT VALUE: Default Value

ADVANCED SETTINGS

Close

Gambar 4. 29 Virtual Pin Datastream

- Selanjutnya masuk pada *web dashboard* dan klik edit. Pilih "Gauge" pada *widget* yang disediakan pada menu di sebelah kiri.



Gambar 4. 30 Widget

#### 4.4.4.4 Script sensor pH

Berikut merupakan *script* yang berfungsi untuk menjelaskan pengkodean sistem yang dilakukan pada *software* Arduino IDE dan diinput kedalam mikrokontroler ESP-32:

Tabel 4. 12 Script Sensor pH

<i>Pseudo Code</i>	Fungsi
<code>#define BLYNK_TEMPLATE_ID "TMPL6p_M1CwOe"</code>	Mendefinisikan template id Blynk yang telah didapat saat penyetelan awal.
<code>#define BLYNK_TEMPLATE_NAME "watermeter"</code>	Mendefinisikan nama template Blynk yang telah didapat saat penyetelan awal.

<i>Pseudo Code</i>	<b>Fungsi</b>
<pre>#define BLYNK_AUTH_TOKEN " g4bu4y6oM7IZU42AqoqTURJ4TTs19UXZ"  #define BLYNK_PRINT Serial  #include &lt;BlynkSimpleEsp32.h&gt; char auth[] = BLYNK_AUTH_TOKEN;  char ssid[] = "mywifi"; char pass[] = "354354354"; BlynkTimer timers;  #include &lt;LiquidCrystal_I2C.h&gt; LiquidCrystal_I2C lcd(0x27, 16, 2);  #include "ph_lib.h" #define pin_ph 34  float ph,voltage,do_val;  #include &lt;EEPROM.h&gt; int phup,phdown;  BLYNK_WRITE(V5){ int pinValue = param.asInt(); phup = pinValue; Serial.print("ph upper:"); Serial.println(pinValue);</pre>	<p>Mendefinisikan token autentikasi yang telah didapat saat penyetelan awal.</p> <p>Mendefinisikan <i>serial print</i> Blynk agar dapat dilakukan pemantauan melalui <i>serial monitor</i>.</p> <p><i>Input library</i> Blynk Cloud</p> <p>Memasukkan token autentikasi yang telah didefinisikan sebelumnya</p> <p><i>Input</i> WiFi meliputi ssid WiFi, Name WiFi, dan <i>Password</i>.</p> <p>Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i>.</p> <p>Menginput <i>library</i> LCD.</p> <p>Berfungsi sebagai pendefinisian LCD I2C pada <i>library</i>.</p> <p><i>Input library</i> pH</p> <p>Mendefinisi pin sensor pH ada di nomor 34</p> <p>Masukan tipe data untuk membaca pH</p> <p>Masukan <i>library</i> EEPROM.h</p> <p>Tipe data untuk pH up dan pH down</p> <p>Membaca input dari Blynk untuk pH upper dan hasil yang didapatkan</p>

<i>Pseudo Code</i>	<b>Fungsi</b>
<pre>EEPROM.write(20,pinValue); EEPROM.commit(); } BLYNK_WRITE(V6){   int pinValue = param.asInt();   phdown = pinValue;   Serial.print("ph lower:");   Serial.println(pinValue);   EEPROM.write(21,pinValue);   EEPROM.commit();    if((int)ph&lt;phdown  ((int)ph&gt;phup) {     if(lock1!=1){       lock1=1;       digitalWrite(pin_relph, 0);     }   }   else{     if(lock1!=2){       lock1=2;       digitalWrite(pin_relph, 1);     }   }   void loop() {     Blynk.run();     timer.run();</pre>	<p>Membaca input dari Blynk untuk pH lower dan hasil yang didapatkan</p> <p>Jika pH normal maka relpH tidak akan aktif atau = 0</p> <p>Jika pH tidak pada batas normal maka relpH akan aktif atau = 1</p> <p>Menjalankan library Blynk serta timer Blynk agar dapat berkomunikasi dengan server.</p>

#### 4.4.5 Pembangunan Sistem *Relay pump* dan *Relay* kincir

##### 4.4.5.1 Peralatan sistem *relay pump* dan *relay* kincir

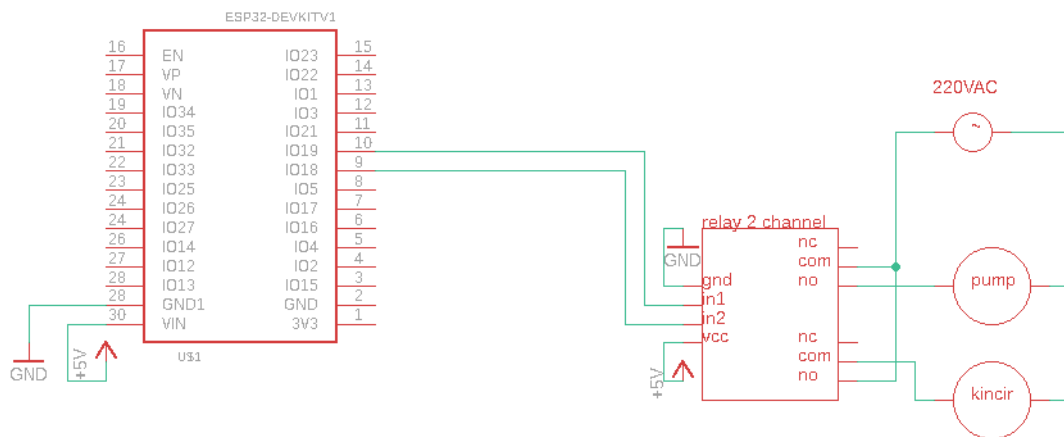
Pada pembangunan sistem *relay pump* dan *relay* kincir diperlukan peralatan sebagai berikut:

Tabel 4. 13 Peralatan *Relay*

No	Nama Alat	Jumlah
1.	Mikrokontroler ESP-32	1
2.	Relay 2 <i>channel</i>	2
3.	Pompa	1
4.	Kincir	1

#### 4.4.5.2 Skematik sistem *relay pump* dan *relay kincir*

Skematik sistem *relay pump* dan *relay kincir* menggunakan *software* Eagle. *Software* ini berguna untuk mendesain rangkaian elektronik sebelum proses perangkaian sebenarnya dilakukan. Berikut merupakan skematik pada sistem *relay pump* dan *kincir*:

Gambar 4. 31 Skematik Sistem *Relay*

Keterangan:

Tabel 4. 14 Sambungan Sistem *Relay*

Awalan	Sambungan
Pin IO19 pada ESP-32	Pin IN 1 pada relay 2 <i>channel</i>
Pin IO18 pada ESP-32	Pin IN 2 pada relay 2 <i>channel</i>
Pin com pada relay 1	Port 220VAC
Pin no pada relay 1	Port pump
Pin com pada relay 2	Port kincir
Pin no pada relay 2	Port 220VAC
Pin GND1 pada ESP-32	GND
Pin GND pada relay 2 <i>channel</i>	GND

#### 4.4.5.3 Pengaturan Blynk Cloud sistem *relay pump* dan *relay kincir*

Sebelum memasuki proses penerapan kode pada sistem, perlu dilakukan penyetelan *input/output* pada Blynk Cloud. Berikut merupakan tahapan penyetelannya:

1. Masuk pada menu “*Templates*”, kemudian *datastreams*. Pada *datastreams* klik “+ *New Datastream*” berupa Virtual Pin. Isikan data untuk relay kincir.

Virtual Pin Datastream

NAME:  ALIAS:

PIN:  DATA TYPE:

UNITS:

MIN:  MAX:  DEFAULT VALUE:

ADVANCED SETTINGS

Gambar 4. 32 *Virtual Pin Datastream Kincir*

2. Masuk pada menu “*Templates*”, kemudian *datastreams*. Pada *datastreams* klik “+ *New Datastream*” berupa Virtual Pin. Isikan data untuk relay pompa.

Virtual Pin Datastream

NAME:  ALIAS:

PIN:  DATA TYPE:

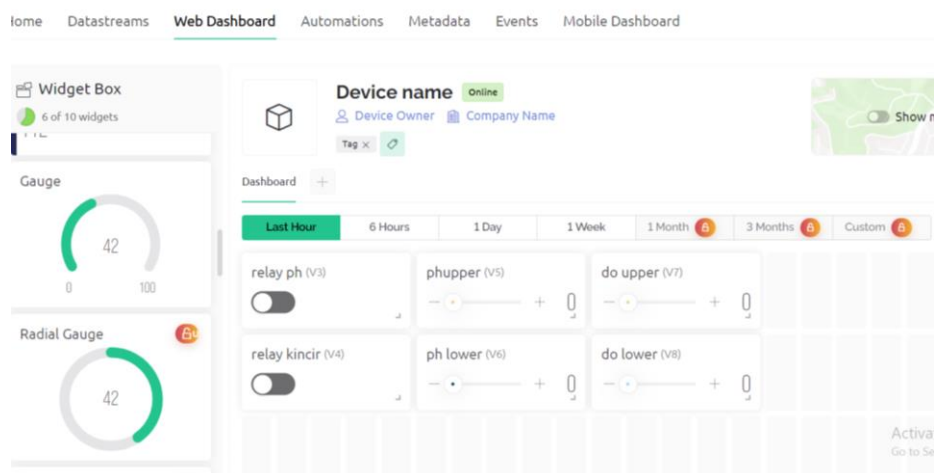
UNITS:

MIN:  MAX:  DEFAULT VALUE:

ADVANCED SETTINGS

Gambar 4. 33 *Virtual Pin Datasream pH*

3. Selanjutnya masuk pada *web dashboard* dan klik edit. Pilih “*Gauge*” pada *widget* yang disediakan pada menu di sebelah kiri.



Gambar 4. 34 Widget

#### 4.4.5.4 script sistem relay pump dan relay kincir

Berikut merupakan tabel *script* yang berfungsi sebagai penjelasan dari pengkodean sistem yang dilakukan pada *software* Arduino IDE dan diinput ke dalam mikrokontroler ESP-32:

Tabel 4. 15 Script Sistem Relay

<i>Pseudo Code</i>	<i>Fungsi</i>
<code>#define BLYNK_TEMPLATE_ID "TMPL6p_M1CwOe"</code>	Mendefinisikan template id Blynk yang telah didapat saat penyetelan awal.
<code>#define BLYNK_TEMPLATE_NAME "watermeter"</code>	Mendefinisikan nama template Blynk yang telah didapat saat penyetelan awal.
<code>#define BLYNK_AUTH_TOKEN "g4bu4y6oM7lZU42AqoqTURJ4TTs19UXZ"</code>	Mendefinisikan token autentikasi yang telah didapat saat penyetelan awal.
<code>#define BLYNK_PRINT Serial</code>	Mendefinisikan <i>serial print</i> Blynk agar dapat dilakukan pemantauan melalui <i>serial monitor</i> .
<code>#include &lt;BlynkSimpleEsp32.h&gt;</code>	<i>Input library</i> Blynk Cloud
<code>char auth[] = BLYNK_AUTH_TOKEN;</code>	Memasukkan token autentikasi yang telah didefinisikan sebelumnya
<code>char ssid[] = "mywifi";</code>	Memasukkan jaringan WiFi dengan menginput ssid Wifi, Nama WiFi,

<i>Pseudo Code</i>	<b>Fungsi</b>
<pre>char pass[] = "354354354";</pre>	<p>dan <i>password</i> Wifi</p> <p>Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i>.</p>
<pre>BlynkTimer timers;</pre>	<p>Menginput <i>library</i> LCD.</p> <p>Berfungsi sebagai pendefinisian LCD I2C pada <i>library</i></p>
<pre>#include &lt;LiquidCrystal_I2C.h&gt; LiquidCrystal_I2C lcd(0x27, 16, 2);</pre>	<p>Membaca <i>input</i> dari Blynk Cloud dan menampilkan <i>output</i> pada serial monitor untuk pompa</p>
<pre>BLYNK_WRITE(V3){   int pinValue = param.asInt();   if(pinValue==1) digitalWrite(pin_relph, 0);   if(pinValue==0) digitalWrite(pin_relph, 1);   Serial.print("R PH:");   Serial.println(pinValue); }</pre>	<p>Membaca <i>input</i> dari Blynk Cloud dan menampilkan <i>output</i> pada serial monitor untuk kincir oksigen</p>
<pre>BLYNK_WRITE(V4){   int pinValue = param.asInt();   if(pinValue==1) digitalWrite(pin_relkcr, 0);   if(pinValue==0) digitalWrite(pin_relkcr, 1);   Serial.print("R KINCIR:");   Serial.println(pinValue); }</pre>	<p>Untuk membuat relay mengikuti perintah secara otomatis jika pH berada dalam kondisi normal maka relay tidak akan menyala</p>
<pre>if((int)ph&lt;phdown  ((int)ph&gt;phup) {   if(lock1!=1){     lock1=1;     digitalWrite(pin_relph, 0);   } }</pre>	<p>Jika pH tidak berada dalam kondisi normal maka relay akan menyala.</p>
<pre>else{</pre>	<p>Untuk membuat relay mengikuti perintah secara otomatis jika Oksigen tidak berada dalam kondisi normal maka relay akan menyala</p>

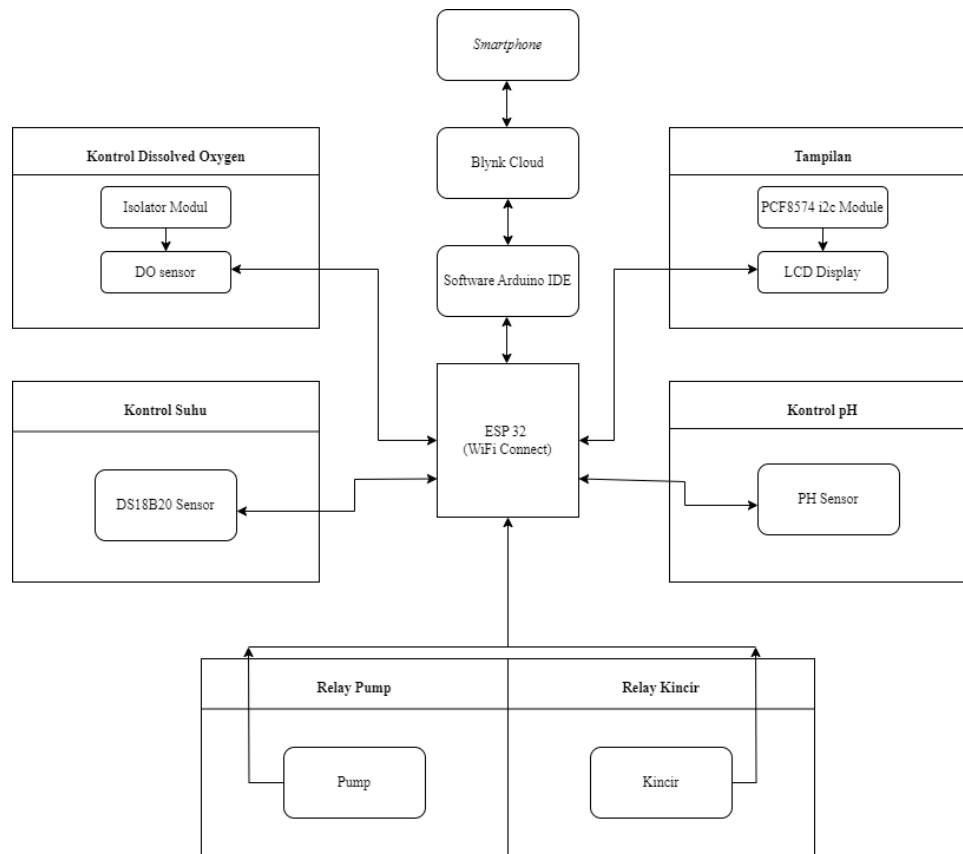
<i>Pseudo Code</i>	<b>Fungsi</b>
<pre> if(lock1!=2){   lock1=2;   digitalWrite(pin_relph, 1); } } </pre>	<p>Jika oksigen berada dalam kondisi normal, maka relay tidak akan menyala</p>
<pre> if((int)do_val&gt;doup) {   if(lock2!=1){     lock2=1;     digitalWrite(pin_relkr, 1);   } } </pre>	<p>Menjalankan <i>library</i> Blynk serta <i>timer</i> Blynk agar dapat berkomunikasi dengan server.</p>
<pre> if((int)do_val&lt;dodown){   if(lock2!=2){     lock2=2;     digitalWrite(pin_relkr, 0);   } } </pre>	
<pre> void loop() {   Blynk.run();   timer.run(); } </pre>	

## 4.6 Evaluasi

### 4.6.1 Evaluasi *prototype*

Sistem yang telah dibuat dalam bentuk *prototype* diberikan dan dipresentasikan kepada Bapak Yogi sebagai pemilik Minaverse Company untuk dilakukan evaluasi pengguna terkait sistem. Berdasarkan hasil diskusi langsung dengan pemilik maka peneliti mengambil kesimpulan terkait diperlukannya kontrol penuh untuk memonitor langsung sistem automasi pengelolaan budidaya ikan yang menggunakan telepon genggam atau *smartphone* serta tambahan notifikasi sebagai pengingat apabila pH air, Oksigen, dan suhu tidak berada dalam kondisi normal. Berdasarkan hal tersebut maka dilakukan perbaikan mengenai diagram blok seperti dibawah ini:





Gambar 4. 35 Evaluasi *Prototype*

#### 4.6.2 Perbaikan *prototype*

Berdasarkan hasil evaluasi yang telah dilakukan, terdapat hasil diskusi antara *user* dan *developer* yang diperlukan perbaikan. Hal yang dibutuhkan perbaikan berdasarkan hasil diskusi antara lain yaitu, pemanfaatan *smartphone* untuk melakukan monitor jarak jauh dan melakukan kontrol sistem, maka dari itu, tahapan perbaikan yang dilakukan antara lain:

##### 4.6.2.1 *Monitoring* dan *controlling* menggunakan *smartphone*

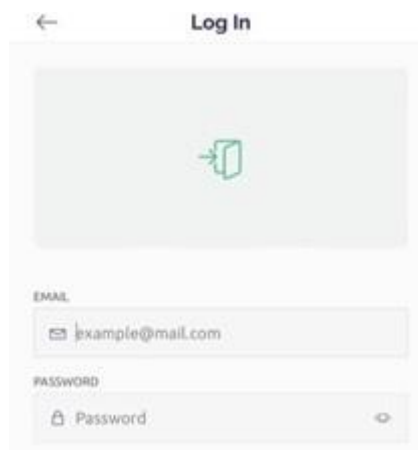
Web Blynk Cloud sebenarnya sudah cukup untuk melakukan kegiatan *controlling* dan *monitoring*, tetapi untuk memudahkan *user* dapat melakukan alternatif lain dengan menggunakan aplikasi yang dapat di unduh menggunakan *smartphone* dari *user*. Berikut merupakan tahapan dalam melakukan hal tersebut:

1. Unduh aplikasi Blynk IoT pada *smartphone*



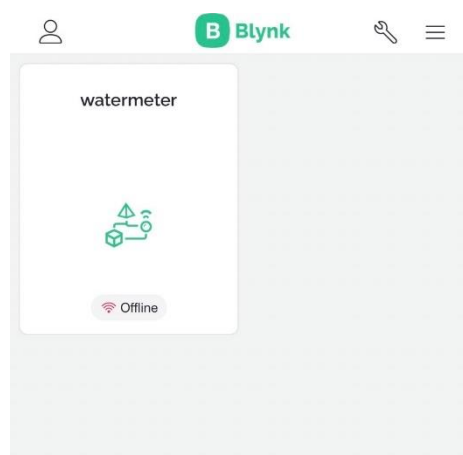
Gambar 4. 36 Unduh Blynk IoT

2. Lakukan *login* dengan akun yang telah dibuat pada akun Blynk Cloud sebelumnya.



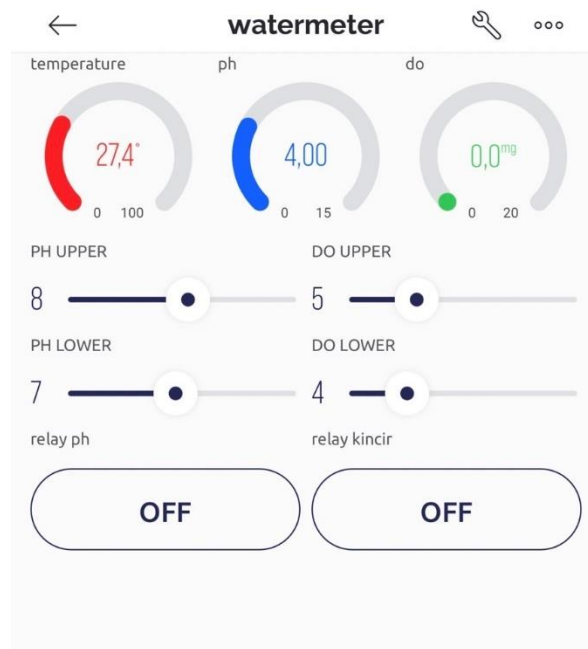
Gambar 4. 37 Login

3. Pada tampilan utama akan muncul perangkat yang telah dibuat pada Blynk Cloud sebelumnya. Klik *device* tersebut.



Gambar 4. 38 Tampilan Aplikasi

4. Pada tampilan akhir akan terlihat *widget* yang telah dibuat pada *web* Blynk Cloud serta dapat diatur pada bagian ini.



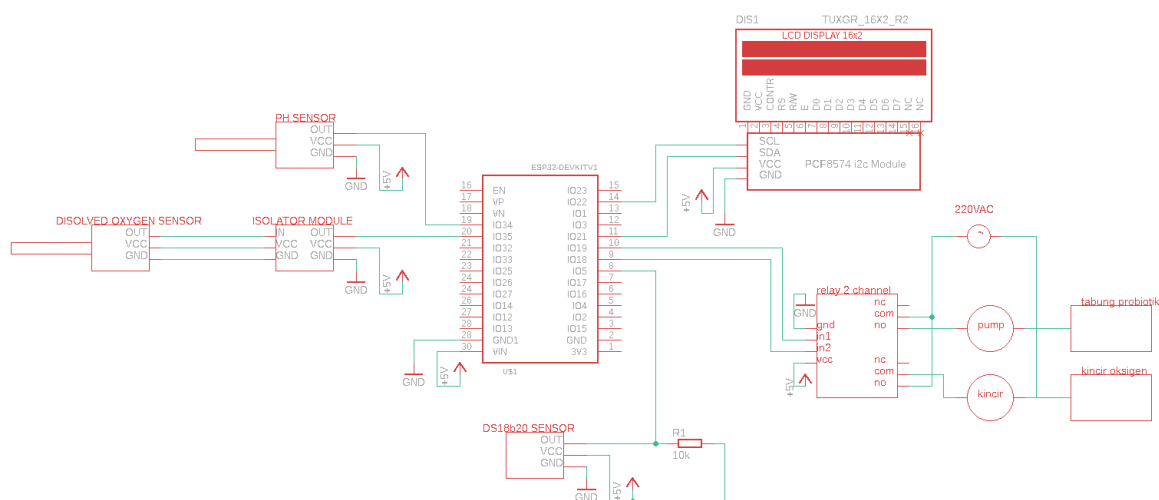
Gambar 4. 39 Watermeter

#### 4.7 Pembuatan dan Perakitan Produk

Berdasarkan hasil evaluasi *prototype* yang telah dibuat, telah ditentukan bentuk akhir dari produk automasi budidaya ikan dan telah memenuhi kriteria yang dibutuhkan *user*, maka *developer* membentuk produk yang sesuai serta memenuhi kebutuhan *user*.

##### 4.7.1 Desain skematik keseluruhan

Skematik sistem dirancang secara menyeluruh menggunakan *software* Eagle, desain skematik dapat memudahkan pihak pengembang dalam membangun rangkaian elektronik. Berikut merupakan skematik akhir pada sistem automasi pengelolaan budidaya ikan nila.



Gambar 4. 40 Skematik Keseluruhan

Keterangan:

Tabel 4. 16 Sambungan Keseluruhan

<b>Awalan</b>	<b>Sambungan</b>
Pin IO35 pada ESP-32	Pin Out pada modul isolator
Pin GND1 pada ESP-32	<i>Ground</i>
Pin IN pada isolator modul	Pin <i>Out</i> pada DO sensor
Pin VCC pada isolator modul	Pin VCC pada DO sensor
Pin GND pada isolator modul	Pin GND pada DO sensor
Pin GND pada isolator modul	<i>Ground</i>
Pin IO5 pada ESP-32	Out R1
Input R1	Output DS18B20 Sensor
VCC pada DS18B20 Sensor	Pin 5v
GND pada DS18B20 Sensor	Ground
Pin GND1 pada ESP-32	Ground
Pin IO34 pada ESP-32	Pin Out PH sensor analog
PIN VIN pada ESP-32	Ground
PIN VCC pada pH sensor analog	5v
Pin GND pada pH sensor	Ground
Pin IO19 pada ESP-32	Pin IN 1 pada relay 2 <i>channel</i>
Pin IO18 pada ESP-32	Pin IN 2 pada relay 2 <i>channel</i>
Pin com pada relay 1	Port 220VAC
Pin no pada relay 1	Port pump
Pin com pada relay 2	Port kincir
Pin no pada relay 2	Port 220VAC
Pin GND1 pada ESP-32	GND
Pin GND pada relay 2 <i>channel</i>	GND
Pin IO21 pada ESP-32	Pin SDA pada LCD
Pin IO22 pada ESP-32	Pin SCL pada LCD
Pin VCC pada LCD	5V
Pin GND pada LCD	GND

#### 4.7.2 *Script* Keseluruhan

Berikut merupakan *script* keseluruhan yang merupakan penjelasan dari pengkodean sistem yang dilakukan pada *software* Arduino IDE dan diinput kedalam mikrokontroler ESP-32.

Tabel 4. 17 *Script* Keseluruhan

<i>Pseudo code</i>	<b>Fungsi</b>
#define BLYNK_TEMPLATE_ID "TMPL6p_M1CwOe"	Mendefinisikan template id Blynk yang telah didapat saat penyetelan awal.
#define BLYNK_TEMPLATE_NAME "watermeter"	Mendefinisikan nama template Blynk yang telah didapat saat penyetelan awal.
#define BLYNK_AUTH_TOKEN "g4bu4y6oM7lZU42AqoqTURJ4TTs19UXZ"	Mendefinisikan token autentikasi yang telah didapat saat penyetelan awal.
#define BLYNK_PRINT Serial // tampilkan informasi blynk di serial	Untuk menampilkan informasi Blynk di serial monitor Memasukkan <i>library</i> Blynk
#include <BlynkSimpleEsp32.h> // library blynk char auth[] = BLYNK_AUTH_TOKEN;	Memasukkan token autentikasi yang telah didefinisikan sebelumnya
char ssid[] = "mywifi"; // type your wifi name char pass[] = "354354354"; // type your wifi password	Memasukkan informasi SSID WiFi Memasukkan <i>password</i> koneksi WiFi yang digunakan.
BlynkTimer timers;	Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i> .
#include "ph_lib.h"	Memasukkan <i>library</i> pH
#include "eeprom_lib.h"	Memasukkan <i>library</i> eeprom
#define led 2 #define pin_ph 34 #define pin_do 35 #define pin_ds 5	Mendefinisikan Led, pin pH, pin DO, pin DS, rel pH, dan rel Kincir sesuai dengan nomor pin yang telah dipilih sebelumnya.
#define pin_relph 19	Memasukkan <i>library</i> Wire.h
#define pin_relkr 18	Memasukkan <i>library</i> LCD

<i>Pseudo code</i>	<b>Fungsi</b>
<code>#include &lt;Wire.h&gt;</code>	Mendefinisikan LCD sesuai
<code>#include &lt;LiquidCrystal_I2C.h&gt;</code>	jenis LCD yang digunakan
<code>LiquidCrystal_I2C lcd(0x27, 16, 2);</code>	Memasukkan tipe data untuk pH, voltage, dan DO
<code>float ph, voltage, do_val;</code>	Memasukkan <i>library</i> OneWire.h
<code>#include &lt;OneWire.h&gt;</code>	Memasukkan <i>library</i> DallasTemperature.h
<code>#include &lt;DallasTemperature.h&gt;</code>	Menjelaskan sensor suhu terhubung pada pin D3
<code>OneWire oneWire(pin_ds);</code>	Menjelaskan tipe sensor
<code>DallasTemperature sensors(&amp;oneWire);</code>	Memasukkan <i>library</i> EEPROM.h
<code>#include &lt;EEPROM.h&gt;</code>	Melakukan integrasi untuk pH dan DO
<code>int phup, phdown;</code>	
<code>int doup, dodown;</code>	Memasukkan tipe data untuk pembacaan temperatur
<code>int lock1, lock2;</code>	
<code>float temperature;</code>	Melakukan definisi antara VREF dengan nilai 3300 dan ADC_RES dengan nilai 4095
<code>#define VREF 3300</code>	
<code>#define ADC_RES 4095</code>	Melakukan definisi untuk melakukan kalibrasi DO
<code>#define TWO_POINT_CALIBRATION 0</code>	Untuk kalibrasi memerlukan CAL1_V dan CAL1_T
<code>int CAL1_V;</code>	Melakukan definisi CAL1_T dengan <i>output celcius</i>
<code>#define CAL1_T (25)</code>	Melakukan definisi CAL2_V dengan <i>output mv</i>
<code>#define CAL2_V (1300)</code>	Melakukan definisi CAL2_T dengan <i>output celcius</i>
<code>#define CAL2_T (15)</code>	Memasukkan bilangan untuk membuat tabel DO 14460, 14220, 13820, 13440, 13090,

<i>Pseudo code</i>	<b>Fungsi</b>
<pre>const uint16_t DO_Table[41] = { 14460, 14220, 13820, 13440, 13090, 12740, 12420, 12110, 11810, 11530, 11260, 11010, 10770, 10530, 10300, 10080, 9860, 9660, 9460, 9270, 9080, 8900, 8730, 8570, 8410, 8250, 8110, 7960, 7820, 7690, 7560, 7430, 7300, 7180, 7070, 6950, 6840, 6730, 6630, 6530, 6410};</pre>	<p>12740, 12420, 12110, 11810, 11530, 11260, 11010, 10770, 10530, 10300, 10080, 9860, 9660, 9460, 9270, 9080, 8900, 8730, 8570, 8410, 8250, 8110, 7960, 7820, 7690, 7560, 7430, 7300, 7180, 7070, 6950, 6840, 6730, 6630, 6530, 6410.</p>
<pre>uint8_t Temperatur; uint16_t ADC_Raw; uint16_t ADC_Voltage; uint16_t DO; int16_t readDO(uint32_t voltage_mv, uint8_t temperature_c) #if TWO_POINT_CALIBRATION == 0 uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35; return (voltage_mv * DO_Table[temperature_c] / V_saturation); #else uint16_t V_saturation = (int16_t)((int8_t)temperature_c - CAL2_T) * ((uint16_t)CAL1_V - CAL2_V) / ((uint8_t)CAL1_T - CAL2_T) + CAL2_V; return (voltage_mv*DO_Table[temperature_c] / V_saturation); #endif }</pre>	<p>Melakukan input bilangan mulai dari temperature, ADC_Raw, ADC_Volt. Dan DO</p> <p>Untuk mendapatkan hasil dari DO</p> <p>Rumus ini melibatkan V_saturation yang dihitung menggunakan nilai CAL1_V, temperature_c dan CAL1_T</p> <p>Untuk rumus ini melibatkan V_saturation, yang dihitung menggunakan nilai CAL2_V, VAL2_T, temperature_c, CAL1_V, dan CAL1_T.</p> <p>Setelah nilai V_saturation dihitung, nilai akhir dikembalikan oleh fungsi untuk dihitung dengan membagi voltage_mv dengan v_saturation, kemudian dikembalikan dengan DO_Table</p> <p>Membaca input dari Blynk,</p>

<i>Pseudo code</i>	<b>Fungsi</b>
<pre> BLYNK_WRITE(V3){  int pinValue = param.asInt();  if(pinValue==1) digitalWrite(pin_relph, 0);  if(pinValue==0) digitalWrite(pin_relph, 1);  Serial.print("R PH:");   Serial.println(pinValue); } </pre>	<p>mendefinisikan pin V3</p> <p>mengambil nilai terkini dari pin virtual V3 dan menyimpannya dalam variabel setelah itu untuk mengonversi nilai tersebut menjadi bilangan bulat (integer).</p> <p>Mengatur pin pin_relph menjadi keadaan rendah. Pin tersebut akan dimatikan atau mematikan perangkat yang terhubung ke pin tersebut.</p> <p>mengatur pin pin_relph menjadi keadaan tinggi. Pin tersebut akan dihidupkan atau mengaktifkan perangkat yang terhubung ke pin tersebut.</p> <p>Menampilkan hasil dari perhitungan pH yang ditampilkan pada serial monitor</p> <p>Membaca <i>input</i> dari Blynk pada pin V4.</p> <p>Digunakan untuk mengambil nilai saat ini dari pin virtual V4 dan menyimpannya dalam variabel pinValue. Digunakan untuk mengonversi nilai tersebut menjadi bilangan bulat.</p>
<pre> BLYNK_WRITE(V4){  int pinValue = param.asInt(); </pre>	<p>Mengatur pin pin_relker menjadi keadaan rendah (LOW). Pin tersebut akan mematikan atau mematikan perangkat yang terhubung ke</p>



<i>Pseudo code</i>	<b>Fungsi</b>
<code>if(pinValue==1) digitalWrite(pin_relkr, 0);</code>	pin tersebut. Mengatur pin pin_relkr menjadi keadaan tinggi (HIGH). Pin tersebut akan hidup atau menghidupkan perangkat yang terhubung ke pin tersebut.
<code>if(pinValue==0) digitalWrite(pin_relkr, 1);</code>	pin tersebut. <i>Output</i> R kincir akan ditampilkan dalam bentuk serial monitor.
<code>Serial.print("R KINCIR:");</code>	Membaca <i>input</i> dari Blynk pada pin V5
<code>Serial.println(pinValue);</code>	Digunakan untuk mengambil nilai saat ini dari pin virtual V5 dan menyimpannya dalam variabel pinValue. Digunakan untuk mengonversi nilai tersebut menjadi bilangan bulat.
<code>BLYNK_WRITE(V5){</code>	Memasukkan nilai phup kedalam pinValue
<code>int pinValue = param.asInt();</code>	Menampilkan <i>output</i> ke serial monitor terkait pH upper
<code>phup = pinValue;</code>	Menampilkan <i>output</i> ke serial monitor terkait pin value
<code>Serial.print("ph upper:");</code>	Digunakan untuk menulis nilai pinValue ke alamat EEPROM dengan alamat 20.
<code>Serial.println(pinValue);</code>	Digunakan untuk mengkomit data yang telah ditulis ke EEPROM.
<code>EEPROM.write(20,pinValue);</code>	Membaca <i>input</i> dari Blynk pada pin V6
<code>EEPROM.commit();</code>	Digunakan untuk mengambil nilai saat ini dari pin virtual V6

<i>Pseudo code</i>	<b>Fungsi</b>
<pre> }  BLYNK_WRITE(V6){  int pinValue = param.asInt();      phdown = pinValue;      Serial.print("ph lower:");      Serial.println(pinValue);      EEPROM.write(21,pinValue);      EEPROM.commit(); }  BLYNK_WRITE(V7){  int pinValue = param.asInt(); </pre>	<p>dan menyimpannya dalam variabel pinValue. Digunakan untuk mengonversi nilai tersebut menjadi bilangan bulat. Memasukkan phdown kedalam pinValue</p> <p>Menampilkan hasil dari ph lower ke serial monitor</p> <p>Menampilkan hasil dari pinValue kedalam serial monitor</p> <p>Digunakan untuk menulis nilai pinValue ke alamat EEPROM dengan alamat 21.</p> <p>Digunakan untuk mengkomit data yang telah ditulis ke EEPROM</p> <p>Membaca <i>input</i> dari Blynk pada pin V7</p> <p>Baris ini mengambil nilai parameter dari fungsi BLYNK_WRITE dan menyimpannya ke dalam variabel pinValue. Mengkonversi nilai parameter menjadi tipe data integer.</p> <p>Nilai doup disimpan kedalam pinValue</p> <p>Menampilkan hasil do upper kedalam serial monitor</p> <p>Menampilkan hasil pinValue kedalam serial monitor</p> <p>Nilai pinValue disimpan di alamat memori EEPROM. Nilai</p>

<i>Pseudo code</i>	<b>Fungsi</b>
<code>doup = pinValue;</code>	tersebut disimpan di alamat memori 22.
<code>Serial.print("do upper:");</code>	Digunakan untuk menyimpan perubahan yang dilakukan pada memori EEPROM.
<code>Serial.println(pinValue);</code>	Membaca input dari Blynk pada pin V8
<code>EEPROM.write(22,pinValue);</code>	Mengambil nilai parameter dari BLYNK_WRITE dan menyimpan ke dalam variabel pinValue. Mengkonversi nilai parameter menjadi tipe data integer.
<code>EEPROM.commit();</code>	Nilai dodown sama dengan nilai pinValue
<code>}</code>	Menampilkan hasil do lower kedalam serial monitor
<code>BLYNK_WRITE(V8){</code>	Menampilkan hasil dari pinValue kedalam serial monitor
<code>int pinValue = param.asInt();</code>	Nilai pinValue disimpan di alamat memori EEPROM Nilai tersebut disimpan di alamat memori 23.
<code>dodown = pinValue;</code>	Perintah ini digunakan untuk menyimpan perubahan yang dilakukan pada memori EEPROM.
<code>Serial.print("do lower:");</code>	Untuk memanggil dan meminta sensor suhu untuk membaca suhu sekarang.
<code>Serial.println(pinValue);</code>	Mengambil suhu yang terbaca dari sensor dan menyimpannya ke dalam variabel. Digunakan
<code>EEPROM.write(23,pinValue);</code>	

<i>Pseudo code</i>	<b>Fungsi</b>
EEPROM.commit(); }	untuk mendapatkan suhu dalam satuan Celsius dari sensor dengan indeks 0.
void tasktimer(){ sensors.requestTemperatures(); temperature = sensors.getTempCByIndex(0);	Baris ini membaca tegangan dari pin analog pin_ph. Nilai bacaan kemudian dikonversi menjadi tegangan nyata dalam satuan mV dan disimpan dalam variabel voltage.
 voltage = analogRead(pin_ph)/4095.0*3300.0;	Untuk memanggil fungsi read_ph untuk mendapatkan nilai pH Membaca nilai mentah dari pin dan disimpan kedalam ADC_Raw Untuk menghitung tegangan yang diukur dari nilai mentah.
 ph = read_ph(voltage, temperature);	Memanggil fungsi untuk mendapatkan nilai DO. Nilai DO dikembalikan dan dibagi dengan 1000 dan disimpan dalam variabel do_val
 ADC_Raw = analogRead(pin_do);	Melakukan pengecekan tipe data pH yang dikonversi ke integer kurang dari phdown atau lebih dari phup
 ADC_Voltage = uint32_t(VREF) * ADC_Raw / ADC_RES;	Memeriksa variabel lock1 tidak sama dengan 1
 do_val = (float)readDO(ADC_Voltage, temperature)/1000.0;	Memberikan nilai 1 ke variabel lock1
 if((int)ph<phdown  ((int)ph>phup) {	Mengatur pin menjadi nilai 0 atau LOW

<i>Pseudo code</i>	<b>Fungsi</b>
<pre> if(lock1!=1){ lock1=1; digitalWrite(pin_relph, 0); } } else{   if(lock1!=2){ lock1=2; digitalWrite(pin_relph, 1); } } if((int)do_val&gt;doup) { if(lock2!=1){ lock2=1; digitalWrite(pin_relkr, 1); } } if((int)do_val&lt;dodown){ </pre>	<p>Jika kondisi 1 tidak terpenuhi dan memeriksa apakah variabel lock1 tidak sama dengan 2</p> <p>Memasukkan nilai 2 ke variabel lock1</p> <p>Mengatur pin menjadi nilai 1 atau HIGH.</p> <p>Melakukan pengecekan do_val yang dikonversi menjadi integer lebih besar dari doup.</p> <p>Memeriksa variabel lock2 tidak sama dengan 1</p> <p>Memasukkan nilai 1 ke variabel lock2</p> <p>Mengatur pin menjadi nilai 1 atau HIGH</p> <p>Melakukan pengecekan apakah do_val yang dikonversi ke integer kurang dari dodown.</p> <p>Memeriksa apakah variabel lock2 tidak sama dengan 2</p> <p>Memasukkan nilai 2 ke variabel lock2</p> <p>Mengatur pin menjadi nilai 0 atau LOW</p> <p>Menampilkan LCD</p> <p>Kursor LCD pada baris 0 dan kolom 0</p> <p>Menampilkan tulisan PH</p> <p>Menampilkan hasil pH</p> <p>Set kursor pada baris 9 kolom 0</p>

<i>Pseudo code</i>	<b>Fungsi</b>
if(lock2!=2){	LCD menampilkan temperatur
	LCD menuliskan 0xdf
lock2=2;	LCD menampilkan hasil suhu
	Mengatur posisi kursor baris
digitalWrite(pin_relkcr, 0);	kedua indeks 1 dan kolom
}	pertama indeks 0
}	Menampilkan tulisan DO pada
lcd.clear();	LCD
lcd.setCursor(0,0);	Mencetak nilai dari do_val pada
	LCD
lcd.print("PH:");	Mengatur posisi kursor pada
lcd.print(ph);	baris kedua indeks 1 dan kolom
lcd.setCursor(9,0);	ke-9 pada LCD
	Mencetak tulisan P pada LCD
	Membaca nilai pin_relph dan
lcd.print(temperature);	menampilkan pada LCD
	Menampilkan tulisan K
lcd.write(0xdf);	Membaca nilai dari pin_relkcr
lcd.print("C");	dan menampilkan hasilnya pada
lcd.setCursor(0,1);	LCD
	Menampilkan tulisan pH pada
lcd.print("DO:");	serial monitor
	Mencetak variabel pH pada
lcd.print(do_val);	serial monitor
	Mencetak tulisan V CAL1 pada
lcd.setCursor(9,1);	serial monitor
	Mencetak nilai variabel
	ADC_Voltage pada serial
	monitor
lcd.print("P:");	Mencetak tulisan Do pada serial
lcd.print(digitalRead(pin_relph));	Mencetak nilai dari variabel
	do_val pada serial monitor
lcd.print(" K:");	Membuat baris baru pada serial
lcd.print(digitalRead(pin_relkcr));	monitor

<i>Pseudo code</i>	<b>Fungsi</b>
<code>Serial.print("Ph:");</code>	Mengirim data temperatur ke <i>widget</i> dengan ID V0 pada Blynk.
<code>Serial.print(ph);</code>	Mengirim data pH ke <i>widget</i> dengan ID V1 pada Blynk
<code>Serial.print(" V CAL1:");</code>	Mengirim data <i>do_val</i> ke <i>widget</i> dengan ID V2
<code>Serial.print(ADC_Voltage);</code>	Mengubah status pin led antara <i>HIGH</i> dan <i>LOW</i> . Jika pin led sebelumnya berstatus <i>HIGH</i> , maka akan diubah menjadi <i>LOW</i> dan sebaliknya.
<code>Serial.print(" Do:");</code>	Awalan untuk melakukan program
<code>Serial.print(do_val);</code>	Melakukan inisiasi EEPROM dengan kapasitas 512 <i>byte</i>
<code>Serial.println();</code>	Membaca nilai dari EEPROM 10 dan mengirim ke CAL1_V
<code>Blynk.virtualWrite(V0,temperature);</code>	Membaca nilai dari EEPROM 20 dan mengirim ke phup
<code>Blynk.virtualWrite(V1,ph);</code>	Membaca nilai dari EEPROM 21 dan mengirim ke phdown
<code>Blynk.virtualWrite(V2,do_val);</code>	Membaca nilai dari EEPROM 22 dan mengirim ke doup
<code>digitalWrite(led,digitalRead(led)^1);</code>	Membaca nilai dari EEPROM 23 dan mengirim ke dodown
<code>}</code>	Melakukan inisiasi komunikasi serial monitor 9600
<code>void setup()</code>	Mengatur mode pin_relph sebagai <i>output</i>
<code>{</code>	Mengatur mode pin_relkr sebagai <i>output</i>
<code>EEPROM.begin(512);</code>	Mengatur status pin_relph menjadi nilai 1 <i>HIGH</i>

<i>Pseudo code</i>	<b>Fungsi</b>
CAL1_V = eeprom_read(10);	Mengatur status pin_relkr menjadi nilai 1 <i>HIGH</i>
phup = EEPROM.read(20);	Mengatur mode pin led menjadi <i>output</i>
phdown = EEPROM.read(21);	Melakukan inisiasi layar LCD Menghidupkan layar belakang LCD
doup = EEPROM.read(22);	Melakukan pembersihan layar LCD Mengatur posisi kursor pada baris pertama 0, dan kolom pertama 0 pada LCD
dodown = EEPROM.read(23);	Menampilkan tulisan SSID Menampilkan nilai variabel SSID
Serial.begin(9600);	Mengatur posisi kursor pada baris kedua indeks baris 1 dan kolom pertama indeks 0 pada LCD
pinMode(pin_relph, OUTPUT);	Menampilkan tulisan pass Menampilkan nilai variabel pass
pinMode(pin_relkr, OUTPUT);	Menampilkan tulisan starting pada serial monitor
digitalWrite(pin_relph, 1);	Melakukan inisiasi pada Blynk dengan autentikasi auth, dan informasi WiFi.
digitalWrite(pin_relkr, 1);	Mengatur intrval waktu 1 detik menggunakan objek timer.
pinMode(led, OUTPUT);	Memanggil fungsi ph_init
lcd.begin();	Memanggil fungsi sensor
lcd.backlight();	Mencetak tulisan ready pada serial monitor Melakukan proses <i>looping</i>



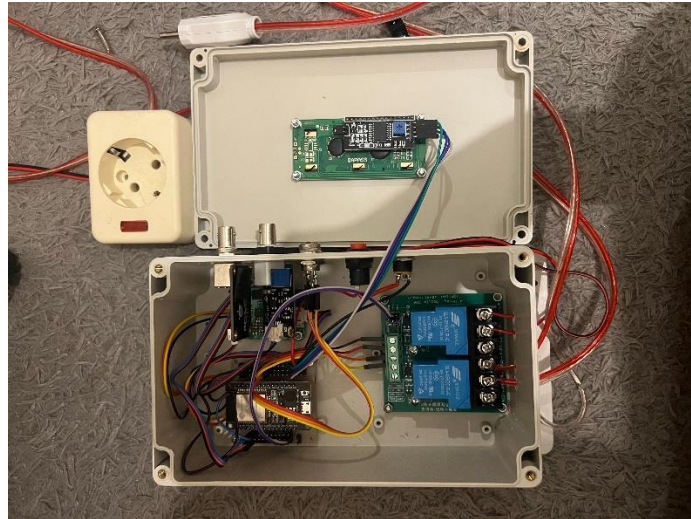
<i>Pseudo code</i>	<b>Fungsi</b>
lcd.clear();	Menjalankan Blynk
lcd.setCursor(0,0);	Menjalankan <i>timer</i>
lcd.print("ssid:");	Melakukan pemeriksaan apakah ada data yang tersedia di <i>input</i> serial. Jika ada data yang tersedia, blok kode di dalamnya akan dieksekusi.
lcd.print(ssid);	
lcd.setCursor(0,1);	Membaca data serial yang tersedia dan menyimpan dalam variabel <i>c</i>
lcd.print("pass:");	memeriksa apakah karakter yang dibaca adalah <i>a</i>
lcd.print(pass);	memanggil fungsi <i>cal_4</i> untuk melakukan kalibrasi dengan menggunakan nilai <i>voltage</i>
Serial.println("starting");	Memeriksa karakter yang dibaca adalah <i>b</i>
Blynk.begin(auth, ssid, pass);	Memanggil fungsi <i>cal_7</i> untuk melakukan kalibrasi.
timers.setInterval(1000L, tasktimer);	Memeriksa apakah karakter yang dibaca adalah <i>c</i>
ph_init();	Mengatur nilai <i>CAL1_V</i> pada proses pengembalian nilai kalibrasi pengukuran <i>DO</i>
sensors.begin();	
Serial.println("ready");	Memanggil fungsi <i>eeprom_write</i> untuk menulis nilai <i>CAL1_V</i>
}	
void loop()	
{	
Blynk.run();	Menampilkan nilai <i>CAL1_V</i> pada serial monitor
timers.run();	Mencetak tulisan <i>CAL DO OK</i> pada serial monitor
if(Serial.available()){	

<i>Pseudo code</i>	<b>Fungsi</b>
<pre> char c = Serial.read();  if(c=='a')  cal_4(voltage);  if(c=='b')  cal_7(voltage);  if(c=='c')  CAL1_V = ADC_Voltage;  eeprom_write(10, CAL1_V);  Serial.println(CAL1_V);  Serial.println("CAL DO OK"); }} </pre>	

#### 4.7.3 Hasil Pemasangan *Prototype*

Pemasangan instalasi dilakukan secara menyeluruh agar dapat dilakukan pengujian sistem sesuai dengan harapan pengguna. Berikut ini merupakan hasil pemasangan *prototype* tata letak sistem pengelolaan peternakan ayam otomotif berbasis mikrokontroler.

Berikut merupakan hasil *prototype* pada gambar 4.41:



Gambar 4. 41 Hasil *Prototype*

Setelah dilakukan pemasangan *prototype*, selanjutnya dilakukan uji *prototype* apakah pembacaan sensor sesuai dengan apa yang ada dengan *output* pada *Blynk Cloud*. Berikut merupakan hasil pembacaan alat yang ditampilkan oleh LCD pada mikrokontroler ESP-32.



Gambar 4. 42 Indikator Alat

Berikut merupakan hasil indikator yang diberikan oleh sensor suhu DS18B20, pH KB2004, dan sensor *dissolved oxygen* pada *Blynk Cloud*.



Gambar 4. 43 Indikator Blynk *Cloud*

Hasil pembacaan alat dan aplikasi Blynk terbukti sesuai pada uji coba alat diatas menunjukkan suhu 26,1°C, untuk pH sebesar 7,87 dan oksigen terlarut sebesar 1,5 mg/L. Uji coba diatas dilakukan secara bersamaan dan tidak terdapat *error* pada saat pengukuran.

## BAB V

### PENGUJIAN DAN PEMBAHASAN SISTEM

#### 5.1 Pengujian Sistem

Pengujian sistem bertujuan untuk melakukan evaluasi sistem *prototype* secara keseluruhan. Selain itu, pengujian dilaksanakan untuk memastikan bahwa sistem dapat menjalankan fungsi dengan benar dalam lingkup produksi pada peternakan ikan yang melibatkan pengujian fungsi dan non fungsi pada keseluruhan sistem. Pengamatan yang dilakukan dapat mengetahui apabila terjadi kesalahan pada sistem ataupun masalahlain yang belum terdeteksi.

##### 5.1.1 Uji Fungsionalitas

Pada penelitian ini dilaksanakan uji fungsionalitas. Uji fungsionalitas ini menggunakan parameter yang disusun berdasarkan fungsi pokok yang disesuaikan dengan kebutuhannya. Pengujian ini dilakukan langsung kepada *user* yaitu pemilik peternakan ikan nila.

Tabel 5. 1 UF-01

Uji Fungsionalitas				
Nama Sistem	Sistem sensor pH dan air pump			
Nomor Pengujian	UF-01			
No.	Fungsi Pokok	Sesuai		
		Ya	Tidak	
1.	Memantau pH didalam air	✓		
2.	Melihat hasil baca pH di LCD	✓		
3.	Melihat hasil baca pH di aplikasi	✓		
4.	Air <i>pump</i> hidup dibawah dan diatas batas normal yaitu 7	✓		
5.	Mematikan dan menghidupkan air <i>pump</i>	✓		
Jumlah		5	0	

Tabel diatas merupakan hasil dari pengujian sistem sensor pH dan air *pump* (cairan probiotik) yang sepenuhnya dapat diakses melalui Blynk Cloud. Uji fungsionalitas yang telah dilaksanakan berdasarkan fungsi pokok diatas memiliki hasil sempurna atau 100% sesuai dengan harapan pengguna atau pada penelitian ini adalah pemilik peternakan nila.

Tabel 5. 2 UF-02

Uji Fungsionalitas			
Nama Sistem	Sistem sensor <i>dissolved</i> oksigen dan kincir		
Nomor Pengujian	UF-02		
No.	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memantau DO didalam air	✓	
2.	Melihat hasil baca DO di LCD	✓	
3.	Melihat hasil baca DO di aplikasi	✓	
4.	Kincir hidup dibawah dan diatas batas normal DO	✓	
5.	Mematikan dan menghidupkan kincir oksigen	✓	
Jumlah		5	0

Tabel diatas merupakan hasil dari pengujian sistem sensor DO dan kincir oksigen yang sepenuhnya dapat diakses melalui Blynk Cloud. Uji fungsionalitas yang telah dilaksanakan berdasarkan fungsi pokok diatas memiliki hasil sempurna atau 100% sesuai dengan harapan pengguna atau pada penelitian ini adalah pemilik peternakan nila.

Tabel 5. 3 UF-03

Uji Fungsionalitas			
Nama Sistem	Sistem sensor suhu		
Nomor Pengujian	UF-03		
No.	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memantau suhu dan kelembaban pada air	✓	
2.	Melihat hasil suhu di LCD	✓	
3.	Melihat hasil baca suhu di aplikasi	✓	

Uji Fungsionalitas					
Nama Sistem	Sistem sensor suhu				
Nomor Pengujian	UF-03				
No.	Fungsi Pokok			Sesuai	
				Ya	Tidak
4.	Mengakses	sistem	melalui	✓	
	<i>smartphone</i>				
Jumlah				4	0

Tabel diatas merupakan hasil dari pengujian sistem sensor suhu yang sepenuhnya dapat diakses melalui Blynk Cloud. Uji Fungsionalitas yang telah dilaksanakan berdasarkan fungsi pokok diatas memiliki hasil sempurna atau 100% sesuai dengan harapan pengguna atau pada penelitian ini adalah pemilik peternakan nila.

## 5.2 Pembahasan Sistem

### 5.2.1 Sistem sensor pH dan air *pump*

Pada sistem sensor pH yang telah terhubung dengan mikrokontroler dilakukan kalibrasi sensor menggunakan cairan pH sebagai tahapan awal untuk mengukur pH baik keasaman maupun kebasahan air. Pada bagian diatas mendapatkan hasil dari pengujian sensor berjalan dengan baik sesuai dengan kemauan *user* dan sistem perbaikan yang telah dilakukan. Air *pump* disini berguna untuk mengalirkan cairan probiotik didalam kolam, cairan ini berguna untuk membuat pH tetap dalam kondisi normal.

Berikut merupakan hasil baca sensor yang telah dilakukan:

Tabel 5. 4 PT-01

<i>Performance Test</i>				
Nama sistem	Sistem sensor pH dan <i>air pump</i>			
Nomor pengujian	PT-01			
No	pH	Indikator LCD dan Blynk	<i>Air Pump</i>	
			Nyala	Tidak
1	4	Nyala	✓	
2	5	Nyala	✓	
3	6	Nyala	✓	

<i>Performance Test</i>				
Nama sistem	Sistem sensor pH dan <i>air pump</i>			
Nomor pengujian	PT-01			
No	pH	Indikator LCD dan Blynk	<i>Air Pump</i>	
			Nyala	Tidak
4	7	Nyala		✓
5	8	Nyala		✓
6	9	Nyala	✓	
7	10	Nyala	✓	

Pada pembacaan sistem diatas maka didapatkan hasil untuk menghidupkan *air pump* untuk penyemprotan probiotik maka kondisi pH dalam air harus kurang dari 6,5 hingga lebih dari 8,5 sesuai dengan kondisi pH normal untuk lingkungan hidup ikan nila. Hasil dari pembacaan pH juga dapat dilihat langsung dari LCD monitor yang terdapat pada rangkaian mikrokontroler dan juga dapat dilihat menggunakan Blynk *Cloud*.

### 5.2.2 Sistem sensor *dissolved oxygen* dan kincir

Sistem sensor DO atau *dissolved oxygen* yang telah terhubung dengan mikrokontroler dilakukan kalibrasi sensor menggunakan suhu air kolam sebagai tahapan awal untuk mengukur oksigen. Pada bagian uji fungsionalitas diatas mendapatkan hasil sempurna dan telah sesuai dengan kemauan *user* dan untuk kincir berguna untuk mnestabilkan oksigen terlarut dalam kolam.

Berikut merupakan hasil baca sensor yang telah dilakukan:

Tabel 5. 5 PT-02

<i>Performance Test</i>				
Nama sistem	Sistem <i>dissolved oxygen</i> dan kincir			
Nomor pengujian	PT-02			
No	DO	Indikator LCD dan Blynk	Kincir	
			Nyala	Tidak
1	1 mg/L	Nyala	✓	
2	2 mg/L	Nyala	✓	
3	3 mg/L	Nyala	✓	



<i>Performance Test</i>				
Nama sistem	Sistem <i>dissolved oxygen</i> dan kincir			
Nomor pengujian	PT-02			
No	DO	Indikator LCD dan Blynk	Kincir	
			Nyala	Tidak
4	4 mg/L	Nyala		✓
5	5 mg/L	Nyala		✓
6	6 mg/L	Nyala		✓
7	7 mg/L	Nyala		✓

Pada pembacaan sistem diatas maka didapatkan hasil untuk menghidupkan kincir oksigen diperlukan jumlah oksigen terlarut kurang dari 3 mg/L jika lebih dari 3 mg/L maka kincir tidak akan hidup atau terindikasi oksigen terlarut dalam kolam memiliki status normal. Hasil dari pembacaan DO juga dapat dilihat di LCD pada rangkaian mikrokontroler atau pada Blynk *Cloud*.

### 5.2.3 Sistem sensor suhu

Sistem sensor suhu atau DS18B20 yang telah terhubung dengan mikrokontroler dilakukan kalibrasi sensor menggunakan suhu air kolam sebagai tahapan awal untuk mengukur suhu. Pada bagian uji fungsionalitas diatas mendapatkan hasil sempurna dan telah sesuai dengan kemauan *user*.

Tabel 5. 6 PT-03

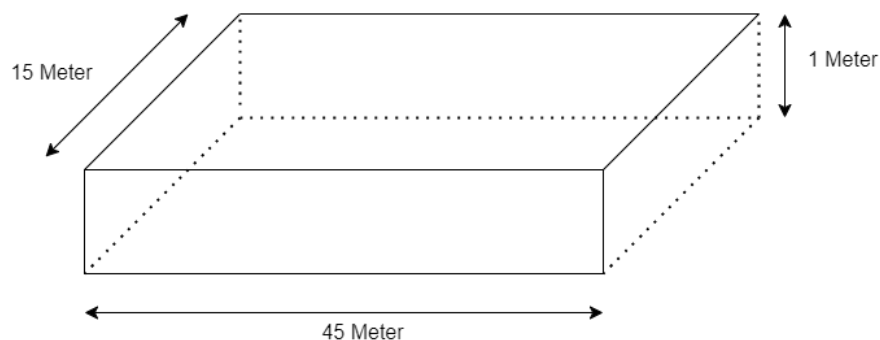
<i>Performance Test</i>				
Nama sistem	Sistem sensor suhu			
Nomor pengujian	PT-03			
No	Suhu	Indikator LCD dan Blynk	Kincir	
			Nyala	Tidak
1	28	Nyala		✓
2	29	Nyala		✓
3	30	Nyala		✓
4	31	Nyala		✓
5	32	Nyala	✓	

<i>Performance Test</i>				
Nama sistem	Sistem sensor suhu			
Nomor pengujian	PT-03			
No	Suhu	Indikator LCD dan Blynk	Kincir	
			Nyala	Tidak
6	33	Nyala	✓	
7	34	Nyala	✓	

Pada pembacaan sistem diatas maka didapatkan hasil untuk suhu yang baik menurut ikan nila dan hasil yang didapatkan pada saat menggunakan mikrokontroler yaitu pada suhu 24 derajat celcius sampai 31 derajat celcius adalah batas normal suhu pada kolam ikan. Hasil dari pembacaan suhu juga dapat dilihat di LCD pada rangkaian mikrokontroler atau pada Blynk *Cloud*.

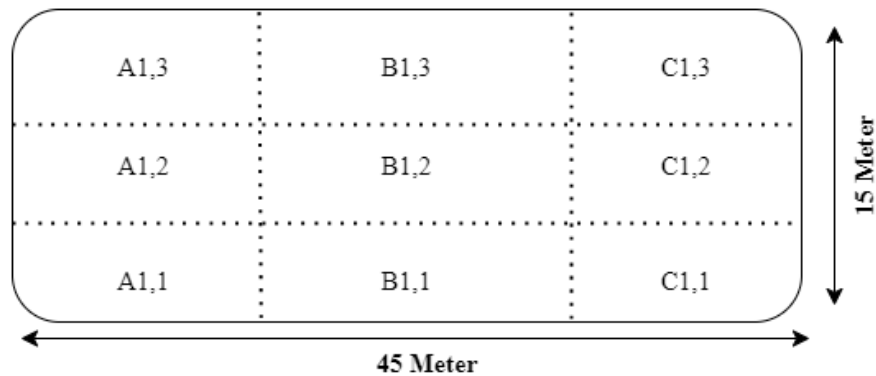
### 5.3 Tata Letak Sensor

Penentuan peletakan sensor sistem automasi pengelolaan kontrol air pada peternakan ikan nila harus dilakukan untuk mengetahui lokasi kolam yang sangat membutuhkan untuk menggunakan alat. Peneliti melakukan uji sampel air yang ada dikolam ikan nila Minaverse Company pada tanggal 6 Juli 2023.



Gambar 5. 1 Ukuran Kolam Ikan

Uji sampel air dilakukan untuk menentukan jumlah pH air, suhu air, dan oksigen pada kolam apakah memiliki perbedaan atau tidak satu dengan yang lainnya tergantung lokasi pengambilan air. Untuk ukuran kolam memiliki panjang yaitu 45 meter dan lebar 15 meter dengan kedalaman kolam 1 meter. Peneliti membuat skema pengambilan air dengan membagi 3 lokasi yaitu A sebagai kolam depan, B kolam tengah, dan C kolam bagian belakang. Skema pengambilan air seperti gambar dibawah ini:



Gambar 5. 2 Skematik Pengambilan Air di Permukaan

Uji sampel air diambil dengan jarak 15 meter panjang dan 5 meter lebar kolam dan dipatkan sebanyak 9 sampel air untuk permukaan kolam. Hasil yang di dapatkan uji sampel air seperti tabel dibawah ini:

Tabel 5. 7 Hasi Uji Air Permukaan

No	Kode Air	pH	DO	Suhu
1	A1,1	7,62	0,90	28,5
2	A1,2	7,67	0,91	28,5
3	A1,3	7,84	0,91	28,5
4	B1,1	8,18	1,54	28,5
5	B1,2	7,80	1,17	28,3
6	B1,3	7,86	1,50	28,3
7	C1,1	7,94	1,54	28,3
8	C1,2	7,86	1,74	28,3
9	C1,3	7,91	1,70	28,1

Hasil yang didapat dari 9 sampel air pada permukaan kolam adalah tidak memiliki banyak perbedaan mengenai pH air, suhu, dan oksigen terlarut dalam kolam. Nilai tertinggi didapat oleh kode air B1,1 dengan nilai pH sebesar 8,18, nilai *dissolved oxygen* atau oksigen terlarut sebesar 1,54 mg/L dan suhu sebesar 28,5°C.

Setelah dilakukan uji permukaan air, selanjutnya dilakukan uji kedalaman air. Kolam yang ada di Minaverse Company memiliki kedalaman 1 meter. Peneliti mengambil air dengan menggunakan pipet dan selang untuk mengambil perbedaan air dengan tujuan apakah air di

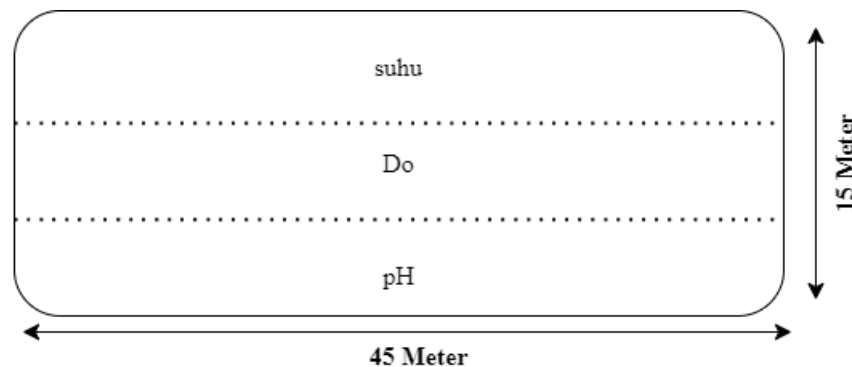
permukaan kolam sama nilainya dengan air yang didalam kolam. Hasil yang didapat pada saat melakukan uji sampel air untuk kedalaman 1 meter adalah sebagai berikut:

Tabel 5. 8 Hasil Uji Air Kedalaman

No	Kode Air	pH	DO	Suhu
1	A1,1	7,62	0,80	28,2
2	A1,2	7,67	0,90	28,3
3	A1,3	7,84	0,91	28,2
4	B1,1	8,18	1,53	28,2
5	B1,2	7,80	1,17	28,2
6	B1,3	7,86	1,50	28,1
7	C1,1	7,94	1,53	28,1
8	C1,2	7,86	1,73	28,2
9	C1,3	7,91	1,70	28,0

Hasil yang didapat dari 9 sampel air pada kedalaman kolam adalah tidak memiliki banyak perbedaan mengenai pH air, suhu, dan oksigen terlarut dalam kolam. Nilai tertinggi didapat oleh kode air B1,1 dengan nilai pH sebesar 8,18, nilai *dissolved oxygen* atau oksigen terlarut sebesar 1,53 mg/L dan suhu sebesar 28,2°C.

Pengujian kinerja sensor yang dilakukan oleh (Bhawiyuga & Yahya, 2019) diperoleh hasil terbaik pada jarak 40 meter untuk sensor yang digunakan yaitu sensor pH KB2004 , sensor suhu DS18B20, dan sensor analog *dissolved oxygen*. Hasil uji dapat diterapkan pada penelitian kali ini untuk kolam ikan dengan panjang 45 meter dan lebar 15 meter seperti gambar dibawah ini:



Gambar 5. 3 Skema peletakan sensor

Kesimpulan yang di dapat adalah tidak ada perbedaan yang signifikan mengenai hasil uji air di permukaan kolam ataupun di kedalaman kolam. Perubahan yang terjadi adalah jika pada kedalaman kolam suhu akan menjadi lebih dingin dibandingkan dengan di permukaan kolam dan kadar oksigen terlarut akan lebih sedikit dibandingkan dengan kadar oksigen terlarut di permukaan kolam. Tetapi pada pH hasil masih tetap sama baik di permukaan maupun di kedalaman kolam. Lokasi terbaik untuk menempatkan hasil *prototype* alat adalah di B1,1.

Menurut (Alfian Pratama et al., 2021) Kualitas lingkungan hidup ikan sangat berpengaruh terhadap peningkatan ikan dan hasil panen ikan. 3 faktor utama yaitu suhu berkisar 28-30°C, pH berkisar 7-8, dan DO berkisar 6-8,5 ppm, jika ketiga faktor tersebut diambang batas normal dan dapat terkontrol secara menyeluruh maka akan meningkatkan hasil panen ikan nila.

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Setelah dilakukan pembangunan serta pengujian sistem, didapatkan kesimpulan dari penelitian ini sebagai berikut:

1. Untuk melakukan desain sistem otomatisasi untuk pendeteksi oksigen terlarut, pH dan suhu air dilakukan pembuatan *prototype* yang diawali dengan *requirements gathering* untuk menentukan kebutuhan sistem yang akan dilakukan, selanjutnya melakukan *quick design* yaitu pembuatan desain awal dan penentuan sensor yang digunakan untuk suhu menggunakan sensor DS18B20, pH menggunakan sensor KB2004 dan untuk *dissolved oxygen* menggunakan sensor analog *dissolved oxygen*. Langkah selanjutnya adalah *build prototype* menggunakan mikrokontroler ESP-32 untuk memberikan *input, proses*, dan *output*. *Evaluation and Refining* dilakukan *testing* kepada pemilik peternakan ikan nila dan mendapatkan saran tentang *prototype* yang telah dibuat. Langkah terakhir adalah pembuatan produk akhir yang telah di evaluasi sesuai dengan kebutuhan dan keinginan pengguna atau pemilik peternakan ikan nila. Hasil dari *prototype* dapat meningkatkan hasil panen jika sesuai dengan batasan normal 3 faktor utama yaitu suhu berkisar 28-30°C, pH berkisar 7-8, dan DO berkisar 6-8,5 ppm.
2. Untuk melakukan pengendalian kadar oksigen terlarut, pH dan suhu air menggunakan 3 sensor utama yang telah terhubung dengan mikrokontroler ESP-32. Untuk oksigen terlarut menggunakan sensor analog *dissolved oxygen*, suhu air menggunakan DS18B20, dan pH menggunakan KB2004. Sistem *controlling* dan *monitoring* untuk ketiga sensor tersebut menggunakan aplikasi Blynk *Cloud* yang telah di program menggunakan mikrokontroler ESP-32 melalui Arduino IDE. Pengguna dapat melakukan pemantauan langsung dari Blynk *Cloud* yang telah dibuat.

## 6.2 Saran

Berikut merupakan saran peneliti agar menjadi pertimbangan dan pandangan yang dapat dikembangkan pada penelitian selanjutnya:

1. Menambahkan beberapa faktor penting untuk menjaga kualitas dari pengelolaan sistem air pada ikan nila, tidak hanya suhu, pH dan oksigen.
2. Melakukan sistem automasi secara penuh meliputi pemberian pakan ikan dan vitamin pada ikan.
3. Melakukan kontrol pada sirkulasi air secara menyeluruh agar dapat menghasilkan ikan yang maksimal.

### DAFTAR PUSTAKA

- Agustia, R. D., & Kiki H, T. A. (2018). Pembangunan Prototype Aplikasi Pengawasan Dan Pengendalian Pembudidayaan Mikroalga Spirulina. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, 7(1), 11–18. <https://doi.org/10.34010/komputa.v7i1.2531>
- Alfian Pratama, M., Arthana Wayan, I., & Raka A.K, G. (2021). Fluktuasi Kualitas Air Budidaya Ikan Nila (*Oreochromis niloticus*) dengan Beberapa Variasi Sistem Resirkulasi. *Current Trends in Aquatic Science IV*, 107(1), 102–107.
- Araneta, A. A. S. (2022). Design of an Arduino-Based Water Quality Monitoring System. *International Journal of Computer Science and Mobile Computing*, 11(3), 152–165. <https://doi.org/10.47760/ijcsmc.2022.v11i03.017>
- Azhari, D., & Tomaso, A. M. (2018). Kajian Kualitas Air dan Pertumbuhan Ikan Nila (*Oreochromis niloticus*) yang Dibudidayakan dengan Sistem Akuaponik. *Akuatika Indonesia*, 3(2), 84. <https://doi.org/10.24198/jaki.v3i2.23392>
- Aziz, F. A., Sarosa, M., & Rohadi, E. (2020). Monitoring system water pH rate, turbidity, and temperature of river water. *IOP Conference Series: Materials Science and Engineering*, 732(1). <https://doi.org/10.1088/1757-899X/732/1/012106>
- Barus, E. E., Pingak, R. K., & Louk, A. C. (2018). OTOMATISASI SISTEM KONTROL pH DAN INFORMASI SUHU PADA AKUARIUM MENGGUNAKAN ARDUINO UNO DAN RASPBERRY PI 3. *Jurnal Fisika : Fisika Sains Dan Aplikasinya*, 3(2), 117–125. <https://doi.org/10.35508/fisa.v3i2.612>
- Bhawiyuga, A., & Yahya, W. (2019). Sistem Monitoring Kualitas Air Kolam Budidaya Menggunakan Jaringan Sensor Nirkabel Berbasis Protokol LoRa. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(1), 99. <https://doi.org/10.25126/jtiik.2019611292>
- Efendi, E. (2015). Efektifitas Sistem Akuaponik Dalam Mereduksi Konsentrasi. *E-Journal Rekayasa Dan Teknologi Budidaya Perairan*, 3(April), 2–5.
- Efendi, Y. (2018). Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile. *Jurnal Ilmiah Ilmu Komputer*, 4(1), 19–26. <https://doi.org/10.35329/jiik.v4i1.48>
- Effendi, H. (2003). *Telaah kualitas air bagi pengelolaan sumberdaya dan lingkungan perairan*.



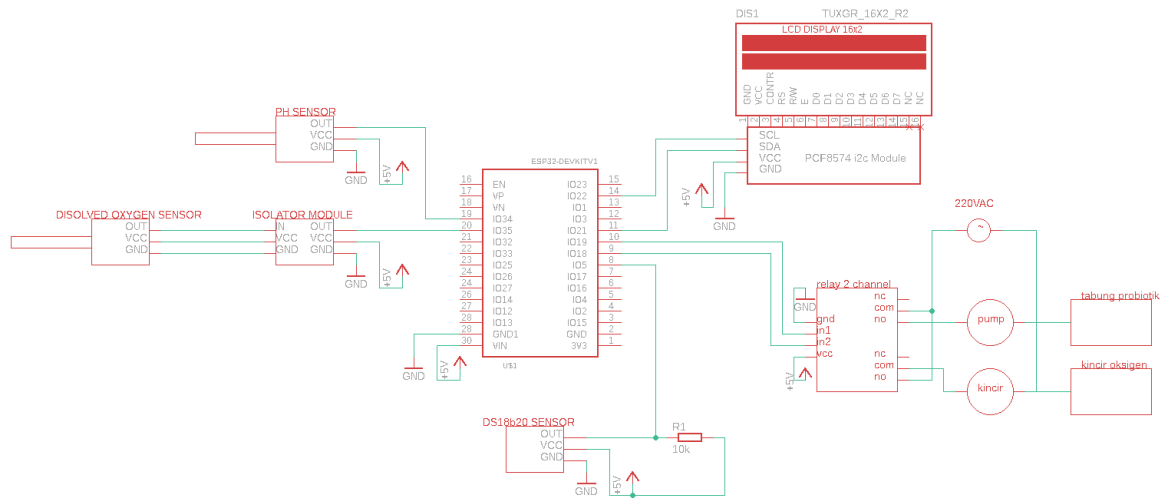
- Endut, N. A., Fahmi, M., Fo, M., Azylia, N., Azam, A., Abu, N. A., Rahayu, S., Aziz, A., Shobirin, A., & Sani, A. (2019). Real-Time Water Monitoring System for Fish Farmers Using Arduino. *Journal of Advanced Research in Computing and Applications*, 14(1), 10–17.
- Gokulanathan, S., Manivasagam, P., Prabu, N., & Venkatesh, T. (2019). GSM Based Water Quality Monitoring System Using Arduino. *Shanlax International Journal of Arts, Science and Humanities*, 6(4), 22–26. <https://doi.org/10.34293/sijash.v6i4.341>
- Imran, A., & Rasul, M. (2020). Pengembangan Tempat Sampah Pintar Menggunakan Esp32. *Jurnal Media Elektrik*, 17(2), 2721–9100. <https://ojs.unm.ac.id/mediaelektrik/article/view/14193>
- Ismail, R., Shafinah, K., & Latif, K. (2020). A Proposed Model of Fishpond Water Quality Measurement and Monitoring System based on Internet of Things (IoT). *IOP Conference Series: Earth and Environmental Science*, 494(1). <https://doi.org/10.1088/1755-1315/494/1/012016>
- Khairuman, S. P. H., Amri, K., & Spi, M. (2013). *Budi Daya Ikan Nila*. books.google.com. [https://books.google.com/books?hl=en&lr=&id=rcMQAgAAQBAJ&oi=fnd&pg=PA1&dq=khairuman+nila&ots=fwB\\_Fa8zS7&sig=iteixuuUFMC4CxpXYZtvv0p-8Tk](https://books.google.com/books?hl=en&lr=&id=rcMQAgAAQBAJ&oi=fnd&pg=PA1&dq=khairuman+nila&ots=fwB_Fa8zS7&sig=iteixuuUFMC4CxpXYZtvv0p-8Tk)
- Kustija, J., & Andika, F. (2021). Control - Monitoring System Of Oxygen Level, Ph, Temperature And Feeding in Pond Based on Iot. *REKA ELKOMIKA: Jurnal Pengabdian Kepada Masyarakat*, 2(1), 1–10. <https://doi.org/10.26760/rekaelkomika.v2i1.1-10>
- Manoj, M., Kumar, V. D., Arif, M., Bulai, E. R., Bulai, P., & Geman, O. (2022). State of the Art Techniques for Water Quality Monitoring Systems for Fish Ponds Using IoT and Underwater Sensors: A Review. *Sensors*, 22(6). <https://doi.org/10.3390/s22062088>
- Marlina, E., Peternakan, J., Studi Budidaya Perikanan Politeknik Negeri Lampung JI Soekarno-Hatta Rajabasa no, P., & Lampung, B. (2016). *Prosiding Seminar Nasional Tahunan Ke-V Hasil-Hasil Penelitian Perikanan dan Kelautan KAJIAN KANDUNGAN AMMONIA PADA BUDIDAYA IKAN NILA (Oreochromis niloticus) MENGGUNAKAN TEKNOLOGI AKUAPONIK TANAMAN TOMAT (Solanum lycopersicum)*. 181–187.
- Mas'ud, F. (2014). Pengaruh kualitas air terhadap pertumbuhan ikan nila (*Oreochromis sp.*) di kolam beton dan terpal. In *Grouper Faperik*.
- Md. Mahbubur Rahman, Chinmay Bapery, Mohammad Jamal Hossain, Zahid Hassan, G.M.

- Jamil Hossain, M. M. I. (2020). Internet of Things (IoT) Based Water Quality Monitoring System 1, Md. *International Journal of Multidisciplinary and Current Educational Research (IJM CER)*, 2(4), 168–180.
- Muzawi, R., Efendi, Y., & Agustin, W. (2018). Sistem Pengendalian Lampu Berbasis Web dan Mobile. *SATIN - Sains Dan Teknologi Informasi*, 4(1), 29–35. <https://doi.org/10.33372/stn.v4i1.292>
- Ogedebe, P. M., & Jacob, B. P. (2012). Software prototyping: a strategy to use when user lacks data processing experience. In *ARNP Journal of Systems and Software*.
- Oktafiadi, R. (2016). Sistem pemantau kekeruhan air dan pemberi makan otomatis pada ikan berbasis mikrokontroler. In *Jurnal Ilmu Teknik Elektro Komputer dan Informatika ... core.ac.uk*. <https://core.ac.uk/download/pdf/295348224.pdf>
- Pasika, S., & Gandla, S. T. (2020). Smart water quality monitoring system with cost-effective using IoT. *Heliyon*, 6(7), e04096. <https://doi.org/10.1016/j.heliyon.2020.e04096>
- Purnomo, D. (2017). Model prototyping pada pengembangan sistem informasi. *JIMP (Jurnal Informatika Merdeka Pasuruan)*. <http://ejurnal.unmerpas.ac.id/index.php/informatika/article/view/67>
- Ratannanda, R. (2011). Penentuan waktu retensi sistem akuaponik untuk mereduksi limbah budidaya ikan nila. In *Institut Pertanian Bogor*.
- Salsabila, M. (2018). *Teknik pembesaran ikan nila (Oreochromis niloticus) di instalasi budidaya air tawar pandaan, Jawa Timur*. repository.unair.ac.id. <https://repository.unair.ac.id/79083/>
- Sampebatu, L., & Kamolan, A. (2021). Rancang Bangun Sistem Otomatisasi dan Telekontrol Pada Pengkondisian Air Kolam Pembibitan Ikan Nila. *Jurnal Ampere*, 6(1), 32. <https://doi.org/10.31851/ampere.v6i1.5981>
- Saputra, Y. A., Latifa, U., Juusan, M., Elektro, T., Teknik, F., Karawang, U. S., Jurusan, D., Elektro, T., Teknik, F., Karawang, U. S., & Pendahuluan, I. (2022). *Perancangan sistem kendali pada budidaya ikan nila berbasis mikrokontroler dan teknologi data logger 1*. 11(2), 224–236.
- Sholikin, N., Rozaq, I. A., Iqbal, M., & Setyaningsih, N. Y. D. (2021). Kontrol Kadar Ph Dan Ketinggian Air Pada Kolam Ikan Nila Berbasis IOT. *Jurnal ELKON*, 01(01), 5–8.
- Siegers, W. H., Prayitno, Y., & Sari, A. (2019). Pengaruh kualitas air terhadap pertumbuhan

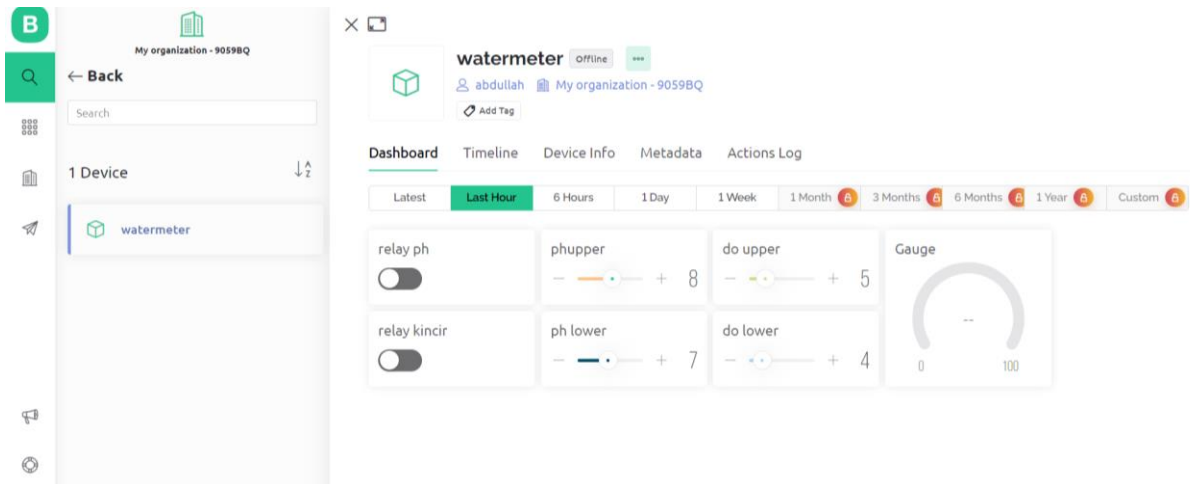
- ikan nila nirwana (*Oreochromis sp.*) pada tambak payau. *The Journal of Fisheries* ....  
<http://jurnal.uniyap.ac.id/jurnal/index.php/Perikanan/article/view/503>
- Sugiyono, D. (2018). Metode penelitian kuantitatif, kualitatif dan R & D/Sugiyono. In *Bandung: Alfabeta*.
- Supandi, F., Desta P, W., Ambar S, Y., & Sudir, M. (2019). Analisis Resiko Pada Pengembangan Perangkat Lunak Yang Menggunakan Metode Waterfall Dan Prototyping. *Prosiding Seminar Nasional Dinamika Informatika 2018 (SENADI 2018)*, 2(1), 83–86. <http://prosiding.senadi.upy.ac.id/index.php/senadi/article/view/86>
- Suriana, S., Lubis, A. P., & Rahayu, E. (2021). Sistem Monitoring Jarak Jauh Pada Suhu Kolam Ikan Nila Bangkok Memanfaatkan Internet of Things (IOT) Berbasis NODEMCUESP8266. *JUTSI (Jurnal Teknologi Dan Sistem Informasi)*, 1(1), 1–8. <https://doi.org/10.33330/jutsi.v1i1.1004>
- Susanti, N. D., Sagita, D., Apriyanto, I. F., Anggara, C. E. W., Darmajana, D. A., & Rahayuningtyas, A. (2022). Design and Implementation of Water Quality Monitoring System (Temperature, pH, TDS) in Aquaculture Using IoT at Low Cost. *Proceedings of the 6th International Conference of Food, Agriculture, and Natural Resource (IC-FANRES 2021)*, 16, 7–11. <https://doi.org/10.2991/absr.k.220101.002>
- Tebbut, T. H. Y. (1998). *The Principle of Water Quality Control*. Butter Worth. Heineman, Linacre House Jordon ....
- Widodo, A., Alfia, R., & Kholis, N. (2021). Sistem Monitoring Kualitas Air Pada Sistem Akuaponik Berbasis Iot. *Jurnal Teknik Elektro*. <https://ejournal.unesa.ac.id/index.php/JTE/article/view/42748>
- Widodo, T., Irawan, B., Prastowo, A. T., & Surahman, A. (2020). Sistem Sirkulasi Air Pada Teknik Budidaya Bioflok Menggunakan Mikrokontroler Arduino UNO R3. *Jurnal Teknik Dan Sistem Komputer*, 1(2), 1–6. <https://doi.org/10.33365/jtikom.v1i2.12>

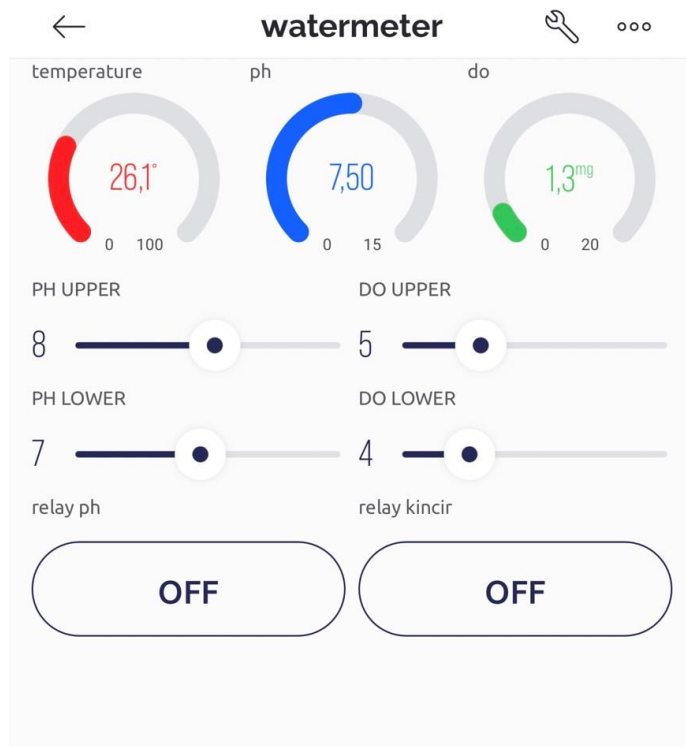
## LAMPIRAN

### A-Hasil Perancangan Skematik



### B-Hasil Tampilan Blynk Cloud





**C-Hasil Pembacaan LCD**



**D-Pengambilan Data Lokasi Peletakan Sensor**





## E-Pengkodean Arduino IDE

```

1
2 #define BLYNK_TEMPLATE_ID "TMPL6p_Ml0w0e"
3 #define BLYNK_TEMPLATE_NAME "watermeter"
4 #define BLYNK_AUTH_TOKEN "g4bu4ysoW7L2U2AqqqTURJ4Tts19UXZ"
5
6 #define BLYNK_PRINT Serial // tampilkan informasi blynk di serial
7 #include <BlynkSimpleEsp32.h> // library blynk
8 // input wifi dan password
9 char auth[] = BLYNK_AUTH_TOKEN;
10 char ssid[] = "mywifi"; // type your wifi name
11 char pass[] = "354354354"; // type your wifi password
12 BlynkTimer timers;
13
14 #include "ph_lib.h"
15 #include "eeprom_lib.h"
16
17 #define led 2
18 #define pin_ph 34
19 #define pin_do 35
20 #define pin_ds 5
21 #define pin_relph 19
22 #define pin_relker 18
23
24 #include <Wire.h>
25 #include <LiquidCrystal_I2C.h>
26 LiquidCrystal_I2C lcd(0x27, 16, 2);
27
28 float ph, voltage, do_val;
29
30 #include <OneWire.h>
31 #include <DallasTemperature.h>
32 OneWire oneWire(pin_ds); // sensor suhu terhubung di D3
33 DallasTemperature sensors(&oneWire);
34
35 #include <EEPROM.h>
36 int phup, phdown;

```



```

27 int lock1,lock2;
28
29
30 float temperature;
31
32 #define VREF 3300 //VREF (mv)
33 #define ADC_RES 4095 //ADC Resolution
34
35 //Single-point calibration Mode=0
36 //Two-point calibration Mode=1
37 #define TWO_POINT_CALIBRATION 0
38
39 int CAL1_V;
40 //Single-point calibration needs to be filled CAL1_V and CAL1_T
41 //#define CAL1_V (2048) //mv
42 #define CAL1_T (25) //°C
43 //Two-point calibration needs to be filled CAL2_V and CAL2_T
44 //CAL1 High temperature point, CAL2 low temperature point
45 #define CAL2_V (1300) //mv
46 #define CAL2_T (15) //°C
47
48 const uint16_t DO_Table[41] = {
49     14460, 14220, 13820, 13440, 13090, 12740, 12420, 12110, 11810, 11530,
50     11260, 11010, 10770, 10530, 10300, 10080, 9860, 9660, 9460, 9270,
51     9080, 8900, 8730, 8570, 8410, 8250, 8110, 7960, 7820, 7690,
52     7560, 7430, 7300, 7180, 7070, 6950, 6840, 6730, 6630, 6530, 6410};
53
54 uint8_t Temperature;
55 uint16_t ADC_Raw;
56 uint16_t ADC_Voltage;
57 uint16_t DO;
58
59 int16_t readDO(uint32_t voltage_mv, uint8_t temperature_c)
60 {
61     #if TWO_POINT_CALIBRATION == 0
62         uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35;

```

```

71     #if TWO_POINT_CALIBRATION == 0
72         uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35;
73         return (voltage_mv * DO_Table[temperature_c] / V_saturation);
74     #else
75         uint16_t V_saturation = (uint16_t)((int8_t)temperature_c - CAL2_T) * ((uint16_t)CAL1_V - CAL2_V) / ((uint8_t)CAL1_T - CAL2_T) + CAL2_V;
76         return (voltage_mv * DO_Table[temperature_c] / V_saturation);
77     #endif
78 }
79
80 // membaca input dari blynk
81 BLYNK_WRITE(V3){
82     int pinValue = param.asInt();
83     if(pinValue==1) digitalWrite(pin_relph, 0);
84     if(pinValue==0) digitalWrite(pin_relph, 1);
85     Serial.print("R PH:");
86     Serial.println(pinValue);
87 }
88
89 // membaca input dari blynk
90 BLYNK_WRITE(V4){
91     int pinValue = param.asInt();
92     if(pinValue==1) digitalWrite(pin_relkr, 0);
93     if(pinValue==0) digitalWrite(pin_relkr, 1);
94     Serial.print("R KIRCR:");
95     Serial.println(pinValue);
96 }
97
98 // membaca input dari blynk
99 BLYNK_WRITE(V5){
100     int pinValue = param.asInt();
101     phup = pinValue;
102     Serial.print("ph upper:");
103     Serial.println(pinValue);
104     EEPROM.write(20,pinValue);
105     EEPROM.commit();

```

```

107 }
108 // membaca input dari blynk
109 BLYNK_WRITE(V6){
110     int pinValue = param.asInt();
111     phdown = pinValue;
112     Serial.print("ph lower:");
113     Serial.println(pinValue);
114     EEPROM.write(21,pinValue);
115     EEPROM.commit();
116 }
117
118 // membaca input dari blynk
119 BLYNK_WRITE(V7){
120     int pinValue = param.asInt();
121     dop = pinValue;
122     Serial.print("do upper:");
123     Serial.println(pinValue);
124     EEPROM.write(22,pinValue);
125     EEPROM.commit();
126 }
127
128 // membaca input dari blynk
129 BLYNK_WRITE(V8){
130     int pinValue = param.asInt();
131     dodown = pinValue;
132     Serial.print("do lower:");
133     Serial.println(pinValue);
134     EEPROM.write(23,pinValue);
135     EEPROM.commit();
136 }
137
138
139 void taskimer(){
140     sensors.requestTemperatures();
141     temperature = sensors.getTempCByIndex(0);

```



```

144 voltage = analogRead(pin_ph)/4095.0*3300.0;
145 ph = read_ph(voltage, temperature);
146
147 ADC_Raw = analogRead(pin_do);
148 ADC_Voltage = uint32_t(VREF) * ADC_Raw / ADC_RES;
149 do_val = (float)readDO(ADC_Voltage, temperature)/1000.0;
150
151
152
153
154 if(((int)ph<phdown||((int)ph>phup) {
155     if(lock1=1){
156         lock1=1;
157         digitalWrite(pin_relph, 0);
158     }
159 }
160 else{
161     if(lock1=2){
162         lock1=2;
163         digitalWrite(pin_relph, 1);
164     }
165 }
166
167 if(((int)do_val>doup) {
168     if(lock2=1){
169         lock2=1;
170         digitalWrite(pin_relcr, 1);
171     }
172 }
173 if(((int)do_val<dodown){
174     if(lock2=2){
175         lock2=2;
176         digitalWrite(pin_relcr, 0);
177     }
178 }

```

```

177     }
178 }
179
180 lcd.clear();
181 lcd.setCursor(0,0);
182 lcd.print("PH:");
183 lcd.print(ph);
184
185 lcd.setCursor(9,0);
186 lcd.print(temperature);
187 lcd.write(0xdf);
188 lcd.print("C");
189
190 lcd.setCursor(0,1);
191 lcd.print("DO:");
192 lcd.print(do_val);
193 lcd.setCursor(9,1);
194 lcd.print("K");
195 lcd.print(digitalRead(pin_relph));
196 lcd.print(" K");
197 lcd.print(digitalRead(pin_relcr));
198
199 Serial.print("PH:");
200 Serial.print(ph);
201 Serial.print(" V CAL1:");
202 Serial.print(ADC_Voltage);
203 Serial.print(" DO:");
204 Serial.print(do_val);
205
206 Serial.println();
207
208 Blynk.virtualWrite(V0,temperature);
209 Blynk.virtualWrite(V1,ph);
210 Blynk.virtualWrite(V2,do_val);
211
212 digitalWrite(led,digitalRead(led)*1);

```

```

210 Blynk.virtualWrite(V2,do_val);
211
212 digitalWrite(led,digitalRead(led)*1);
213
214 }
215
216 void setup()
217 {
218     EEPROM.begin(512);
219     CAL1_V = eeprom_read(10);
220     phup = EEPROM.read(20);
221     phdown = EEPROM.read(21);
222     doup = EEPROM.read(22);
223     dodown = EEPROM.read(23);
224
225     Serial.begin(9600);
226     pinMode(pin_relph, OUTPUT);
227     pinMode(pin_relcr, OUTPUT);
228     digitalWrite(pin_relph, 1);
229     digitalWrite(pin_relcr, 1);
230     pinMode(led, OUTPUT);
231
232     lcd.begin();
233     lcd.backlight();
234     // tampilan awal lcd
235     lcd.clear();
236     lcd.setCursor(0,0);
237     lcd.print("%ssid%");
238     lcd.print(ssid);
239     lcd.setCursor(0,1);
240     lcd.print("pass:");
241     lcd.print(pass);
242
243     Serial.println("starting");
244     // inisialisasi blynk
245     Blynk.begin(auth, ssid, pass);

```

```
239   lcd.setCursor(0,2);
240   lcd.print("pass:");
241   lcd.print(pass);
242
243   Serial.println("starting");
244   // inisialisasi blynk
245   Blynk.begin(auth, ssid, pass);
246   timers.setInterval(1000L, tasktimer);
247
248   ph_init();
249   sensors.begin();
250   Serial.println("ready");
251 }
252
253 void loop()
254 {
255
256   Blynk.run(); // jalankan blynk
257   timers.run(); // jalankan timer
258
259   if(Serial.available()){
260     char c = Serial.read();
261     if(c=='a'){ // ketik huruf a untuk kalibrasi ph 4
262       cal_4(voltage);
263     }
264     if(c=='b'){ // ketik huruf b untuk kalibrasi ph 7
265       cal_7(voltage);
266     }
267     if(c=='c'){ // ketik huruf c untuk kalibrasi do
268       CAL1_V = ADC_Voltage;
269       eeprom_write(10, CAL1_V);
270       Serial.println(CAL1_V);
271       Serial.println("CAL DO OK");
272     }
273   }
274 }
```

## F-Diskusi Dengan Pemilik Minaverse Company

