

**SISTEM KENDALI TDS DAN pH PADA AKUAPONIK
MENGUNAKAN METODE LOGIKA FUZZY BERBASIS IoT**

SKRIPSI

Disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (S.T)



Disusun oleh:

ALIFIAN DZIKRI RAMADHAN

3332150049

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SULTAN AGENG TIRTAYASA**

2023

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya sebagai penulis skripsi berikut:

Judul : Sistem Kendali TDS Dan pH Pada Akuaponik
Menggunakan Metode Logika Fuzzy Berbasis *IoT*
Nama Mahasiswa : Alifian Dzikri Ramadhan
NIM : 3332150049
Fakultas/Jurusan : Teknik / Teknik Elektro

Menyatakan dengan sesungguhnya bahwa skripsi tersebut diatas adalah benar hasil karya asli saya dan tidak memuat hasil karya orang lain, kecuali dinyatakan melalui rujukan yang benar dan dapat dipertanggungjawabkan. Apabila dikemudian hari ditemukan hal-hal yang menunjukkan bahwa sebagian atau seluruh karya ini bukan karya saya, maka saya bersedia dituntut melalui hukum yang berlaku. Saya juga bersedia menanggung segala akibat hukum yang timbul dari pernyataan yang secara sadar dan sengaja saya nyatakan melalui lembar ini.

Cilegon, 18 September 2023



Alifian Dzikri Ramadhan
NPM.3332150049

LEMBAR PENGESAHAN

Dengan ini ditetapkan bahwa Skripsi berikut:

Judul : Sistem Kendali TDS dan pH Pada Akuaponik
Menggunakan Metode Logika *Fuzzy* Berbasis IoT

Nama Mahasiswa : Alifian Dzikri Ramadhan

NPM : 3332150049

Fakultas/Jurusan : Teknik/Teknik Elektro

Telah di uji dan di pertahankan pada tanggal 30 September 2022 melalui Sidang Skripsi di Fakultas Teknik Universitas Sultan Ageng Tirtayasa Cilegon dan dinyatakan LULUS.

Dewan Penguji

Pembimbing I : Dr. Alimuddin, S.T., M.M., M.T

Penguji I : Dr. Irma Saraswati, S.Si., M.T.

Penguji II : Heri Haryanto, S.T., M.T.

Tanda Tangan



Mengetahui,

Ketua Jurusan



Dr. Romi Wiryadinata, S.T., M.Eng

NIP. 198307032009121006

PRAKATA

Puji dan syukur kehadirat kepada Allah SWT karena atas Rahmat dan Karunia-Nya akhirnya saya dapat menyelesaikan laporan skripsi ini dengan judul, Sistem Kendali TDS dan pH Pada Akuaponik Menggunakan Metode Logika Fuzzy Berbasis IoT skripsi ini disusun sebagai salah satu syarat dalam memperoleh gelar Sarjana Teknik Elektro di Universitas Sultan Ageng Tirtayasa. Shalawat serta salam selalu tercurahkan kepada suri tauladan terbaik sepanjang masa, yaitu Nabi Muhammad SAW.

Penulisan skripsi ini tidak bisa terwujud tanpa adanya bantuan dari pihak lain. Oleh karena itu, dalam kesempatan ini saya menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah banyak membantu dalam pelaksanaan, penulisan dan penyelesaian skripsi ini, yaitu:

- (1) Kedua orang tua ibu dan bapak tercinta, dan adik-adik saya yang telah memberikan nasehat, semangat, doa, dan materi yang tak terhingga nilainya.
- (2) Bapak Dr. Romi Wiryadinata, S.T., M. Eng., selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Sultan Ageng Tirtayasa sekaligus Dosen Pembimbing Akademik.
- (3) Bapak Dr. Alimuddin, S.T., M.M., M.T. selaku Dosen Pembimbing I Skripsi yang dengan sabar telah banyak memberikan arahan, bimbingan dan motivasi kepada saya dalam menyelesaikan skripsi.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Cilegon, 18 Agustus 2022

Alifian Dzikri Ramadhan

ABSTRAK

Alifian Dzikri Ramadhan
Teknik Elektro

Sistem Kendali Tds dan pH pada Akuaponik Menggunakan Metode Logika Fuzzy

Kualitas air sangat berpengaruh pada perkembangan akuaponik, pH dan TDS merupakan parameter penting dalam kualitas air. Penelitian ini mengendalikan nilai pH dan TDS menggunakan metode logika Fuzzy untuk mencapai *setpoint*. Metode logika Fuzzy nantinya bisa mengendalikan pompa untuk merubah nilai pH dan TDS sesuai dengan *setpoint*. *Setpoint* yang dirancang pada penelitian ini untuk pH yaitu 6,5 dan TDS yaitu 1100 PPM (*parts per million*). Sistem kendali pH dan TDS pada akuaponik menggunakan metode logika Fuzzy berhasil mendapatkan performa terbaik untuk pH dengan *settling time* 300 detik dengan *error steady-state* sebesar 0,31% dan performa terbaik untuk TDS dengan *settling time* 210 detik dengan *error steady-state* sebesar 0,68%.

Kata kunci: Akuaponik, Logika Fuzzy, pH, TDS

ABSTRACT

Alifian Dzikri Ramadhan

Electrical Engineering

System Control pH and Tds In Aquaponics Using Fuzzy Logic

Water quality very affects the growth progress of plants and fish, pH and TDS are important parameters in water quality. This research controls the pH and TDS values using Fuzzy logic method to reach the setpoint value. The Fuzzy logic method will control the pump to adjust the pH and TDS values according to the setpoint. The setpoint designed in this research for pH is 6.5 and TDS is 1100 PPM (parts per million). The control system using the Fuzzy logic method got the best performance for pH at settling time 300 seconds and *error steady-state* of 0.31%. And the best performance for TDS at settling time 210 seconds and *error steady-state* of 0.68%.

Keywords: Aquaponic, Fuzzy Logic, pH, TDS.

DAFTAR ISI

| | |
|--|------------|
| HALAMAN JUDUL | i |
| LEMBAR PERNYATAAN KEASLIAN SKRIPSI | ii |
| LEMBAR PENGESAHAN | iii |
| PRAKATA | iv |
| ABSTRAK | v |
| ABSTRACT | vi |
| DAFTAR ISI..... | vii |
| DAFTAR GAMBAR..... | xi |
| DAFTAR TABEL | xii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Tujuan Penelitian | 2 |
| 1.4 Manfaat Penelitian | 3 |
| 1.5 Batasan Masalah..... | 3 |
| 1.6 Sistematika Penulisan..... | 3 |
| BAB II TINJAUAN PUSTAKA | 5 |
| 2.1 Akuaponik | 5 |
| 2.2 Kualitas Air | 5 |
| 2.3 Jenis-jenis ikan | 6 |
| 2.4 Jenis-jenis Tanaman | 7 |
| 2.5 Sistem Kendali | 7 |
| <u>2.5.1</u> Karakteristik Respon Kendali | 8 |
| 2.6 Fuzzy <i>Logic</i> | 9 |
| 2.7 Sensor | 12 |
| <u>2.7.1</u> Karakteristik Statik Alat ukur | 12 |
| <u>2.7.2</u> Sensor TDS Meter | 13 |
| <u>2.7.3</u> Sensor pH <i>meter analog kit</i> | 14 |
| 2.8 Komponen Sistem Pengendali | 14 |
| 2.9 Kajian Pustaka..... | 16 |

| | |
|--|-----------|
| BAB III METODE PENELITIAN..... | 19 |
| 3.1 Alur Penelitian | 19 |
| 3.2 Perancangan Model Akuaponik | 20 |
| 3.3 Perancangan <i>Software</i> | 20 |
| <u>3.3.1</u> Perancangan Metode Logika Fuzzy | 21 |
| <u>3.3.2</u> Perancangan Program Mikrokontroler | 23 |
| 3.4 Perancangan <i>Hardware</i> | 24 |
| <u>3.4.1</u> Perancangan Sistem Mikrokontroler | 26 |
| <u>3.4.2</u> Rangkaian Sensor TDS | 26 |
| <u>3.4.3</u> Rangkaian Sensor pH | 26 |
| <u>3.4.4</u> Rangkaian <i>Water Pump</i> | 27 |
| <u>3.5</u> Perancangan Aplikasi | 28 |
| <u>3.5.1</u> Perancangan <i>Database</i> | 28 |
| <u>3.5.2</u> Perancangan Tampilan Aplikasi Android | 28 |
| BAB IV HASIL DAN PEMBAHASAN..... | 30 |
| 4.1 Pengujian <i>Hardware</i> | 30 |
| <u>4.1.1</u> Pengujian Sensor pH | 30 |
| <u>4.1.2</u> Pengujian <i>Sensor TDS</i> | 32 |
| <u>4.1.3</u> Pengujian Relay dengan Pompa Air | 35 |
| 4.2 Pengukuran Debit Air | 35 |
| 4.3 Pengujian Monitoring..... | 35 |
| <u>4.3.1</u> Pengujian <i>Monitoring pH</i> | 36 |
| <u>4.3.2</u> Pengujian <i>Monitoring TDS</i> | 36 |
| 4.4 Pengujian Sistem Kendali Pada Perubahan <i>Setpoint</i> | 37 |
| <u>4.4.1</u> Pengujian Sistem Kendali pH | 37 |
| <u>4.4.2</u> Pengujian Sistem Kendali pH pada Pekan Pertama | 39 |
| <u>4.4.3</u> Pengujian Sistem Kendali pH pada Pekan Kedua..... | 40 |
| <u>4.4.4</u> Pengujian Sistem Kendali pH Pada Pekan Ketiga | 41 |
| <u>4.4.5</u> Pengujian Sistem Kendali pH Pada Pekan Keempat | 42 |
| <u>4.4.6</u> Pengujian Sistem Kendali TDS..... | 43 |
| <u>4.4.7</u> Pengujian Sistem Kendali TDS Pada Pekan Pertama | 45 |
| <u>4.4.8</u> Pengujian Sistem Kendali TDS Pada Pekan Kedua..... | 46 |

| | | |
|-----------------------|--|-----------|
| 4.4.9 | Pengujian Sistem Kendali TDS Pada Pekan Ketiga | 47 |
| 4.4.10 | Pengujian Sistem Kendali TDS Pada Pekan Keempat..... | 48 |
| 4.5 | Pengujian Aplikasi | 49 |
| 4.6 | Pengujian <i>Database</i> | 50 |
| 4.7 | Laju Pertumbuhan Akuaponik | 50 |
| BAB V | PENUTUP | 52 |
| 5.1 | Kesimpulan | 52 |
| 5.2 | Saran..... | 52 |
| DAFTAR PUSTAKA | | 53 |
| LAMPIRAN | | A |
| | <i>Listing Code</i> | A |
| | Hasil Pengujian | B |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Sistem akuaponik | 5 |
| Gambar 2.2 Blok Diagram Sistem Kendali Loop Terbuka..... | 8 |
| Gambar 2.3 Blok Diagram Sistem Kendali Loop Tertutup | 8 |
| Gambar 2.4 Representasi linier naik | 10 |
| Gambar 2.5 Representasi kurva segitiga | 10 |
| Gambar 2.6 Representasi kurva bahu | 11 |
| Gambar 2.7 TDS <i>meter analog kit</i> | 13 |
| Gambar 2.8 pH <i>meter analog kit</i> | 14 |
| Gambar 3.1 Rancangan Sistem Akuaponik | 20 |
| Gambar 3.2 Blok Perancangan | 21 |
| Gambar 3.3 Figure FIS pH Dan TDS | 21 |
| Gambar 3.4 Keanggotaan Error pH Dan Deltaerror pH | 22 |
| Gambar 3.5 Keanggotaan Error pH Dan Deltaerror pH | 22 |
| Gambar 3.6 Keanggotaan <i>Output</i> MotorpH Dan MotorTDS..... | 23 |
| Gambar 3.7 Perancangan <i>Hardware</i> | 24 |
| Gambar 3.8 Perancangan Sensor TDS | 26 |
| Gambar 3.9 Perancangan sensor pH..... | 27 |
| Gambar 3.10 Perancangan Water Pump..... | 27 |
| Gambar 3.11 Tampilan Home Aplikasi | 27 |
| Gambar 3.12 Tampilan Block Programming..... | 29 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Spesifikasi ESP 32 | 15 |
| Tabel 4.1 Perbandingan Sensor pH dan pH Meter pH <i>Buffer</i> 4,08..... | 31 |
| Tabel 4.2 Perbandingan Sensor pH dan pH Meter pH <i>Buffer</i> 6.88..... | 31 |
| Tabel 4.3 Perbandingan Sensor pH dan pH Meter pH <i>Buffer</i> 9,18..... | 32 |
| Tabel 4.4 Perbandingan Sensor TDS Dan TDS Meter <i>Buffer</i> TDS 1016 PPM.... | 34 |
| Tabel 4.5 Perbandingan Sensor TDS dan TDS Meter TDS <i>Buffer</i> 500 PPM..... | 34 |
| Tabel 4.6 Hasil Pengujian Sistem Kendali pH Terhadap <i>Setpoint</i> | 38 |
| Tabel 4.7 Respon Kendali Terhadap Grafik pH Pekan Pertama..... | 38 |
| Tabel 4.8 Respon Kendali Terhadap Grafik pH Pekan Kedua | 41 |
| Tabel 4.9 Respon Kendali Terhadap Grafik pH Pekan Ketiga | 42 |
| Tabel 4.10 Respon Kendali Terhadap Grafik pH Pekan Keempat | 43 |
| Tabel 4. 11 Hasil Pengujian Sistem Kendali TDS Terhadap <i>Setpoint</i> | 44 |
| Tabel 4.12 Respon Kendali Terhadap Grafik TDS Pekan Pertama..... | 45 |
| Tabel 4.13 Respon Kendali Terhadap Grafik TDS Pekan Kedua..... | 47 |
| Tabel 4.14 Respon Kendali Terhadap Grafik TDS Pekan Ketiga | 48 |
| Tabel 4.15 Respon Kendali Terhadap Grafik TDS Pekan Keempat..... | 49 |
| Tabel 4.16 Hasil Pertumbuhan Akuaponik | 51 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Akuaponik merupakan sebuah metode gabungan dari budidaya hidroponik dengan akuakultur [1]. Banyak faktor yang mempengaruhi keberhasilan budidaya akuaponik salah satunya kualitas pH dan TDS (*Total Dissolve Solid*) air akuaponik [2]. Aquaponik secara signifikan mengurangi persyaratan untuk *input* nutrisi dan pembuangan limbah, sebagai lawan dari operasi terpisah dari dua sistem. Akuaponik ramah lingkungan karena tingginya tingkat air [3].

Kandungan pH dan TDS sangatlah berpengaruh pada proses pertumbuhan akuaponik, maka dari itu ada besaran idealnya agar akuaponik bisa tumbuh secara maksimal, untuk kandungan pH sekitar 5,5 sampai 6,5 PH dan TDS sekitar 1050 sampai 1200 PPM [4] *Monitoring* pH air sangat penting dilakukan untuk mengetahui baik buruknya kualitas air. Penyediaan air bersih dengan kualitas yang buruk dapat mengakibatkan dampak yang buruk bagi kesehatan tanaman dan ikan yaitu timbulnya berbagai penyakit. [5]. Pengguna hidroponik ada yang menggunakan PPM sebagai acuan dalam menanam secara hidroponik ada juga yang menggunakan *EC meter* sebagai acuan dalam berkebun hidroponik [6].

Logika Fuzzy digunakan karena metode tersebut terkenal dapat mengakomodir data bukan *biner* dan bersifat *non linier* sehingga logika Fuzzy cocok digunakan karena menggunakan nilai *linguistic* yang tidak *linier* [7]. Dalam teori logika Fuzzy suatu nilai bias bernilai benar atau salah secara Bersama [8]. Dalam memanfaatkan logika Fuzzy, ada beberapa hal yang harus dipertimbangkan, salah satunya adalah bagaimana mengolah *input* menjadi *output* melalui sistem inferensi Fuzzy. Logika Fuzzy bisa digunakan untuk menentukan lamanya waktu atau kecepatan putaran motor dengan mengatur siklus kerja untuk setiap motor [9]. Logika Fuzzy sangat diperlukan pada suatu kondisi yang hanya bisa dijawab dengan 'ya' atau 'tidak'. Logika Fuzzy muncul sebagai akibat dari ketidakpastian suatu data atau informasi [10]. Teori Fuzzy menggunakan variabel linguistik yang nilainya adalah kata-kata Fuzzy, atau lebih bersifat ekspresi yang berakar pada bahasa alami [11].

Internet of Thing atau *IoT* adalah sebuah istilah yang dimaksudkan dalam penggunaan internet yang lebih besar, mengadopsi komputasi yang bersifat mobile dan konektivitas kemudian menggabungkannya ke dalam kehidupan sehari-hari [10]. Teknologi *Internet of Things* (*IoT*) menjawab permasalahan untuk mengatasi proses perawatan yang membutuhkan waktu lebih, dan teknologi *IoT* bisa meringankan beberapa proses perawatan diantaranya adalah pemantauan kadar pH dan TDS pada kolam akuaponik [12].

IoT adalah jaringan cerdas yang menghubungkan semua hal ke internet untuk tujuan pertukaran informasi dan berkomunikasi melalui perangkat penginderaan informasi sesuai dengan protokol yang disepakati [13]. *Firebase* adalah API yang disediakan google untuk penyimpanan dan penyelarasan data ke dalam aplikasi Android, iOS, atau web [14]. *Firebase* terdiri dari fitur pelengkap yang bisa dipadukan sesuai dengan kebutuhan (*Google*). Digunakannya *firebase*, karena platform yang telah dikembangkan oleh *Google* ini memiliki fitur cukup mumpuni untuk dimanfaatkan pada beberapa aplikasi [15]. Penelitian ini memonitoring pH dan TDS pada akuaponik kedalam jaringan publik untuk mengetahui kondisi tanaman dan ikan secara terus menerus (*realtime*) dan mengendalikan pompa secara otomatis agar pH dan TDS pada akuaponik dapat disesuaikan kebutuhan.

1.2 Rumusan Masalah

Berikut adalah rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana merancang sistem kendali pH dan TDS pada akuaponik menggunakan metode logika Fuzzy?
2. Bagaimana mengimplementasikan sistem kendali pH dan TDS pada akuaponik menggunakan metode logika Fuzzy?

1.3 Tujuan Penelitian

Berikut adalah tujuan dalam penelitian ini yaitu:

1. Menghasilkan perancangan sistem kendali pH dan TDS pada akuaponik menggunakan metode logika Fuzzy.

2. Dapat mengimplementasikan sistem kendali pH dan TDS pada akuaponik menggunakan metode logika Fuzzy.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Penelitian ini dapat mengendalikan pH dan TDS pada akuaponik menggunakan metode logika Fuzzy.
2. Mengetahui bagaimana performa sistem kendali pada sistem ini.
3. Membantu dan mempermudah dalam melakukan pemantauan guna menjaga kualitas pH dan TDS pada akuaponik.

1.5 Batasan Masalah

Pada penelitian terdapat beberapa batasan masalah yaitu:

1. Membahas secara khusus pemantauan pH dan TDS pada akuaponik menggunakan mikrokontroler ESP32.
2. Sistem dikhususkan untuk memantau dua parameter saja yaitu pH dan TDS.
3. Sistem kendali yang digunakan yaitu kendali Fuzzy
4. Tidak membahas sumber listrik cadangan.
5. Menggunakan Firebase database sebagai tempat penyimpanan data dan MIT App Inventor2 sebagai pembuat aplikasi untuk pemantauan.
6. Penelitian ini menggunakan *software* Matlab dan Arduino IDE.

1.6 Sistematika Penulisan

Skripsi ini terdiri dari lima bab. Bab I yaitu pendahuluan yang menjelaskan tentang segala sesuatu yang dilakukan dalam penelitian meliputi latar belakang masalah dalam penelitian, perumusan masalah, tujuan penelitian, manfaat penelitian dan batasan masalah.

Bab II yaitu tinjauan pustaka yang berisikan tentang perkanalan akuaponik, jenis-jenis perairan, jenis ikan, jenis tanaman, sistem kendali, metode logika Fuzzy dan komponen-komponen *hardware* dan teori *IoT*.

Bab III yaitu metodologi penelitian yang berisikan tentang tentang tahap-tahap perancangan sistem, baik secara *hardware* maupun *software* kemudian

rancangan logika Fuzzy. perancangan aplikasi android untuk pemantauan, penyimpanan data pada *database firebase*. Pembuatan aplikasi menggunakan MIT app inventor.

Bab IV yaitu hasil dan pembahasan yang berisikan data-data hasil monitoring dan kendali yang dilakukan setiap pengambilan data dari ikan dan tanaman yang nantinya dianalisis sesuai dengan sistem kendali yang telah dirancang selain itu juga menganalisa bagaimana pengaruh metode logika Fuzzy pada media ini terhadap dampak akuaponik.

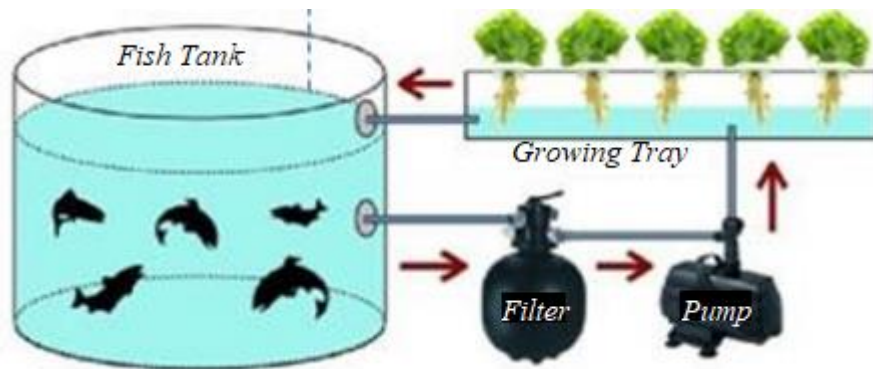
Bab V yaitu penutup yang berisikan hasil akhir dari penelitian yang telah dilakukan kemudian ditulis dalam kesimpulan dan saran untuk melengkapi kekurangan dari penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1 Akuaponik

Akuaponik adalah sistem terintegrasi yang menghubungkan hidroponik produksi tanaman dengan budidaya resirkulasi. Salah satu subsistem yang paling kompleks dan penting dari budidaya resirkulasi adalah biofiltrasi dan pembuangan limbah ikan [16]. Makanan ikan yang tertinggal dikolam bisa menjadi racun bagi ikan itu sendiri. Tanaman mengambil manfaat nutrisi dari kotoran ikan, sedangkan ikan menggunakan air bersih yang disaring oleh akar tanaman. Sistem ini menggunakan metode *Deep Flow Technique* (DFT) dimana air dari kolam mengalir ke akar tanaman, dan dengan kemiringan tertentu, air mengalir lagi ke kolam [17]. Contoh sistem akuaponik dapat dilihat pada Gambar 2.1.



Gambar 2.1 Sistem akuaponik [17]

Gambar 2.1 merupakan salah satu sistem pada akuaponik. Air dari kolam ikan mengalir ke tanaman dengan menggunakan pompa air. Limbah ikan bisa menjadi nutrisi bagi tanaman, sedangkan ikan mendapatkan air bersih yang sudah disaring oleh akar tanaman.

2.2 Kualitas Air

Kualitas air tidak terbatas pada karakteristik air, tetapi lebih dinamis yang merupakan hasil dari proses faktor lingkungan dan proses biologi, untuk menghasilkan kualitas air yang baik maka perlu ada kegiatan monitoring yang rutin. Kebutuhan kualitas air tiap spesies berbeda bahkan dalam setiap tahap perubahan dalam satu siklus hidup dalam satu spesies, sehingga kondisi air media

harus diuji terlebih dahulu sebelum membuat keputusan dan mengambil tindakan selanjutnya. Pembudidayaan harus memahami hal-hal penting yang perlu mendapat perhatian ketika sedang melakukan budidaya [18]. Berikut beberapa penjelasan singkat mengenai parameter kualitas air yang harus diperhatikan dalam pengembangan akuaponik.

1. Derajat Keasaman (pH)

pH adalah ukuran seberapa asam atau basa pada air. Rentang pH adalah 0 sampai 14, dengan 7 adalah netral dan pH kurang dari 7 menunjukkan keasaman, sedangkan pH lebih besar dari 7 menunjukkan basa. pH benar-benar merupakan ukuran jumlah relatif ion hidrogen dan hidroksil bebas didalam air. Air yang memiliki lebih banyak ion hidrogen bebas adalah bersifat asam, sedangkan air yang memiliki lebih banyak ion hidroksil bebas bersifat basa [18]. Alkalinitas merupakan kapasitas air untuk menetralkan tambahan asam, tanpa menurunkan pH larutan.

2. Alkalinitas

Alkalinitas merupakan penyangga (*buffer*) terhadap pengaruh pengasaman. Alkalinitas dinyatakan dalam mg CaCO₃/liter air (PPM) [19].

3. Suhu

Suhu merupakan faktor yang sangat berpengaruh terhadap pertumbuhan dan kelangsungan hidup ikan. Suhu pada media budidaya ikan bisa berpengaruh terhadap laju pertumbuhan, laju metabolisme serta nafsu makan ikan. Suhu air mempengaruhi kerja enzim pada bakteri, yaitu semakin tinggi suhu air maka proses metabolisme bakteri bisa meningkat sehingga aktifitas penguraian nitrogen semakin cepat [20].

4. TDS

Salah satu faktor yang sangat penting dan menentukan bahwa air yang layak konsumsi adalah kandungan TDS (*Total Dissolved Solid*) atau total kandungan unsur mineral dalam air [21].

2.3 Jenis-jenis ikan

Sesuai dengan standard PP No. 82 tahun 2001, kisaran TDS untuk kegiatan budidaya ikan yaitu 1000 mg/L, yang artinya semakin kecil konsentrasi yang

berada diperairan tersebut semakin baik juga untuk pemeliharaan ikan [22]. Berikut contoh jenis-jenis ikan yang biasa untuk dibudidaya.

1. Ikan Mas

Ikan mas dapat hidup dan berkembang baik pada ketinggian 150 sampai 600 m diatas permukaan laut (dpl) dengan suhu optimal 25 s.d. 30 °C. pH antara 6,5 sampai 7 dan dengan nilai TDS 1100 PPM [23].

2. Ikan Nila

Ikan nila pH pada air kolam sangat berpengaruh terhadap pertumbuhan dan kelangsungan hidup ikan nila. Nilai pH yang baik untuk ikan nila yaitu 7 s.d. 8,5 [24].

2.4 Jenis-jenis Tanaman

Setiap tanaman membutuhkan PPM dan pH yang berbeda-beda, agar bisa tumbuh maksimal, pemberian nutrisi juga harus tepat. Berikut contoh jenis tanaman yang biasa untuk dibudidaya:

1. Bayam

Bayam (*Amaranthus*) merupakan tumbuhan yang biasa ditanam untuk dikonsumsi daunnya sebagai sayuran hijau [25]. TDS pada bayam sekitar 1260 PPM sampai 1610 PPM dan untuk nilai pH berkisar 6,0 s.d. 7,0 [25].

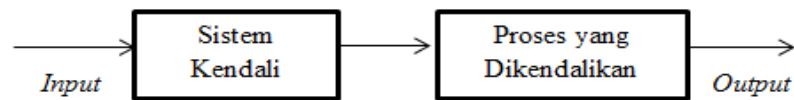
2. Pakcoy

Pakcoy (*Brassica rapa L.*) adalah tanaman sayuran daun yang satu genus dengan sawi [21]. Tingkat keasaman pada pakcoy sekitar 6,5 pH dan nilai TDS adalah 1100 PPM. Tanaman selada (*lactuca sativa L*) termasuk dalam *family compositae*. TDS pada selada sekitar 560 PPM sampai 840 PPM dan nilai pH yang optimum berkisar antara 6 s.d. 7 [26].

2.5 Sistem Kendali

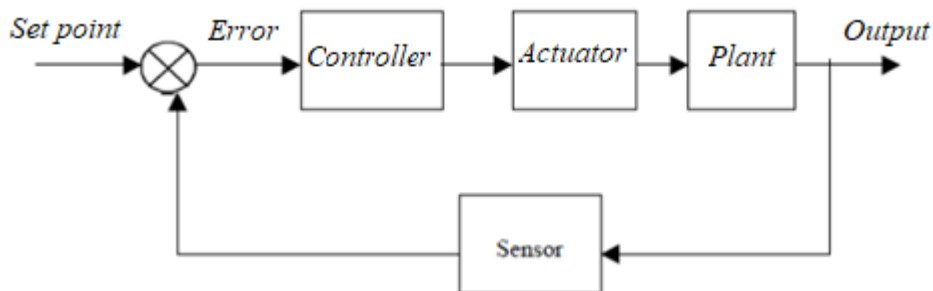
Sistem kendali didasarkan pada dasar-dasar teori umpan balik dan analisis sistem linear. Sebuah sistem kendali adalah interkoneksi komponen yang membentuk konfigurasi sistem yang akan memberikan respon sistem yang diinginkan [27]. Sistem kendali dibedakan menjadi dua, yaitu jaringan tertutup (*closed loop*) dan jaringan terbuka (*open loop*).

Sistem kendali *loop* terbuka merupakan sistem kendali dengan sinyal keluaran yang tidak memiliki pengaruh terhadap aksi pengendaliannya. Sinyal keluaran tidak diumpan balik untuk dibandingkan dengan sinyal masukan [28]. Berikut blok diagram sistem kendali *loop* terbuka dapat dilihat pada Gambar 2.2.



Gambar 2.2 Blok Diagram Sistem Kendali *Loop* Terbuka [28]

Gambar 2.2 merupakan diagram sistem kendali terbuka (*open loop*) adalah sistem yang nilai keluarannya tidak diumpan-balikan ke masukannya, artinya nilai acuan (*setpoint*) pada masukan akan bernilai tetap sehingga tanggapan keluaran tergantung dari kalibrasi sistem itu sendiri [29]. Sistem pengaturan *loop* tertutup merupakan suatu sistem pengaturan dimana sinyal keluaran mempunyai pengaruh langsung terhadap aksi kontrol. Sistem ini terdapat jaringan umpan balik [30].



Gambar 2.3 Blok Diagram Sistem Kendali *close loop* [30]

Pada gambar 2.3 merupakan diagram sistem kendali tertutup (*close loop*). Keluaran sistem akan selalu memberikan *feedback* ke masukan sampai hasil keluarannya sesuai yang diinginkan (diatur). Jadi keluaran akan berhenti memberikan *feedback* apabila hasilnya sudah sesuai.

2.5.1 Karakteristik Respon Kendali

Seringkali, karakteristik performansi sistem kendali dinyatakan dalam bentuk respon transien. Dalam menentukan karakteristik respon transien sistem kendali terhadap masukan tangga satuan ditunjukkan seperti pada Gambar 2.4 dan biasanya dicari parameter berikut [31]:

4. Waktu tunda (t_d), adalah waktu yang diperlukan respon untuk mencapai setengah harga naik yang pertama kali.
5. Waktu naik (t_r), adalah waktu yang diperlukan respon untuk naik dari 10 sampai 90%, 5 sampai 95%, atau 0 sampai 100% dari harga akhirnya, untuk sistem orde kedua redaman kurang, biasanya digunakan waktu 0 s.d. 100%. Sistem redaman lebih, biasanya digunakan waktu naik 10 s.d. 90%.
6. Waktu puncak (t_p), adalah waktu yang diperlukan respon untuk mencapai puncak lewatan yang pertama kali.
7. (Persen) lewatan maksimum (M_p), adalah harga puncak maksimum dari kurva respon yang diukur dari satu. Jika harga keadaan tunak respon tidak sama dengan satu, maka biasa digunakan persen lewatan maksimum.
5. *Error Steady State* (ESS) adalah nilai *error* yang terjadi setelah menjadi *setpoint*.
6. Waktu penetapan (t_s), adalah waktu yang diperlukan kurva respon untuk mencapai dan menetap dalam daerah sekitar harga akhir yang ukurannya ditentukan dengan presentase mutlak dari harga akhir (biasanya 5% atau 2%).

2.6 Fuzzy Logic

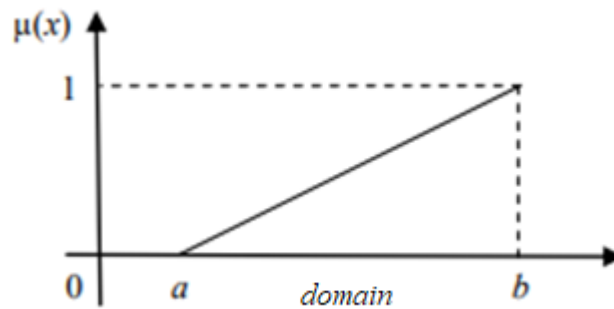
Fuzzy logic digunakan untuk menyatakan hukum operasional dari suatu sistem dengan ungkapan bahasa, bukan dengan persamaan matematis. Banyak sistem yang terlalu kompleks untuk dimodelkan secara akurat, meskipun dengan persamaan matematis yang kompleks, dalam kasus seperti itu, ungkapan bahasa yang digunakan dalam Fuzzy logic dapat membantu mendefinisikan karakteristik operasional sistem dengan lebih baik. Teori himpunan klasik yang disebut juga dengan himpunan *crisp* (himpunan tegas) hanya dikenal dua kemungkinan dalam fungsi keanggotaannya, yaitu kemungkinan termasuk keanggotaan himpunan (logika 1) atau kemungkinan berada diluar keanggotaannya (logika 0). Teori himpunan Fuzzy tidak hanya memiliki dua kemungkinan dalam menentukan sifat keanggotaannya tetapi memiliki derajat keanggotaan yang nilainya antara 0 dan 1 [32].

Fuzzifikasi adalah proses dimana mengolah masukan berupa bilangan *crisp* menjadi himpunan Fuzzy [33]. Terdapat beberapa bentuk fungsi yang bisa

digunakan untuk mendapatkan nilai keanggotaan dengan melalui pedekatan fungsi [34]:

1. Representasi linier

Representasi linier naik merupakan pemetaan *input* ke derajat keanggotaannya yang digambarkan menggunakan suatu garis lurus. Terdapat dua keadaan himpunan Fuzzy yang linier.

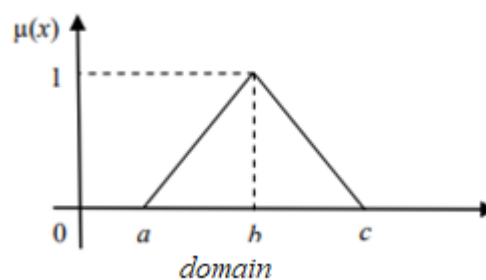


Gambar 2.4 Representasi linier naik

Pada Gambar 2.4 kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.

2. Representasi kurva segitiga

Representasi kurva Segitiga merupakan gabungan antara dua garis linier yaitu garis linier naik dan garis linier turun.

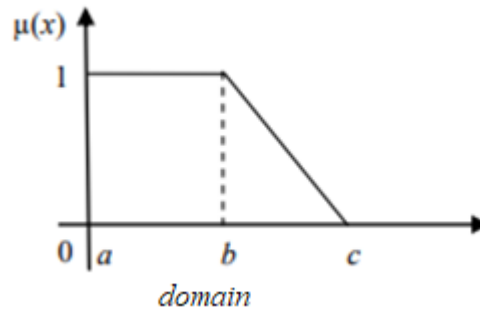


Gambar 2.5 Representasi kurva segitiga

Pada gambar 2.5 representasi kurva segitiga, pemetaan *input* ke derajat keanggotaannya digambarkan dengan bentuk segitiga dimana pada dasarnya bentuk segitiga tersebut gabungan antara 2 garis (*linear*). Nilai-nilai disekitar b memiliki derajat keanggotaan turun yang cukup tajam.

3. Representasi kurva bahu

Representasi kurva bentuk bahu terdiri dari kurva bentuk bahu kiri dan kurva bentuk bahu kanan. Representasi kurva bentuk bahu kiri merupakan gabungan dari garis yang memiliki nilai keanggotaan satu dengan garis linier turun.



Gambar 2.6 Representasi kurva bahu

Pada gambar 2.6 himpunan Fuzzy bahu, bukan segitiga, digunakan untuk mengakhiri *variabel* suatu daerah Fuzzy. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar.

Inferensia Fuzzy merupakan sistem yang menggunakan teori himpunan Fuzzy untuk memetakan *input* menjadi *output*. Struktur dasar dari sistem inferensia Fuzzy terdiri dari tiga komponen konseptual yaitu [35]:

1. Aturan dasar, terdiri dari aturan Fuzzy yang telah diseleksi.
2. Basis data (kamus), dimana mendefinisikan fungsi keanggotaan yang digunakan pada aturan Fuzzy.
3. Mekanisme penalaran, yang melakukan prosedur inferensi pada aturan dan diberikan fakta untuk mendapatkan *output* yang wajar atau kesimpulan.

Input dari proses defuzzifikasi adalah suatu himpunan Fuzzy yang diperoleh dari komposisi aturan-aturan Fuzzy, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan Fuzzy tersebut, sehingga jika diberikan suatu himpunan Fuzzy dalam *range* tertentu, maka harus dapat diambil suatu nilai crisp tertentu sebagai keluarannya. Ada beberapa metode defuzzifikasi pada mamdai [36].

2.7 Sensor

Sensor merupakan perangkat pendukung untuk mengubah besaran fisik menjadi besaran listrik. Secara umum semua sensor bekerja secara *analog*. Besaran yang dihasilkan oleh sensor adalah besaran *analog*, yaitu berupa arus listrik dengan nilai tegangan tertentu, agar arus listrik yang dihasilkan sensor dapat diproses secara *digital* maka besaran tersebut harus diubah menjadi besaran *digital* [37].

Pada dasarnya sensor dan transduser mempunyai definisi sama yaitu menerima rangsangan dari luar dan mengubahnya menjadi sinyal listrik. Sensor dan transduser mempunyai perbedaan yang sangat kecil yaitu pada koefisien konversi energi. Sensor itu sendiri terdiri dari transduser atau tanpa penguat atau pengolahan sinyal yang terbentuk dalam satu indera [38].

Sensor tidak beroperasi sendiri, umumnya sensor merupakan bagian dari sistem yang terdiri dari pengkondisi sinyal dan berbagai rangkaian pemrosesan sinyal *analog* atau *digital* [39].

2.7.1 Karakteristik Statik Alat ukur

Karakteristik statis instrumen merupakan hubungan antara *output* sebuah elemen (instrumen) dengan *inputnya* ketika *inputnya* konstan maupun berubah perlahan, yang termasuk dalam karakteristik statis adalah *range*, linieritas, sensitivitas, resolusi, akurasi, presisi, toleransi [40].

1. *Range* adalah nilai minimum hingga maksimum suatu elemen. *Range* terdiri dari *range input* dan *range output*.
2. *Span* merupakan selisih nilai maksimum dengan nilai minimum. *Span* terdiri dari *span input* dan *span output*.
3. Linieritas merupakan hubungan nilai *input* dan *output* alat ukur ketika terletak pada garis lurus. Garis lurus ideal merupakan garis yang menghubungkan titik minimum *input* atau *output* dengan titik maksimum *input* atau *output*.
4. Dalam beberapa keadaan, bahwa dari persamaan linieritas muncul garis yang tidak lurus yang biasa disebut non-linier atau tidak linier. Didalam fungsi garis yang tidak linier ini menunjukkan perbedaan antara hasil pembacaan actual atau nyata dengan garis lurus idealnya.

5. Sensitivitas diartikan seberapa peka sebuah sensor terhadap besaran yang diukur. Sensitivitas juga bisa diartikan sebagai perubahan *output* alat dibandingkan perubahan *input* dalam satu satuan.
6. Resolusi merupakan perubahan terbesar dari *input* yang dapat terjadi tanpa adanya perubahan pada *output*
7. Akurasi merupakan ketepatan alat ukur untuk memberikan nilai pengukuran yang mendekati nilai sebenarnya.
8. Presisi adalah kemampuan instrument atau elemen untuk menampilkan nilai yang sama pada pengukuran berulang singkat.

2.7.2 Sensor TDS Meter

TDS Sensor merupakan sensor kompatibel arduino yang digunakan untuk mengukur kadar TDS pada air. TDS sendiri merupakan kadar konsentrasi objek solid yang terlarut dalam air. Semakin tinggi nilai TDS nya maka semakin keruh airnya, begitupun sebaliknya. Semakin rendah nilai TDS nya maka semakin jernih pula air tersebut [41].



Gambar 2.7 TDS meter analog kit [41]

Gambar 2.7 merupakan sensor TDS meter yang berfungsi untuk membaca nilai TDS pada air. Satuan nilai pada TDS yaitu PPM. Sensor TDS meter mempunyai 3 pin.

2.7.3 Sensor pH meter analog kit

Sensor pH *Meter Analog Kit* adalah alat ukur pada tingkat keasaman cairan (pH) yang menggunakan penginderaan pengukur standar industri sebagai komponen utamanya. Elektroda sensor terbuat dari membran kaca sensitif dengan impedansi kecil sehingga menghasilkan hasil pengukuran dengan respon cepat dan stabilitas terhadap suhu tinggi. Hasil pembacaan sensor bisa langsung diakses oleh mikrokontroler melalui antarmuka pH yang terdapat pada sensor. Sensor ini sangat ideal untuk aplikasi pengukuran pH cairan dalam jangka panjang [42].



Gambar 2.8 pH meter analog kit [42]

Gambar 2.8 merupakan sensor pH yang berfungsi untuk membaca nilai keasaman pada air. Satuan nilai pada keasaman yaitu pH. Sensor pH meter memiliki 3 pin.

2.8 Komponen Sistem Pengendali

Pada alat yang dibuat terdapat beberapa komponen yang berfungsi sebagai pendukung agar alat yang dibuat berjalan sesuai. Berikut beberapa komponen yang dibutuhkan untuk alat yang dibuat:

ESP32 adalah nama dari mikrokontroler menawarkan jaringan *wifi* yang mandiri sebagai jembatan dari mikrokontroler yang ada ke jaringan *wifi*. ESP32 menggunakan prosesor dual core yang berjalan. mikrokontroler ESP32 dapat dijadikan pilihan untuk digunakan pada alat peraga *interface* mikrokontroler karena mikrokontroler ini memiliki *interface* yang lengkap, juga memiliki *wifi*

yang sudah tertanam pada mikrokontroler sehingga tepat untuk digunakan pada alat peraga atau *trainer Internet of Things*. ESP32 memiliki spesifikasi seperti yang ditampilkan pada Tabel 2.1 [43].

Tabel 2.1 Spesifikasi ESP 32

| No | Atribut | Detail |
|----|--------------------|-------------------------------------|
| 1 | Tegangan | 3,3 V |
| 2 | Prosesor | <i>Tensilica</i> L108 32 bit |
| 3 | Kecepatan prosesor | Dual 160MHz |
| 4 | RAM | 520K |
| 5 | GPIO | 34 |
| 6 | ADC | 7 |
| 7 | Dukungan 802.11 | 11b/g/n/e/i |
| 8 | <i>Bluetooth</i> | BLE (<i>Bluetooth Low Energy</i>) |
| 9 | SPI | 3 |
| 10 | I2C | 2 |
| 11 | UART | 3 |

Pada tabel 2.1 merupakan spesifikasi pada mikrokontroler esp32. Esp 32 memiliki 38 kaki pin, dan tegangan yang dihasilkan dari esp 32 yaitu 3.3V. Esp 32 juga memiliki modul *wifi* yang bisa langsung digunakan oleh *user*.

Catu daya (*power supply*) adalah suatu rangkaian elektronik yang mengubah arus listrik bolak-balik menjadi arus listrik searah. Catu daya merupakan sebuah peralatan yang berfungsi sebagai penyedia daya untuk peralatan lainnya [44].

Relay adalah Saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen *electromechanical* (elektromekanikal) yang terdiri dari 2 bagian utama yakni elektromagnet (Coil) dan mekanikal (seperangkat kontak saklar atau *switch*). Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi [45].

Pompa atau *water pump* adalah suatu alat atau mesin yang digunakan untuk memindahkan cairan dari suatu tempat ke tempat yang lain melalui suatu media perpipaan dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung secara terus menerus. Pompa berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (penggerak) menjadi tenaga kinetis (kecepatan), dimana tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada sepanjang pengaliran [46].

MIT App Inventor adalah aplikasi *web open-source* yang awalnya disediakan oleh Google untuk memungkinkan pendatang baru dalam pemrograman komputer membuat aplikasi perangkat lunak pada android menggunakan antarmuka grafis hanya dengan *drag-and-drop* saja. *App Inventor* bertujuan untuk membuat pemrograman menyenangkan dan mudah diakses oleh pemula [47].

Basis data (*database*) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data tersimpan diperangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau *user* [48]. *Database* pada konsep yang sekarang ini lebih diterapkan pada penyimpanan awan (*cloud*). Banyak sistem penyimpanan *cloud* yang dapat digunakan pada sekarang ini yaitu seperti, *MySQL*, *Firebase*, *Thingspeak* dan lain-lain.

Firebase merupakan salah satu dari sejumlah penyedia layanan mBaaS. *Firebase* memberikan layanan untuk *service develop* pada saat pengembangan aplikasi yaitu (1) *realtime database* (2) *authentication* (3) *cloud messaging* (4) *storage* (5) *hosting* (6) *test lab* (7) *crash reporting* dan (7) *cloud functions* [49]. *Firebase* merupakan *platform* dibawah perusahaan Google.

2.9 Kajian Pustaka

Berdasarkan penelitian yang dilakukan, terdapat acuan dari beberapa penelitian terdahulu yang menjadi sangat penting untuk ditinjau dan dibahas dengan tujuan untuk mengetahui hubungan antara penelitian yang dilakukan dengan penelitian terdahulu, sehingga dengan menambahkan acuan tersebut, dapat menghindari adanya suatu duplikasi dalam penelitian yang dilakukan.

1. Pada penelitian dengan judul Kendali pH untuk sistem IoT hidroponik *deep flow technique* berbasis *Fuzzy logic controller*. Penelitian ini membahas tentang sistem kendali pH pada hidroponik dengan metode *Fuzzy logic*. Penelitian ini menggunakan mikrokontroller Arduino Mega2560, sehingga

perlu menambahkan modul *wifi* agar dapat akses ke internet. Data yang digunakan *input* pada penelitian yaitu *error delta error* pH. Penelitian ini menggunakan 3 himpunan disetiap nilai *inputnya*, sehingga mendapatkan 9 aturan [50]. Penelitian menggunakan 2 parameter yaitu pH dan TDS. Penelitian menggunakan 5 himpunan disetiap nilai himpunan, dan penelitian ini menjadi referensi dalam pembuatan *input, output* dan aturan logika Fuzzy pada sistem pengendalian pH dan TDS.

2. Penelitian tentang sistem akuaponik dengan judul Respon Pertumbuhan Berbagai Jenis Tanaman Selada (*Lactuca sativa*) Menggunakan Sistem Akuaponik Dengan Pada Tebar Berbeda Ikan Nila Merah (*Oreochromis sp*) Pada Teknologi Bioflok. Penelitian ini membahas tentang kualitas air yang baik bagi tanaman selada dan ikan nila merah. Nutrisi yang dibutuhkan pada tanaman selada sama dengan tanaman pakcoy. Nutrisi untuk ikan nilai sama dengan nilai ikan mas. Nilai keasaman atau pH pada penelitian ini yaitu 6 sampai 7 pH dan untuk nilai TDS yaitu sekitar 1100 PPM [51]. *Setpoint* yang digunakan pada penelitian merujuk pada penelitian ini. Penelitian hanya menggunakan nilai tunggal *setpoint* tidak berupa *range*. *Setpoint* pada penelitian yaitu pH 6 dan untuk TDS 1100 PPM.
3. Pada penelitian yang dilakukan Berdasarkan PH dan TDS Dengan Sistem IoT Menggunakan Logika Fuzzy. Metode Fuzzy sebagai pengendali dan penentu nilai tambah air asam atau basah pada aquascape. pH yang ditetapkan pada sistem ini antara 6,5 s.d. 7,3, dan pada TDS sistem akan melakukan pengurusan terhadap air yang ada pada *aquascape*, dan menambahkan dengan air berkadar TDS lebih rendah, dengan mengkalkulasi berapa penambah yang tepat untuk menjaga TDS air pada aquascape tetap dibawa 150PPM [52].
4. Pada penelitian yang dilakukan oleh Annisa Nurul Sholihah, dkk, mereka membuat Kendali TDS nutrisi hidroponik *deep flow technique* berbasis IoT menggunakan Fuzzy logic, untuk memantau nilai kendali dan nilai parameter peneliti menggunakan aplikasi Android *Blynk* dengan menggunakan *module wifi* NodeMCU ESP8266 sebagai *wireless* [53] Pada penelitian aplikasi untuk memantau nilai parameter yaitu pH dan TDS menggunakan MIT App Inventor 2.

5. Pada penelitian sebelumnya dilakukan membuat sistem pengaturan pH otomatis terhadap air akuarium ikan gurami dengan media aquaponik menggunakan Fuzzy logic control. Peneliti menggunakan mikrokontroler Arduino serta NodeMCU, dan menggunakan sensor pH meter serta sensor HCSR-04. Penelitian ini menggunakan metode Fuzzy logic control. Perhitungan metode Fuzzy dilakukan menggunakan dua parameter *input* yaitu ketinggian air dan pH air akuarium. *Output* dari hasil perhitungan berupa *timer*[54].

BAB III

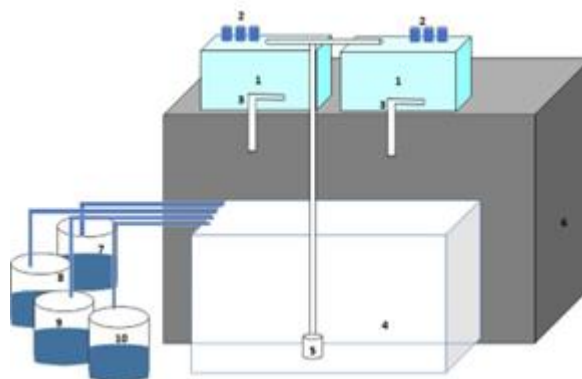
METODE PENELITIAN

3.1 Alur Penelitian

Tahapan penelitian dalam penelitian ini dibuat dalam bentuk diagram alir (*flowchart*). Tahap pertama yaitu melakukan pembelajaran terhadap referensi-referensi dari jurnal penelitian, baik jurnal penelitian nasional maupun internasional, serta buku dan artikel juga digunakan sebagai referensi dalam penelitian ini. Tahap kedua adalah melakukan desain arsitektur sistem mulai dari membuat perancangan desain alat sampai menentukan parameter. Tahap ketiga adalah merancang *hardware* alat, perancangan terbagi menjadiperancangan *box* alat, perancangan sistem penggerak pompa air, perancangan mikrokontroler serta peletakkan komponen-komponen yang digunakan. Tahap keempat adalah pengujian *hardware* meliputi pengujian serial dan logika Fuzzy, apakah sensor terbaca dengan baik pada mikrokontroler, kemudain pengujian kendali terhadap pH dan TDS, apakah alat yang dirancang berfungsi mengendalikan setiap faktor dan parameter yang sudah ditentukan, apabila terjadi kesalahan maka akan kembali ketahap 3 yaitu perancangan *hardware* sampai *hardware* dapat berfungsi dengan baik. Tahap kelima adalah perancangan perangkat lunak yang disesuaikan dengan kebutuhan. Perangkat lunak yang digunakan antara lain Arduino IDE untuk pemrograman ESP32 dan MIT App Inventor2 untuk membuat dan mengatur tampilan antarmuka pemantauan pada android. Tahap keenam adalah pengujian *software* agar sistem dapat berjalan dengan baik. Jika pengujian *software* ini gagal, maka dilakukan perbaikan pada rancangan *software*. Tahap ketujuh adalah pengambilan data yang dilakukan untuk menganalisa sistem yang sudah dibuat. Pengambilan data meliputi nilai pH dan TDS sesuai parameter yang telah ditentukan serta kinerja sistem kendali yang diterapkan pada alat. Tahap kedelapan adalah penulisan laporan hasil dari penelitian yang telah dilaksanakan.

3.2 Perancangan Model Akuaponik

Pembuatan sistem akuaponik diawali dengan perancangan *hardware* akuaponik. Model akuaponik yang digunakan dalam penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Rancangan Sistem Akuaponik

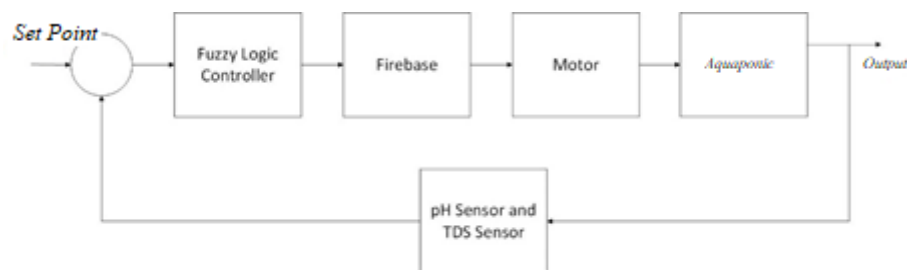
Pada gambar 3.1 perancangan model akuaponik pada penelitian ini dimulai dengan pembuatan media ikan dan tumbuhan, selanjutnya pembuatan *box panel* untuk sensor. Media ikan yang digunakan adalah akuarium dan untuk tanaman adalah wadah. Penelitian ini menggunakan akuarium dengan ukuran 65 cm x 35 cm x 40 cm (panjang, lebar dan tinggi) yang diisi dengan 15 L air tawar dan diisi ikan mas sebanyak 4 ekor. Dilanjutkan dengan pembuatan wadah pH *up*, pH *down*, TDS *up* dan TDS *down* untuk mengimbangi nilai pH dan TDS pada akuarium yang digunakan. Wadah tanaman yang digunakan sebagai media tanaman berukuran 40 cm x 20 cm (panjang dan tinggi) dan terdapat 10 netpot untuk tanaman pakcoy, untuk ukuran meja akuaponik ini yaitu 100 cm x 50 cm x 100 cm (panjang, lebar dan tinggi).

3.3 Perancangan Software

Pada penelitian sistem kendali pH dan TDS ini perancangan *software* yang dirancang meliputi perancangan program metode logika Fuzzy menggunakan MATLAB dan perancangan mikrokontroler yang menggunakan *software* Arduino IDE dengan mikrokontroler yang digunakan adalah ESP 32.

3.3.1 Perancangan Metode Logika Fuzzy

Perancangan metode logika Fuzzy pada penelitian sistem kendali pH dan TDS ini menggunakan *software* Matlab. Perancangan logika Fuzzy ini dimulai dari tahapan merancang fungsi keanggotaan (*member function*) berdasarkan masukan dan keluaran, kemudian merancang basis aturan (*rule base*) yang digunakan sebagai tindakan keluaran dari Fuzzy dan terakhir fungsi defuzzifikasi. Hasil dari perancangan sistem logika Fuzzy ini diimplementasikan dalam bahasa C ke mikrokontroler yang digunakan seperti yang ditunjukkan pada Gambar 3.2.



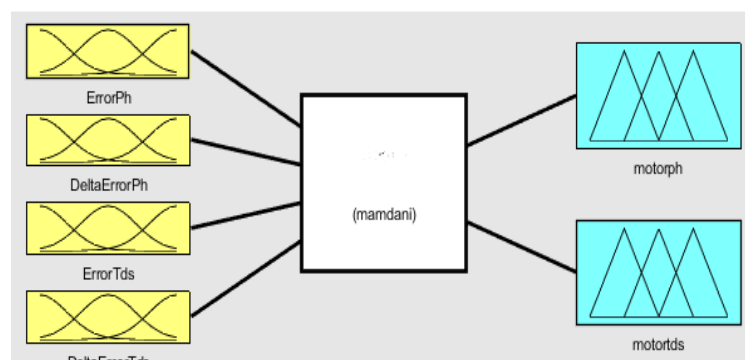
Gambar 3.2 Blok Perancangan

Pada gambar 3.2 Perancangan metode logika Fuzzy ini terdapat 3 tahapan logika Fuzzy yaitu:

fuzzifikasi, *Fuzzy logic control* dan defuzzifikasi.

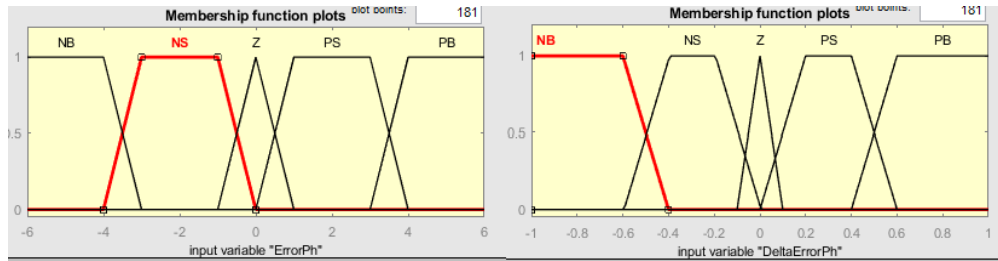
1. Fuzzifikasi

Tahapan fuzzifikasi terdiri dari empat *input* dan dua *output*. Nilai *input* untuk FIS berupa *error* pH, *deltaerror* pH, *error* TDS dan *deltaerror* TDS. Metode logika Fuzzy yang digunakan yaitu metode mamdani dan menghasilkan *output* yaitu motorpH dan motorTDS yang terdiri dari 4 pompa air.



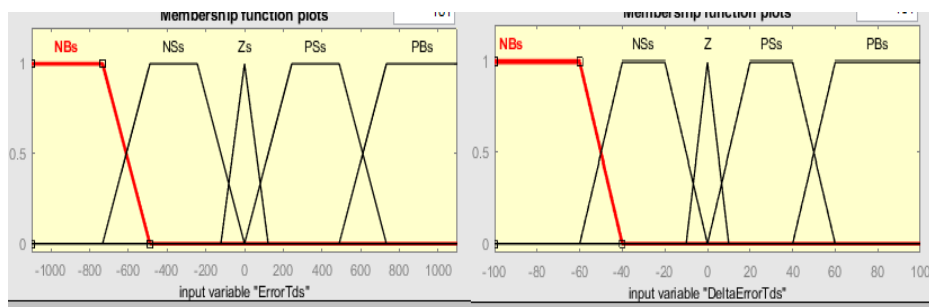
Gambar 3.3 Figure FIS pH Dan TDS

Pada gambar 3.3 merupakan figure FIS pada matlab untuk sistem *Input* pH menggunakan nilai *error* dan *deltaerror* dan *range* untuk nilai *error* pH dimulai -6 sampai 6 dan untuk *range deltaerror* pH dimulai dari -1 s.d. 1. Keanggotaan Fuzzy untuk *error* pH dan *deltaerror* pH terbagi menjadi lima. Keanggotaan *error* pH dan *deltaerror* pH yaitu NB, NS, Z, PS dan PB.



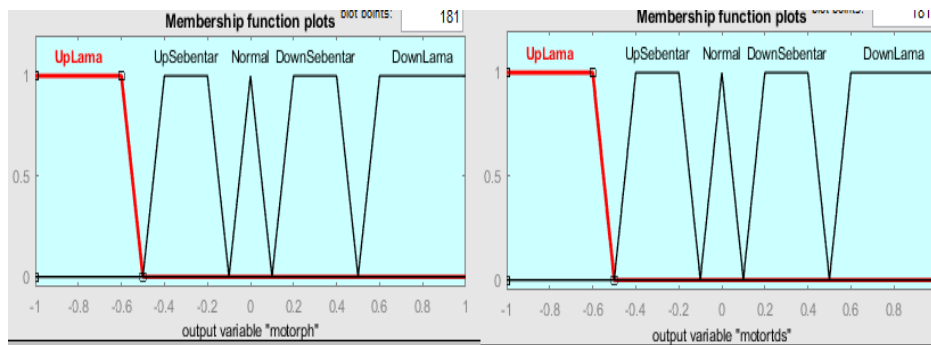
Gambar 3.4 Keanggotaan *Error* pH Dan *Deltaerror* pH

Gambar 3.4 merupakan gambar keanggotaan Fuzzy dari *error* pH dan *deltaerror* pH. *Input* TDS menggunakan nilai *error* dan *deltaerror* dan *range* untuk nilai *error* TDS dimulai -1100 s.d. -1100 dan untuk *range deltaerror* TDS dimulai dari -100 s.d. 100. Keanggotaan Fuzzy untuk *error* TDS dan *deltaerror* TDS terbagi menjadi lima. Keanggotaan *error* TDS dan *deltaerror* TDS yaitu NBs, NSs, Zs, PSs dan PBs.



Gambar 3.5 Keanggotaan *Error* pH Dan *Deltaerror* pH

Gambar 3.5 merupakan gambar keanggotaan Fuzzy dari *error* TDS dan *deltaerror* TDS. Berdasarkan Gambar 3.5 dan Gambar 3.6 diatas yang merupakan *input* dari sistem Fuzzy sehingga menentukan *output* yang berupa pompa air.



Gambar 3.6 Keanggotaan *Output* MotorpH Dan MotorTDS

Gambar 3.6 menunjukkan *output* Fuzzy motorpH dan motorTDS. Keanggotaan *output* masing-masing terdiri dari *up* lama, *up* sebentar, *normal*, *down* sebentar dan *down* lama.

2. Fuzzy Logic Control

Logika Fuzzy pada sistem kendali ini bekerja dengan logika 0 dan 1 atau sama dengan *control on* dan *off*. *Control* menggunakan logika Fuzzy terbentuk dengan sebuah *rules IF-THEN*. Sistem Fuzzy ini menggunakan 50 *rules* untuk menentukan perlakuan yang harus dilakukan pada *actuator*.

3. Defuzzifikasi

Defuzzifikasi adalah proses terakhir dalam suatu sistem logika Fuzzy. Defuzzifikasi berfungsi mengubah setiap hasil dari *inference engine* yang diekspresikan dalam bentuk Fuzzy *set* ke suatu bilangan *real*. Metode yang digunakan dalam defuzzifikasi adalah metode *centroid* yang merupakan metode yang mengambil nilai akhir berdasarkan nilai tengahnya.

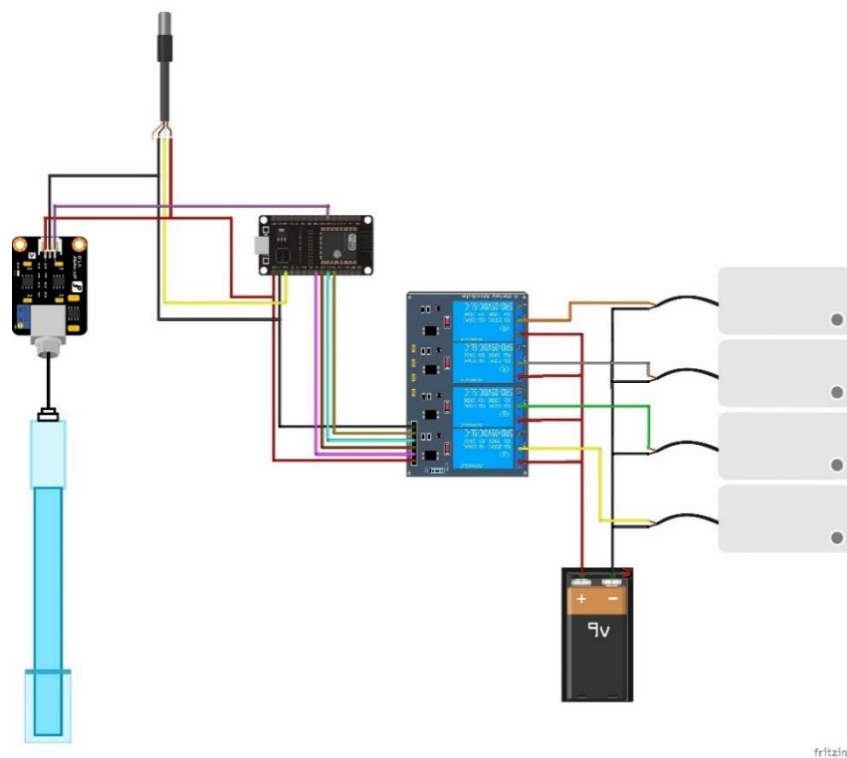
3.3.2 Perancangan Program Mikrokontroler

Perancangan program mikrokontroler yang digunakan pada penelitian ini menggunakan aplikasi *software* Arduino IDE dengan menggunakan bahasa C. Metode logika Fuzzy yang sudah dirancang sebelumnya menggunakan *software* MATLAB diubah menjadi bahasa C. Pembacaan nilai yang kurang sesuai dengan nilai *setpoint* akan dikendalikan untuk mencapai nilai *setpoint*. Pompa air akan mengalirkan nutrisi pada air untuk mencapai nilai *setpoint* yaitu untuk pH 6,5 dan

untuk TDS 1100 PPM dan mempertahankan nilai *setpoint* nilai nutrisi pada akuaponik.

3.4 Perancangan *Hardware*

Penelitian ini melakukan perancangan *hardware* yang dimaksudkan merancang sensor-sensor yang dipakai seperti *sensor* TDS dan *sensor* pH, relay dan *mikrokontroler* ESP 32. Perancangan *hardware* ini berguna untuk mempermudah dalam penelitian sehingga *actuator* berupa pompa dapat bekerja sesuai dengan pengendalian yang sudah dirancang dalam perancangan *program* sebelumnya.



Gambar 3.7 Perancangan *Hardware*

Gambar 3.7 merupakan rancangan *hardware* untuk sistem yang akan dibuat. Sistem kendali pH dan TDS pada akuaponik ini menggunakan metode logika Fuzzy sebagai sistem pengendaliannya dengan cara kerja sebagai berikut. Jika nilai pH pada akuarium melebihi nilai *setpoint* maka pompa pH *down* menyala dan menyalurkan nutrisi untuk menurunkan nilai pH ke akuarium, sebaliknya jika nilai pH pada akuarium kurang dari nilai *setpoint* maka pompa pH *up* akan

menyala dan menyalurkan nutrisi untuk menaikkan nilai pH ke akuarium. Nilai *setpoint* yang digunakan untuk menentukan besaran pH yaitu 6,5. Selanjutnya jika nilai TDS pada akuarium melebihi nilai *setpoint* maka pompa TDS *down* menyala dan menyalurkan nutrisi untuk menurunkan nilai TDS ke akuarium, sebaliknya jika nilai TDS pada akuarium kurang dari nilai *setpoint* maka pompa TDS *up* akan menyala dan menyalurkan nutrisi untuk menaikkan nilai TDS ke akuarium. Nilai *setpoint* yang digunakan untuk menentukan besaran TDS yaitu 1100 PPM.

Sistem kendali pH dan TDS ini memiliki beberapa prinsip kerja untuk dapat menjalankan sistem kendali yang baik yaitu ssebagai berikut:

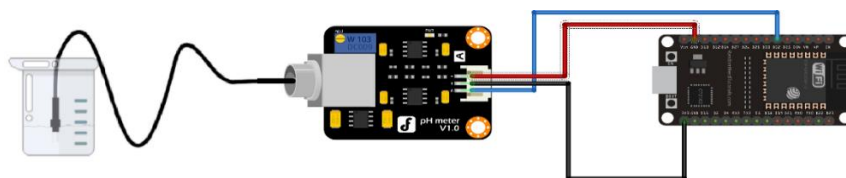
1. Sensor pH yang digunakan yaitu pH meter SKU SEN0161 yang digunakan untuk mengukur nilai pH yang terkandung pada air. Nilai akurasi sensor pH yang digunakan pada penelitian ini adalah $\pm 0.1\text{pH}$ (25 °C).
2. Sensor TDS yang digunakan yaitu TDS meter untuk mengukur nilai TDS yang terkandung pada air. Nilai akurasi sensor ini adalah $\pm 10\%$ F.S. (25 °C).
3. Pada penelitian ini menggunakan relay sebagai saklar untuk *actuator* yaitu *water pump* yang dikendalikan melali logika Fuzzy sesuai dengan kondisi nya masing-masing. Relay yang digunakan yaitu relay dengan 4 *channel* dengan tegangan yaitu 5 V.
4. *Power supply* atau catu daya yang digunakan berfungsi untuk memberikan tegangan pada *actuator*. *Water pump* membutuhkan tegangan DC 12 V. Masing masing *actuator* akan diputus tegangan nya dari *power supply* melalui relay yang dihubungkan pada masing-masing *actuator*.
5. Mikrokontroler yang digunakan pada penelitian ini menggunakan ESP 32. Mikrokontroler diberikan program logika Fuzzy untuk mengendalikan nilai pH dan TDS kemudian meyimpan data yang telah diambil dari sensor pH dan sensor TDS. ESP 32 memiliki fitur *wifi*, yang nantinya data akan ditampilkan pada aplikasi android.
6. *Actuator* yang digunakan pada penilitan ini yaitu *water pump* dengan tegangan DC 12 V. Tegangan yang dibutuhkan *actuator* diberikan oleh *power supply* yang mendapatkan perintah dari relay.

3.4.1 Perancangan Sistem Mikrokontroler

Perancangan sistem mikrokontroler dirancang guna mengetahui hubungan antara mikrokontroler dengan komponen yang digunakan baik *input* maupun *output*.

3.4.2 Rangkaian Sensor TDS

Sensor TDS yang digunakan untuk membaca nilai TDS pada air adalah sensor TDS *meter analog* yang memiliki pengkondisian sinyal yang terhubung dengan mikrokontroler. Sensor TDS ini dapat membaca nilai TDS dengan rentang dari 0 sampai 14.

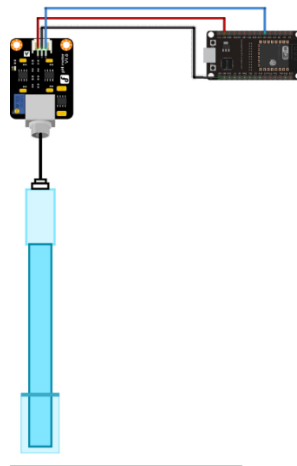


Gambar 3.8 Perancangan Sensor TDS

Gambar 3.8 merupakan rancangan pada sensor tds meter. Sensor ini memiliki 3 *port pin* yaitu *pin vcc* terhubung dengan sumber tegangan 3,3 V pada mikrokontroler, *pin gnd* terhubung dengan *groud* pada mikrokontroler, dan *pin sinyal analog* terhubung dengan *pin D32* pada mikrokontroler.

3.4.3 Rangkaian Sensor pH

Sensor pH yang digunakan untuk membaca nilai pH pada air adalah sensor pH *meter analog* yang memiliki pengkondisian sinyal yang terhubung dengan mikrokontroler. Sensor pH ini dapat membaca nilai pH dengan rentang dari 0 sampai 14.

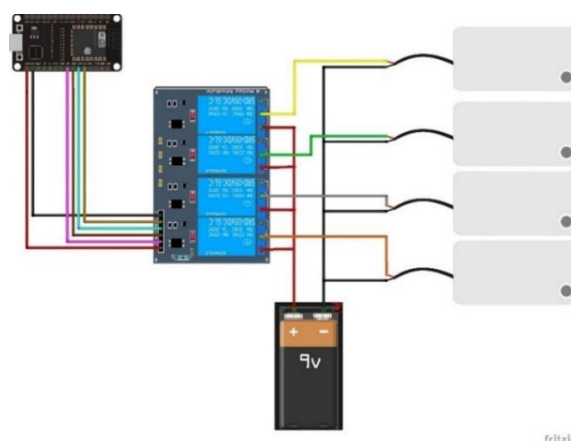


Gambar 3.9 Perancangan sensor pH

Gambar 3.9 merupakan perancangan sensor pH meter. Sensor ini memiliki 3 port pin yaitu pin vcc terhubung dengan sumber tegangan 3,3 V pada mikrokontroler, pin gnd terhubung dengan *groud* pada mikrokontroler, dan pin sinyal *analog* terhubung dengan pin D32 pada mikrokontroler.

3.4.4 Rangkaian *Water Pump*

Pada penelitian ini menggunakan relay 4 channel yang berfungsi untuk *control water pump*. *Water pump* sendiri berfungsi untuk memberi nutrisi pada akuaponik. Gambar 3.11 adalah perancangan *water pump*.



Gambar 3.10 Perancangan *Water Pump*

Pada gambar 3.10 merupakan perancangan *water pump* yang akan dikontrol secara otomatis *on* atau *off* oleh *relay*. *Water pump* yang digunakan sejumlah 4 buah.

3.5 Perancangan Aplikasi

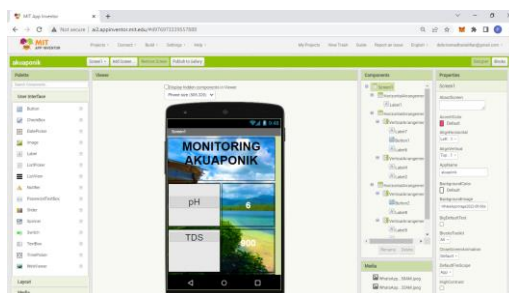
Data Sistem pemantauan atau *monitoring* pada penelitian ini disimpan pada *firebase*. *Monitoring* pada penelitian ini menggunakan aplikasi dari MIT app inventor 2 yang akan dipasang pada android. Sistem *monitoring* ini akan dilakukan secara *realtime*.

3.5.1 Perancangan Database

Mikrokontroler ESP32 yang terhubung ke internet akan mengirim data hasil dari metode logika Fuzzy ke *firebase database*. *Firestore database* akan menerima data berupa *input pH*, *error pH*, *deltaerror pH*, *actuator pH*, *input TDS*, *error TDS*, *deltaerror TDS* dan *actuator TDS*.

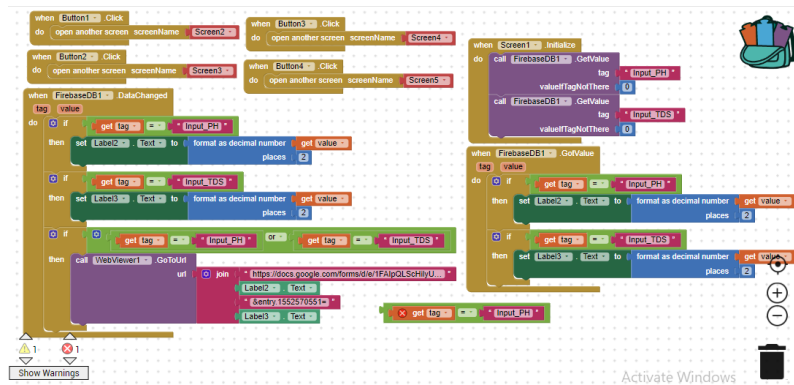
3.5.2 Perancangan Tampilan Aplikasi Android

Sistem pemantauan pada penelitian ini menggunakan aplikasi android yang dibuat dengan menggunakan MIT App Inventor, pada halaman MIT App Inventor terdapat dua halaman utama, yaitu halaman *designer* dan halaman *block*. Halaman *designer* digunakan untuk mendesain tampilan aplikasi dengan berbagai komponen dan *layout* yang disediakan sesuai dengan keinginan. Sedangkan halaman *block* digunakan untuk memprogram jalannya aplikasi android sesuai dengan tujuan. Setelah program dijalankan maka akan menampilkan hasil perancangan aplikasi pemantauan yang didalamnya terdapat nilai pH dan TDS serta kondisi dari aktuator. Tampilan perancangan aplikasi pemantauan dapat dilihat pada Gambar 3.11.



Gambar 3.11 Tampilan *Home* Aplikasi

Pada gambar 3.11 merupakan tampilan *home* pada aplikasi yang dibuat, pada tampilan *home* terdapat nilai pH dan nilai TDS secara *real time* pada kondisi akuaponik. Selanjutnya yaitu membuat *block programming* pada halaman *block* untuk membuat program halaman awal dan perancangan antarmuka *monitoring* pada tiap-tiap halaman. Perancangan *block programming* aplikasi *monitoring* dapat dilihat pada Gambar 3.12.



Gambar 3.12 Tampilan *Block Programming*

Pada gambar 3.12 adalah tampilan beberapa *block programming* yang sudah dibuat pada aplikasi MIT App Inventor. Aplikasi MIT App Inventor terdapat beberapa fitur untuk membuat *block programming* untuk membantu pengguna.

BAB IV

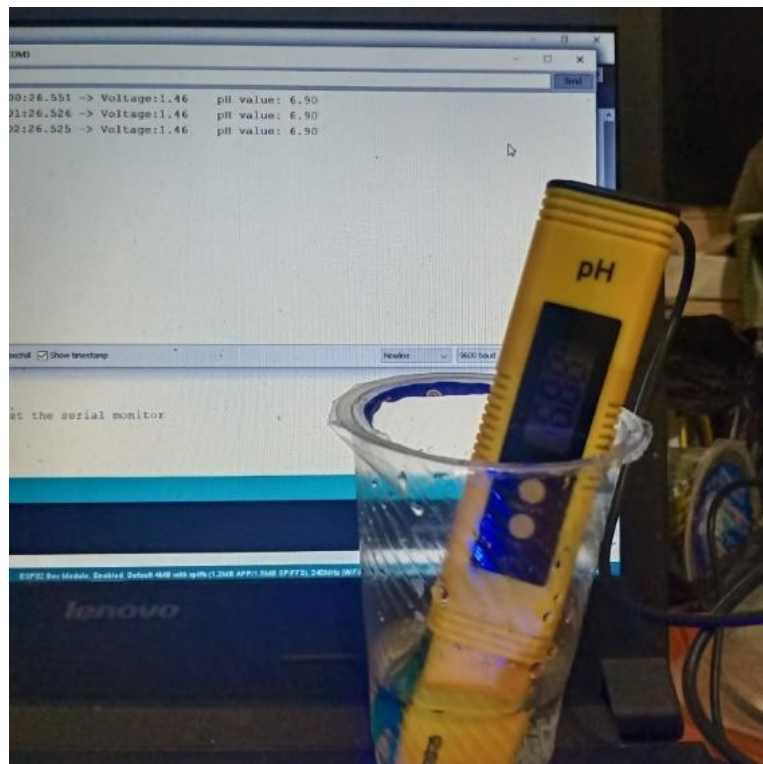
HASIL DAN PEMBAHASAN

4.1 Pengujian *Hardware*

Penelitian ini dilaksanakan pada *palm hills* dan akan disimpan dalam *box panel* yang berisi mikrokontroler yaitu ESP 32, relay dan *actuator* yaitu pompa air. Terdapat dua sensor yang digunakan pada penelitian ini yaitu sensor pH sebagai pembaca nilai pH dalam air dan sensor TDS sebagai pembaca nilai TDS pada air, dan terdapat *power supply* yang berfungsi untuk memberi tegangan.

4.1.1 Pengujian Sensor pH

Langkah awal pada penelitian ini dilakukan dengan kalibrasi *sensor* yaitu membandingkan *sensor pH meter* dengan *pH meter*. Pengujian kalibrasi *sensor* dilakukan selama 5 menit dengan *buffer* pH yang disimpan pada gelas ukur sebanyak 50 ml. Pengujian *sensor* pH dapat dilihat pada Gambar 4.1.



Gambar 4.1 Pengujian *Sensor* pH dan *pH Meter*

Pada gambar 4.1 Pengujian pertama dilakukan menggunakan pH *buffer* yang dilarutkan dengan air aquades 50 ml. pH *buffer* yang digunakan bernilai 4,08 pH. Pengujian ini dilakukan selama 5 menit pada *sensor* pH dan pH *meter*. Hasil dari pengujian yang telah dilakukan dengan pH *buffer* 4.08 dapat dilihat pada Tabel 4.1.

Tabel 4.1 Perbandingan Sensor pH dan pH *Meter* Menggunakan pH *Buffer* 4,08

| Waktu (menit) | <i>Sensor</i> pH (pH) | pH <i>meter</i> (pH) | <i>Error</i> (%) |
|------------------------|-----------------------|----------------------|------------------|
| 1 | 4,16 | 4,08 | 1,96 |
| 2 | 4,16 | 4,08 | 1,96 |
| 3 | 4,17 | 4,08 | 2,21 |
| 4 | 4,17 | 4,08 | 2,21 |
| 5 | 4,17 | 4,08 | 2,21 |
| Rata-rata <i>Error</i> | | | 2,11 |

Pada tabel 4.1 pengujian pertama pada *sensor* pH dan pH *meter* menggunakan pH *buffer* 4.08 yang dilarutkan air aquades 50 ml didapatkan hasil *error* sebesar 2,11%. Hasil dari persentase *error* pada pengujian ini dibawah 5% dapat membuktikan bahwa *sensor* pH dapat digunakan pada penelitian ini. Selanjutnya dilakukan pengujian kedua dilakukan menggunakan pH *buffer* yang dilarutkan dengan air aquades 50 ml. pH *buffer* yang digunakan bernilai 6,88 pH. Pengujian ini dilakukan selama 5 menit pada *sensor* pH dan pH *meter*. Hasil dari pengujian yang telah dilakukan dengan pH *buffer* 6.88 dapat dilihat pada Tabel 4.2.

Tabel 4.2 Perbandingan *Sensor* pH dan pH *Meter* Menggunakan pH *Buffer* 6.88

| Waktu (menit) | <i>Sensor</i> pH (pH) | pH <i>meter</i> (pH) | <i>Error</i> (%) |
|------------------------|-----------------------|----------------------|------------------|
| 1 | 6,90 | 6,88 | 0,29 |
| 2 | 6,90 | 6,88 | 0,29 |
| 3 | 6,98 | 6,88 | 1,45 |
| 4 | 6,98 | 6,88 | 1,45 |
| 5 | 6,98 | 6,88 | 1,45 |
| Rata-rata <i>Error</i> | | | 0,99 |

Pada tabel 4.2 pengujian kedua pada *sensor* pH dan pH *meter* menggunakan pH *buffer* 6,88 yang dilarutkan air aquades 50 ml didapatkan hasil *error* sebesar 0,99%. Hasil dari persentase *error* pada pengujian ini dapat membuktikan bahwa *sensor* pH dapat digunakan pada penelitian ini. Selanjutnya dilakukan pengujian

ketiga dilakukan menggunakan pH *buffer* yang dilarutkan dengan air aquades 50 ml. pH *buffer* yang digunakan bernilai 9,18 pH. Pengujian ini dilakukan selama 5 menit pada *sensor* pH dan pH *meter*. Hasil dari pengujian yang telah dilakukan dengan pH *buffer* 9,18 dapat dilihat pada Tabel 4.3.

Tabel 4.3 Perbandingan *Sensor* pH dan pH *Meter* Menggunakan pH *Buffer* 9,18

| Waktu (menit) | <i>Sensor</i> pH (pH) | pH <i>meter</i> (pH) | <i>Error</i> (%) |
|------------------------|-----------------------|----------------------|------------------|
| 1 | 9,23 | 9,18 | 0,55 |
| 2 | 9,23 | 9,18 | 0,55 |
| 3 | 9,23 | 9,18 | 0,55 |
| 4 | 9,27 | 9,18 | 0,98 |
| 5 | 9,27 | 9,18 | 0,98 |
| Rata-rata <i>Error</i> | | | 0,72 |

Pada tabel 4.3 pengujian ketiga pada *sensor* pH dan pH *meter* menggunakan pH *buffer* 9,18 yang dilarutkan air aquades 50 ml didapatkan hasil *error* sebesar 0,72%. Hasil dari persentase *error* ini mendakati pengujian ketiga dapat membuktikan bahwa *sensor* pH dapat digunakan pada penelitian ini.

4.1.2 Pengujian *Sensor* TDS

Pengujian *sensor* TDS ini membandingkan *sensor* TDS *meter* dengan TDS *meter* yang biasa digunakan untuk mengukur TDS pada air. Pengujian kalibrasi *sensor* dilakukan sebanyak 5 menit dengan *buffer* TDS yang disimpan pada gelas ukur sebanyak 50 ml. Hasil pengujian *sensor* TDS yang dapat dilihat pada Gambar 4.2.



Gambar 4.2 Pengujian *Sensor* TDS dan TDS *Meter*

Pada gambar 4.2 pengujian pertama dilakukan menggunakan TDS *buffer* 50 ml. TDS *buffer* yang digunakan bernilai 1016 PPM. Pengujian ini dilakukan selama 5 menit pada *sensor* TDS dan TDS *meter*. Nilai perbandingan dapat dilihat pada Tabel 4.4.

Tabel 4.4 Perbandingan *Sensor* TDS Dan TDS *Meter* Menggunakan *Buffer* TDS 1016 PPM

| Waktu (menit) | <i>Sensor</i> TDS (PPM) | TDS <i>meter</i> (PPM) | <i>Error</i> (%) |
|------------------------|-------------------------|------------------------|------------------|
| 1 | 1020 | 1016 | 0,39 |
| 2 | 1020 | 1016 | 0,39 |
| 3 | 1024 | 1016 | 0,79 |
| 4 | 1024 | 1016 | 0,79 |
| 5 | 1020 | 1016 | 0,39 |
| Rata-rata <i>Error</i> | | | 0,55 |

Pada tabel 4.4 pengujian pertama pada *sensor* TDS dan TDS *meter* menggunakan 50 *buffer* TDS 1016 PPM didapatkan hasil *error* sebesar 0,55%.

Pengujian kedua dilakukan menggunakan TDS *buffer* 50 ml. TDS *buffer* yang digunakan bernilai 500 PPM. Pengujian ini dilakukan selama 5 menit pada *sensor* TDS dan TDS *meter*. Hasil dari pengujian yang telah dilakukan dengan TDS *buffer* 500 PPM dapat dilihat pada Tabel 4.5.

Tabel 4.5 Perbandingan *Sensor* TDS dan TDS *Meter* Menggunakan TDS *Buffer* 500 PPM

| Waktu (menit) | <i>Sensor</i> TDS (PPM) | TDS <i>meter</i> (PPM) | <i>Error</i> (%) |
|------------------------|-------------------------|------------------------|------------------|
| 1 | 506 | 500 | 1,2 |
| 2 | 506 | 500 | 1,2 |
| 3 | 503 | 500 | 0,6 |
| 4 | 505 | 500 | 1 |
| 5 | 505 | 500 | 1 |
| Rata-rata <i>Error</i> | | | 1 |

Pada tabel 4.5 pengujian kedua pada *sensor* TDS dan TDS *meter* menggunakan TDS *buffer* 500 PPM 50 ml didapatkan hasil *error* sebesar 1%. Hasil dari persentase *error* ini dapat membuktikan bahwa *sensor* TDS dapat digunakan pada penelitian ini.

4.1.3 Pengujian Relay dengan Pompa Air

Penelitian relay yang digunakan berada pada pin 18, 19, 21 dan 3 pada ESP 32. Pin 18 merupakan pin yang digunakan relay untuk menyalakan pompa air pada pH *up* dan pin 19 merupakan pin yang digunakan relay untuk menyalakan pompa air pada pH *down* selanjutnya pin 21 merupakan pin yang digunakan relay untuk menyalakan pompa air pada TDS *up* dan pada pin 3 merupakan pin yang digunakan relay untuk menyalakan pompa air pada TDS *down*.

4.2 Pengukuran Debit Air

Pengujian ini dilakukan untuk mengetahui waktu yang diperlukan oleh pompa untuk mengeluarkan 1,5L air. Hasil dari pengukuran debit air membutuhkan 80 detik untuk mengeluarkan 1,5L air. Hasil pengukuran debit air yang dilakukan dapat dilihat pada Gambar 4.3.



Gambar 4.3 Pengukuran Debit Air

Pada gambar 4.3 adalah cara untuk mengetahui pompa yang digunakan bekerja dengan baik, dan untuk mengetahui waktu yang dibutuhkan kinerja pompa yang digunakan.

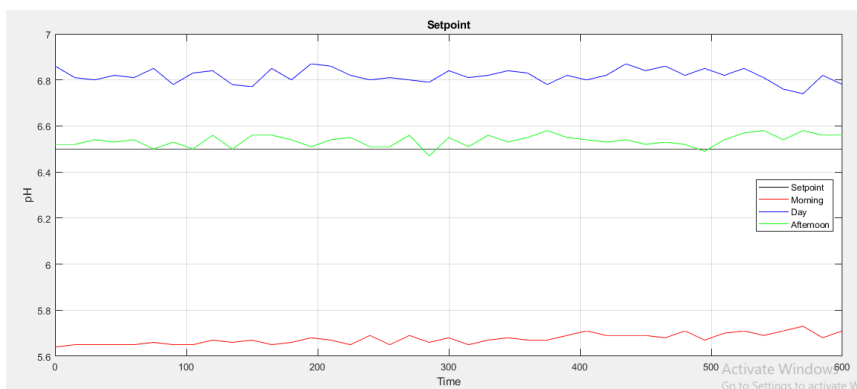
4.3 Pengujian Monitoring

Pengujian monitoring pada pH dan TDS dilakukan untuk mengetahui kondisi nilai pH dan nilai TDS menggunakan Fuzzy *logic control* yang dilakukan pada palm hills. Pengujian monitoring bisa dilihat pada aplikasi yang sudah dibuat

pada android. Monitoring pH dan TDS ini dilakukan pada pagi hari pukul 10.00, siang hari pukul 13.00 dan sore hari pukul 16.00. pengujian dilakukan selama 20 menit.

4.3.1 Pengujian *Monitoring pH*

Pengujian *monitoring pH* sebelum menggunakan sistem yang akan dibuat. Pengujian *monitoring* menggunakan mikrokontroler ESP 32 yang terkoneksi internet dan aplikasi MIT app inventor 2 yang sudah ter-*install* pada android dengan menampilkan data pH. Pembacaan nilai pH dilakukan setiap 30 detik selama pengambilan data dalam jangka waktu 5 menit. Grafik kondisi nilai pH dapat dilihat pada Gambar 4.4.



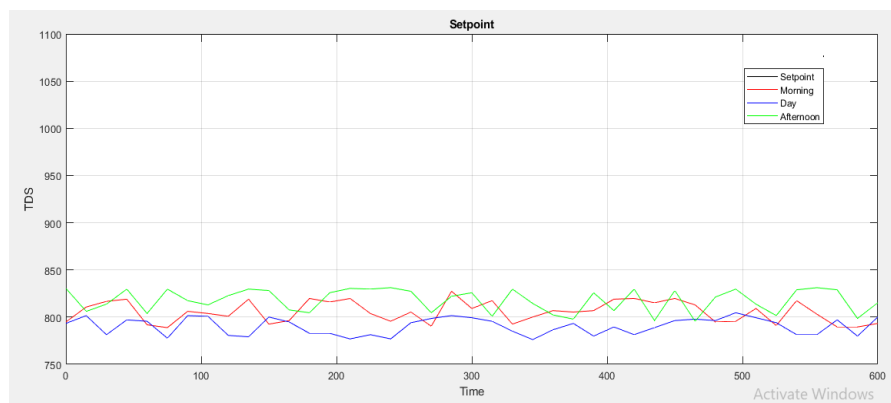
Gambar 4.4 Grafik *Monitoring pH*

Hasil pengujian *monitoring pH* yang terlihat pada gambar 4.5 menunjukkan pada pagi hari terjadi penurunan pada nilai pH sebesar 5,6 pH. Menjelang siang hari nilai pH sudah sesuai dengan nilai *setpoint* yaitu sebesar 6,5 pH, pada sore hari nilai pH mengalami perubahan signifikan yaitu 6,85. Perubahan nilai pH tidak mencapai *setpoint* dikarenakan kondisi *plant* berada pada *outdoor*.

4.3.2 Pengujian *Monitoring TDS*

Pengujian *monitoring TDS* menggunakan mikrokontroler ESP 32 pada *serial monitor* dan aplikasi MIT app inventor 2 pada android dengan menampilkan data TDS. Pembacaan nilai TDS dilakukan setiap 30 detik selama pengambilan data dalam jangka waktu 5 menit. Nilai TDS mengalami perubahan

tetapi tidak terlalu signifikan. Grafik kondisi nilai TDS dapat dilihat pada Gambar 4.5.



Gambar 4.5 Grafik *Monitoring* TDS

Hasil pengujian *monitoring* TDS yang terlihat pada gambar 4.5 menunjukkan nilai TDS pagi, siang dan sore tidak sesuai dengan nilai *setpoint*. Nilai TDS yang didapatkan pada air akuaponik 800 PPM, nilai TDS yang dibutuhkan pada plant yaitu 1100 PPM. Perubahan nilai TDS tidak mencapai *setpoint* dikarenakan kondisi plant berada pada *outdoor*.

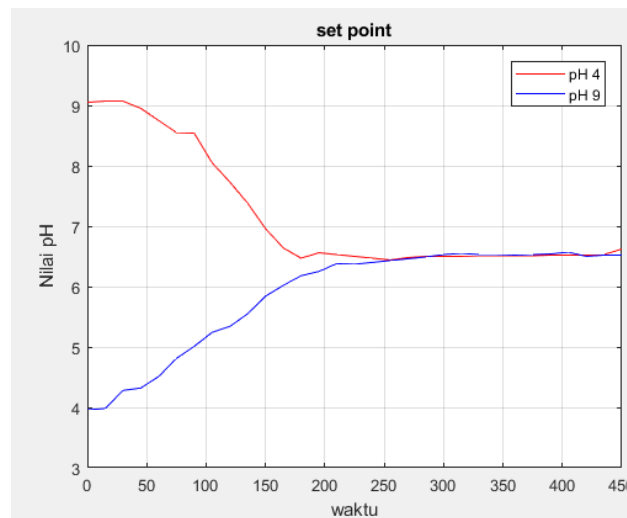
4.4 Pengujian Sistem Kendali Pada Perubahan *Setpoint*

Pengujian Sistem kendali dilakukan untuk mengetahui kondisi dan status *actuator* dapat bekerja sesuai dengan kondisi hasil *Fuzzy logic control* terhadap beberapa nilai *setpoint*. Pengujian sistem kendali menggunakan mikrokontroler ESP 32 yang sudah *diprogram*. Pengujian dilakukan pada pagi, siang dan sore hari. Pengujian ini dilakukan bertujuan untuk mengetahui kemampuan pengendali baik atau tidaknya terhadap perubahan *setpoint*.

4.4.1 Pengujian Sistem Kendali pH

Pengujian sistem kendali pH menggunakan *Fuzzy logic control* terhadap perubahan beberapa nilai *setpoint*. Hal ini dilakukan untuk mengetahui apakah sistem kendali yang dibuat mampu bekerja dengan baik atau tidak terhadap perubahan *setpoint* yang ditentukan. Pengujian sistem kendali pH terhadap perubahan nilai *setpoint* dilakukan dengan memberikan tiga perubahan nilai

setpoint yang berbeda-beda yaitu 4 pH, 6,5 pH dan 9 pH. Grafik dari hasil hasil pengujian sistem kendali pH terhadap *setpoint* dapat dilihat pada Gambar 4.6.



Gambar 4.6 Hasil Pengujian sistem Kendali pH Terhadap *Setpoint*

Pengujian sistem kendali pH pada perubahan nilai *setpoint* dimulai dengan kondisi nilai pH sebesar 4 pH pada air. Sistem kendali akan berjalan dan memberi perintah terhadap *actuator* untuk memberi nutrisi pH *up* menjadi kondisi nilai pH sebesar 6,5 pH. Pengujian selanjutnya nilai *setpoint* dimulai dengan kondisi nilai pH sebesar 9 pH. Sistem kendali akan berjalan dan memberi perintah terhadap *actuator* untuk memberi nutrisi pH *down* sehingga kondisi nilai pH pada air berubah sesuai *setpoint* yaitu sebesar 6,5 pH. Hasil pengujian sistem kendali pH terhadap *setpoint* dapat dilihat pada Tabel 4.6.

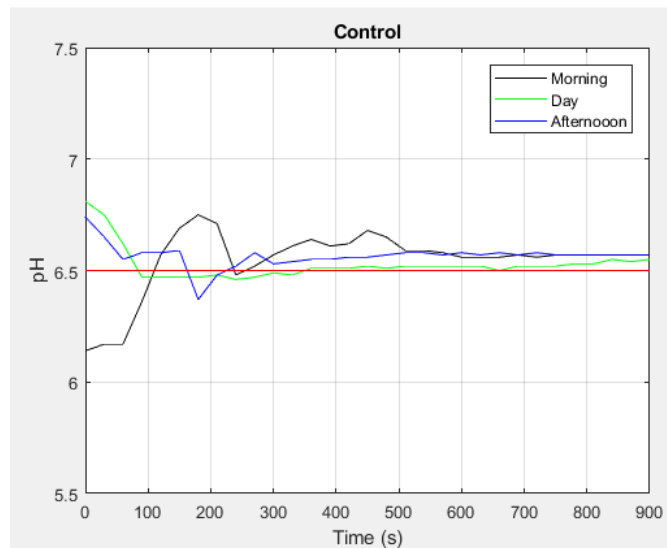
Tabel 4.6 Hasil Pengujian Sistem Kendali pH Terhadap *Setpoint*

| Kendali | Pengujian | <i>Setpoint</i> pH | pH Awal | Td (detik) | Tr (detik) |
|--------------|-----------|--------------------|---------|------------|------------|
| Logika Fuzzy | Pertama | 6,5 | 3,96 | 83,72 | 167,44 |
| | Kedua | 6,5 | 9,05 | 51,49 | 102,98 |

Pada tabel 4.6 didapatkan pengujian pertama dari pH awal menuju *setpoint* nilai td 83,72 detik dan tr 167,44 detik. Pengujian kedua nilai pH awal 9,05 menuju *setpoint* didapatkan nilai td 51,49 detik dan nilai tr 102,98 detik.

4.4.2 Pengujian Sistem Kendali pH pada Pekan Pertama

Pengujian kendali pH pada pekan pertama dilakukan pada pagi, siang dan sore hari. Pengujian ini dilakukan untuk mengendalikan nilai pH pada plan akuaponik agar sesuai dengan *setpoint* yang diinginkan. *Setpoint* yang diberikan pada plan akuaponik pH sebesar 6,5 pH. Hasil pengujian dapat dilihat pada Gambar 4.7.



Gambar 4.7 Grafik Kendali pH Pekan Pertama

Pada kendali yang ditunjukkan pada Gambar 4.7 menunjukkan hasil dari pengujian kendali pH, pada pagi hari mendapatkan nilai pH awal sebesar 6,14 pH, siang hari 6,81 pH dan sore hari 6,74 pH. Waktu yang dibutuhkan oleh *sistem* untuk mencapai *setpoint* pada pagi hari adalah 110 detik, pada siang hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* pada siang hari adalah 84 detik, pada sore hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* adalah 160 detik. Hasil pengujian kendali pH pada pekan pertama terdapat beberapa nilai dari respon kendali yang dapat dilihat pada Tabel 4.6.

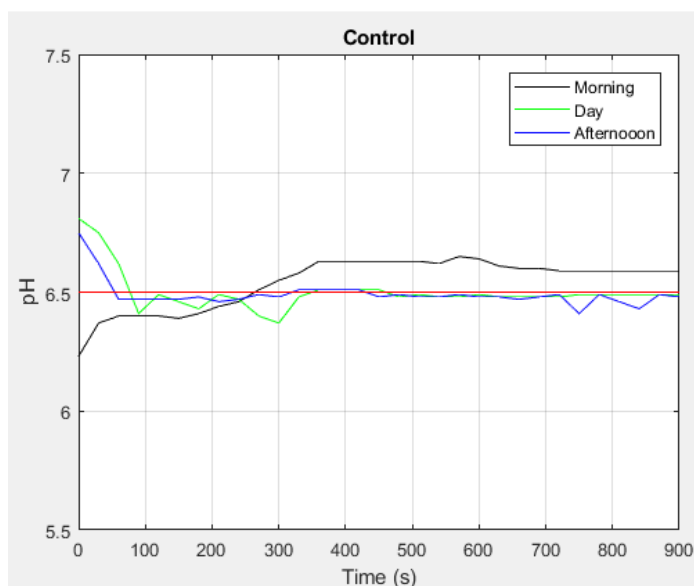
Tabel 4.6 Respon Kendali Terhadap Grafik pH Pekan Pertama

| Waktu | <i>Setpoint</i> pH | pH Awal | Td (detik) | Tr (detik) | Tp (detik) | Mp (%) | Ts (detik) | Ess (%) |
|-------|-----------------------|------------|---------------|---------------|---------------|-----------|---------------|------------|
| Pagi | 6,5 | 6,14 | 55 | 110 | 180 | 3,85 | 600 | 0,92 |
| Siang | 6,5 | 6,81 | 42 | 84 | - | - | 420 | 0,15 |
| Sore | 6,5 | 6,74 | 80 | 160 | - | - | 390 | 0,77 |

Pada tabel 4.6 didapatkan hasil pada pagi hari untuk t_d 55 detik dan untuk t_r 110 detik, pada siang hari untuk nilai t_d 42 detik dan t_r 84 detik, dan untuk sore hari didapatkan hasil t_d 80 detik dan t_r 160 detik.

4.4.3 Pengujian Sistem Kendali pH pada Pekan Kedua

Pengujian kendali pH pada pekan kedua dilakukan pada pagi, siang dan sore hari. Pengujian dilakukan untuk mengendalikan nilai pH pada akuaponik agar sesuai dengan *setpoint*. *Setpoint* yang diberikan pada plan akuaponik pH sebesar 6,5 pH. Perubahan nilai pH pada pekan kedua mendekati dengan pengujian pada pekan pertama. Hasil pengujian dapat dilihat pada Gambar 4.8.



Gambar 4.8 Grafik Kendali pH Pekan Kedua

Grafik kendali yang ditunjukkan pada Gambar 4.8 menunjukkan hasil dari pengujian kendali pH, pada pagi hari mendapatkan nilai pH awal sebesar 6,23 pH, siang hari 6,81 pH dan sore hari 6,75 pH. Waktu yang dibutuhkan oleh sistem untuk mencapai *setpoint* pada pagi hari adalah 264 detik, pada siang hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* pada siang hari adalah 77 detik, pada sore hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* adalah 55 detik. Hasil pengujian kendali pH pada pekan kedua terdapat beberapa nilai dari respon kendali yang dapat dilihat pada Tabel 4.7.

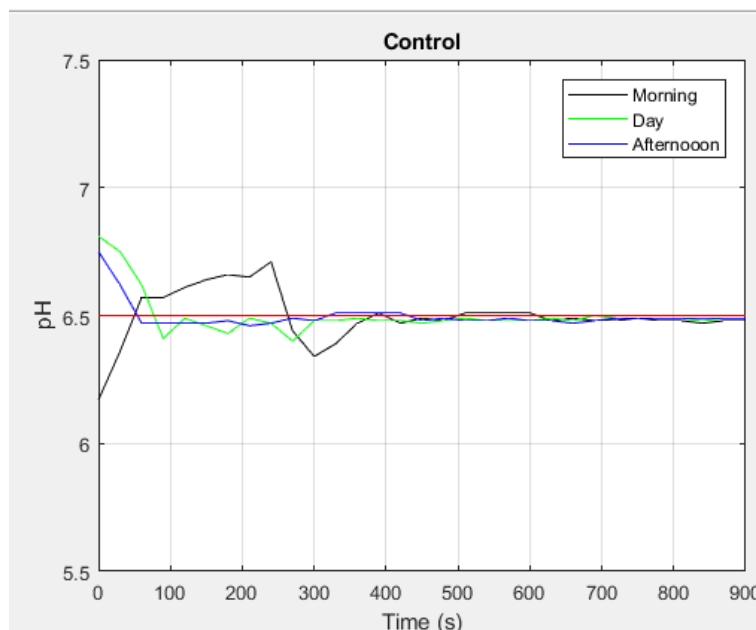
Tabel 4.7 Respon Kendali Terhadap Grafik pH Pekan Kedua

| Waktu | Setpoint pH | pH Awal | Td (detik) | Tr (detik) | Tp (detik) | Mp (%) | Ts (detik) | Ess (%) |
|-------|-------------|---------|------------|------------|------------|--------|------------|---------|
| Pagi | 6,5 | 6,23 | 132 | 264 | - | - | 360 | 2 |
| Siang | 6,5 | 6,81 | 38,5 | 77 | - | - | 380 | 0,15 |
| Sore | 6,5 | 6,75 | 27,5 | 55 | - | - | 300 | 0,31 |

Pada tabel 4.7 didapatkan hasil pada pagi hari untuk td 132 detik dan untuk tr 264 detik, pada siang hari untuk nilai td 38 detik dan tr 77 detik, dan untuk sore hari didapatkan hasil td 27 detik dan tr 55 detik.

4.4.4 Pengujian Sistem Kendali pH Pada Pekan Ketiga

Pengujian kendali pH pada pekan ketiga dilakukan pada pagi, siang dan sore hari. Pengujian ini dilakukan untuk mengendalikan nilai pH pada plan akuaponik agar sesuai dengan *setpoint* yang diinginkan. *Setpoint* yang diberikan pada plan akuaponik pH sebesar 6,5 pH. Pekan ketiga terjadi perubahan nilai pH pada sore hari sangat tinggi. Hasil pengujian dapat dilihat pada Gambar 4.9.



Gambar 4.9 Grafik Kendali pH Pekan Ketiga

Grafik kendali yang ditunjukkan pada Gambar 4.9 menunjukkan hasil dari pengujian kendali pH, pada pagi hari mendapatkan nilai pH awal sebesar 6,17 pH, siang hari 6,81 pH dan sore hari 6,75 pH. Nilai awal pH pada pagi hari waktu

yang dibutuhkan oleh sistem untuk mencapai *setpoint* pada pagi hari adalah 50 detik, pada siang hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* pada siang hari adalah 78 detik, pada sore hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* adalah 60 detik. Hasil pengujian kendali pH pada pekan ketiga terdapat beberapa nilai dari respon kendali yang dapat dilihat pada Tabel 4.8.

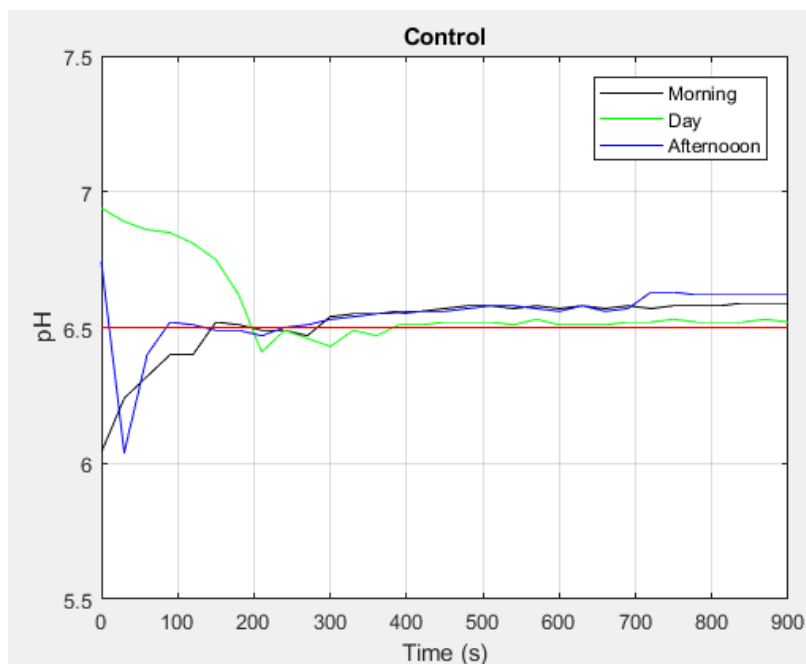
Tabel 4.8 Respon Kendali Terhadap Grafik pH Pekan Ketiga

| Waktu | Setpoint pH | pH Awal | Td (detik) | Tr (detik) | Tp (detik) | Mp (%) | Ts (detik) | Ess (%) |
|-------|-------------|---------|------------|------------|------------|--------|------------|---------|
| Pagi | 6,5 | 6,17 | 25 | 50 | 240 | 3,32 | 420 | 0,46 |
| Siang | 6,5 | 6,81 | 39 | 78 | - | - | 300 | 0,31 |
| Sore | 6,5 | 6,75 | 30 | 60 | - | - | 198 | 0,31 |

Pada tabel 4.8 didapatkan hasil pada pagi hari untuk td 25 detik dan untuk tr 50 detik, pada siang hari untuk nilai td 39 detik dan tr 78 detik, dan untuk sore hari didapatkan hasil td 30 detik dan tr 60 detik.

4.4.5 Pengujian Sistem Kendali pH Pada Pekan Keempat

Pengujian kendali pH pada pekan keempat dilakukan pada pagi, siang dan sore hari, pada pekan keempat nilai pH pada akuaponik mendekati pekan pertama dan kedua. Hasil pengujian dapat dilihat pada Gambar 4.10.



Gambar 4.10 Grafik Kendali pH Pekan Keempat

Grafik kendali yang ditunjukkan pada Gambar 4.10 menunjukkan hasil dari pengujian kendali pH, pada pagi hari mendapatkan nilai pH awal sebesar 6,04 pH, siang hari 6,94 pH dan sore hari 6,74 pH. Waktu yang dibutuhkan oleh sistem untuk mencapai *setpoint* pada pagi hari adalah 145 detik, pada siang hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* pada siang hari adalah 200 detik, pada sore hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* adalah 26 detik. Hasil pengujian kendali pH pada pekan keempat terdapat beberapa nilai dari respon kendali yang dapat dilihat pada Tabel 4.9.

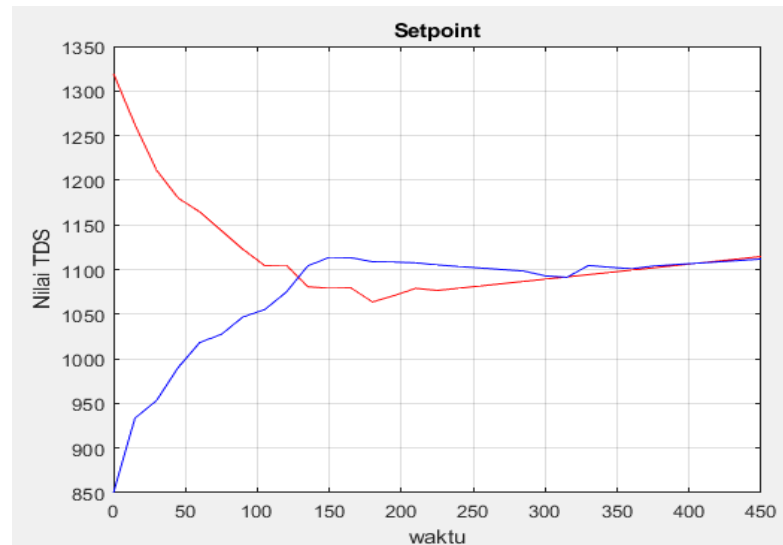
Tabel 4.9 Respon Kendali Terhadap Grafik pH Pekan Keempat

| Waktu | <i>Setpoint</i> pH | pH Awal | Td (detik) | Tr (detik) | Tp (detik) | Mp (%) | Ts (detik) | Ess (%) |
|-------|-----------------------|------------|---------------|---------------|---------------|-----------|---------------|------------|
| Pagi | 6,5 | 6,04 | 72,5 | 145 | - | - | 330 | 0,77 |
| Siang | 6,5 | 6,94 | 100 | 200 | - | - | 390 | 0,15 |
| Sore | 6,5 | 6,74 | 13 | 26 | - | - | 385 | 0,77 |

Pada tabel 4.9 didapatkan hasil pada pagi hari untuk td 72 detik dan untuk tr 145 detik, pada siang hari untuk nilai td 100 detik dan tr 200 detik, dan untuk sore hari didapatkan hasil td 13 detik dan tr 26 detik.

4.4.6 Pengujian Sistem Kendali TDS

Pengujian sistem kendali TDS menggunakan *Fuzzy logic control* terhadap perubahan beberapa nilai *setpoint*. Pengujian sistem kendali TDS terhadap perubahan nilai *setpoint* dilakukan dengan memeberikan tiga perubahan nilai *setpoint* yang berbeda-beda yaitu 850PPM, 1.100PPM dan 1.300PPM. Berikut grafik dari hasil hasil pengujian sistem kendali TDS terhadap *setpoint* dapat dilihat pada Gambar 4.11.



Gambar 4.11 Hasil Pengujian Sistem Kendali TDS Terhadap *Setpoint*

Pengujian sistem kendali TDS pada perubahan nilai *setpoint* dimulai dengan kondisi nilai TDS sebesar 850PPM pada air. Sistem kendali akan berjalan dan memberi perintah terhadap *actuator* untuk memberi nutrisi TDS *up* menjadi kondisi nilai TDS sesuai dengan *setpoint*. Pengujian sistem kendali TDS selanjutnya nilai TDS dimulai pada kondisi 1319PPM. Sistem kendali akan berjalan dan memberi perintah terhadap *actuator* untuk merubah nilai TDS sesuai dengan *setpoint*. Hasil pengujian sistem kendali TDS terhadap *setpoint* dapat dilihat pada Tabel 4.10.

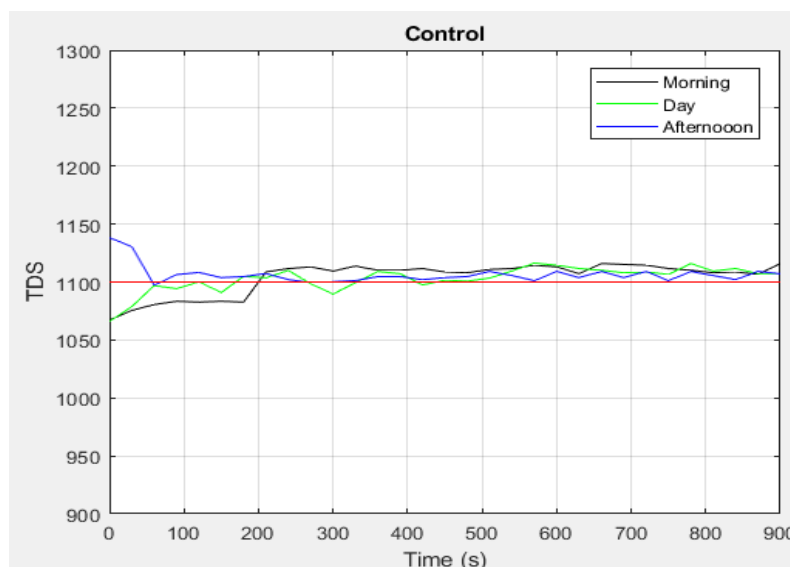
Tabel 4.10 Hasil Pengujian Sistem Kendali TDS Terhadap *Setpoint*

| Kendali | Pengujian | <i>Setpoint</i> TDS | TDS Awal | Td (detik) | Tr (detik) |
|--------------|-----------|---------------------|----------|------------|------------|
| Logika Fuzzy | Pertama | 1100 | 850,29 | 57,86 | 115,71 |
| | Kedua | 1100 | 1.319,49 | 42,4 | 84,80 |

Pada tabel 4.10 didapatkan pengujian pertama dari TDS awal menuju *setpoint* nilai td 57,86 detik dan tr 115,71 detik, pada pengujian kedua nilai TDS awal 1.319,49 menuju *setpoint* didapatkan nilai td 42,4 detik dan nilai tr 84,80 detik.

4.4.7 Pengujian Sistem Kendali TDS Pada Pekan Pertama

Pengujian kendali TDS pada pekan pertama dilakukan pada pagi, siang dan sore hari. Pengujian ini dilakukan untuk mengendalikan nilai TDS pada plan akuaponik agar sesuai dengan *setpoint* yang diinginkan. *Setpoint* TDS yang diberikan pada plan akuaponik adalah 1.100PPM. Hasil pengujian dapat dilihat pada Gambar 4.12.



Gambar 4.12 Grafik Kendali TDS Pekan Pertama

Grafik kendali yang ditunjukkan pada Gambar 4.12 menunjukkan hasil dari pengujian kendali TDS, pada pagi hari mendapatkan nilai TDS awal sebesar 1.067,27PPM, siang hari 1.066,5PPM dan sore hari 1.138,21PPM. Waktu yang dibutuhkan oleh sistem untuk mencapai *setpoint* pada pagi hari adalah 200 detik, pada siang hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* pada siang hari adalah 120 detik. Waktu yang dibutuhkan pada sore hari untuk mencapai nilai *setpoint* adalah 57,5 detik. Hasil pengujian kendali TDS pada pekan pertama terdapat beberapa nilai dari respon kendali yang dapat dilihat pada Tabel 4.11.

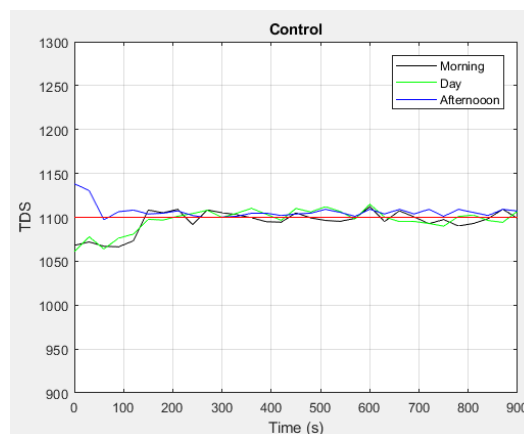
Tabel 4.11 Respon Kendali Terhadap Grafik TDS Pekan Pertama

| Waktu | <i>Setpoint</i> TDS | TDS Awal | Td (detik) | Tr (detik) | Tp (detik) | Mp (%) | Ts (detik) | Ess (%) |
|-------|------------------------|-------------|---------------|---------------|---------------|-----------|---------------|------------|
| Pagi | 1.100 | 1.067,27 | 100 | 200 | - | - | 300 | 0,8 |
| Siang | 1.100 | 1.066,05 | 60 | 120 | - | - | 450 | 0,14 |
| Sore | 1.100 | 1.138,21 | 28,75 | 57,5 | - | - | 210 | 0,68 |

Pada tabel 4.11 didapatkan hasil pada pagi hari untuk td 100 detik dan untuk tr 200 detik, pada siang hari untuk nilai td 60 detik dan tr 120 detik, dan untuk sore hari didapatkan hasil td 28,75 detik dan tr 57,5 detik.

4.4.8 Pengujian Sistem Kendali TDS Pada Pekan Kedua

Pengujian kendali TDS pada pekan kedua dilakukan pada pagi, siang dan sore hari. Pengujian ini dilakukan untuk mengendalikan nilai TDS pada plan akuaponik agar sesuai dengan *setpoint* yang diinginkan. *Setpoint* yang diberikan pada plan akuaponik TDS sebesar 1.100 PPM. Hasil pengujian dapat dilihat pada Gambar 4.13.



Gambar 4.13 Grafik Kendali TDS Pekan Kedua

Grafik kendali yang ditunjukkan pada Gambar 4.13 menunjukkan hasil dari pengujian kendali TDS, pada pagi hari mendapatkan nilai TDS awal sebesar 1.067,99PPM, siang hari 1.060,71PPM dan sore hari 1.154,29PPM. Waktu yang dibutuhkan oleh sistem untuk mencapai *setpoint* pada pagi hari adalah 142,8 detik, pada siang hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* pada siang hari adalah 200 detik. Waktu yang dibutuhkan pada sore hari untuk mencapai nilai *setpoint* adalah 240 detik. Hasil pengujian kendali TDS pada pekan pertama terdapat beberapa nilai dari respon kendali yang dapat dilihat pada Tabel 4.12.

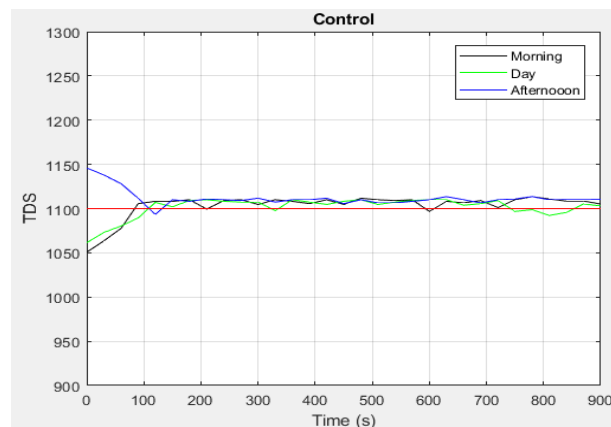
Tabel 4.12 Respon Kendali Terhadap Grafik TDS Pekan Kedua

| Waktu | Setpoint TDS | TDS Awal | Td (detik) | Tr (detik) | Tp (detik) | Mp (%) | Ts (detik) | Ess |
|-------|--------------|----------|------------|------------|------------|--------|------------|------|
| Pagi | 1.100 | 1.067,99 | 71,4 | 142,8 | - | - | 450 | 0,45 |
| Siang | 1.100 | 1.060,71 | 100 | 200 | - | - | 270 | 0,73 |
| Sore | 1.100 | 1.154,29 | 120 | 240 | - | - | 270 | 0,90 |

Pada tabel 4.12 didapatkan hasil pada pagi hari untuk td 71,4 detik dan untuk tr 142,8detik, pada siang hari untuk nilai td 100 detik dan tr 200 detik, dan untuk sore hari didapatkan hasil td 120 detik dan tr 240 detik.

4.4.9 Pengujian Sistem Kendali TDS Pada Pekan Ketiga

Pengujian kendali TDS pada pekan ketiga dilakukan pada pagi, siang dan sore hari. Pengujian ini dilakukan untuk mengendalikan nilai TDS pada plan akuaponik agar sesuai dengan *setpoint* yang diinginkan. *Setpoint* yang diberikan pada plan akuaponik TDS sebesar 1.100 PPM. Pekan ketiga terjadi perubahan nilai TDS pada sore hari sangat tinggi. Hasil pengujian dapat dilihat pada Gambar 4.14.



Gambar 4.14 Grafik Kendali TDS Pekan Ketiga

Grafik kendali yang ditunjukkan pada Gambar 4.14 menunjukkan hasil dari pengujian kendali TDS, pada pagi hari mendapatkan nilai TDS awal sebesar 1050,71PPM, siang hari 1061,48PPM dan sore hari 1145,91PPM. Waktu yang dibutuhkan oleh sistem untuk mencapai *setpoint* pada pagi hari adalah 84 detik, pada siang hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* pada siang hari adalah 108 detik. Waktu yang dibutuhkan pada sore hari untuk mencapai nilai

setpoint adalah 109 detik. Hasil pengujian kendali TDS pada pekan pertama terdapat beberapa nilai dari respon kendali yang dapat dilihat pada Tabel 4.13.

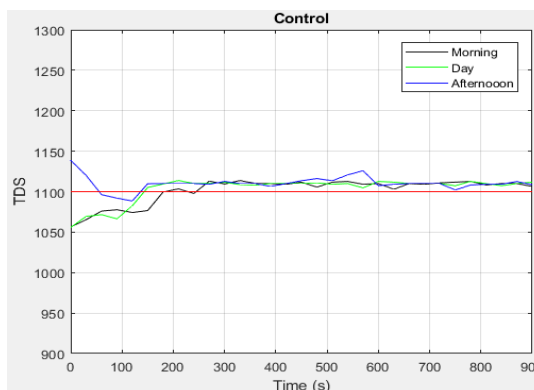
Tabel 4.13 Respon Kendali Terhadap Grafik TDS Pekan Ketiga

| Waktu | Setpoint TDS | TDS Awal | Td (detik) | Tr (detik) | Tp (detik) | Mp (%) | Ts (detik) | Ess |
|-------|--------------|----------|------------|------------|------------|--------|------------|------|
| Pagi | 1.100 | 1.050,71 | 42 | 84 | - | - | 270 | 0,90 |
| Siang | 1.100 | 1.061,48 | 54 | 108 | - | - | 210 | 0,9 |
| Sore | 1.100 | 1.145,91 | 54,5 | 109 | - | - | 300 | 1,07 |

Pada tabel 4.13 didapatkan hasil pada pagi hari untuk td 42 detik dan untuk tr 84 detik, pada siang hari untuk nilai td 54 detik dan tr 108 detik, dan untuk sore hari didapatkan hasil td 54,5 detik dan tr 109 detik.

4.4.10 Pengujian Sistem Kendali TDS Pada Pekan Keempat

Pengujian kendali TDS pada pekan keempat dilakukan pada pagi, siang dan sore hari. Pengujian ini dilakukan untuk mengendalikan nilai TDS pada plan akuaponik agar sesuai dengan *setpoint* yang diinginkan. *Setpoint* yang diberikan pada plan akuaponik TDS sebesar 1100 PPM, pada pekan keempat nilai TDS pada akuaponik mendekati pekan pertama dan kedua. Hasil pengujian dapat dilihat pada Gambar 4.15.



Gambar 4.15 Grafik Kendali TDS Pekan Keempat

Grafik kendali yang ditunjukkan pada Gambar 4.15 menunjukkan hasil dari pengujian kendali TDS, pada pagi hari mendapatkan nilai TDS awal sebesar 1.056,41PPM, siang hari 1.055,84PPM dan sore hari 1.138,67PPM. Waktu yang dibutuhkan oleh sistem untuk mencapai *setpoint* pada pagi hari adalah 184 detik, pada siang hari waktu yang dibutuhkan untuk mencapai nilai *setpoint* pada siang

hari adalah 143,2 detik. Waktu yang dibutuhkan pada sore hari untuk mencapai nilai *setpoint* adalah 55,3 detik. Hasil pengujian kendali TDS pada pekan pertama terdapat beberapa nilai dari respon kendali yang dapat dilihat pada Tabel 4.14.

Tabel 4.14 Respon Kendali Terhadap Grafik TDS Pekan Keempat

| Waktu | Setpoint TDS | TDS Awal | Td (detik) | Tr (detik) | Tp (detik) | Mp (%) | Ts (detik) | Ess |
|-------|--------------|----------|------------|------------|------------|--------|------------|------|
| Pagi | 1.100 | 1.056,41 | 92 | 184 | - | - | 330 | 1,2 |
| Siang | 1.100 | 1.055,84 | 71,6 | 143,2 | - | - | 300 | 1,05 |
| Sore | 1.100 | 1.138,67 | 27,65 | 55,3 | - | - | 300 | 1,15 |

Pada tabel 4.14 didapatkan hasil pada pagi hari untuk td 92 detik dan untuk tr 184 detik, pada siang hari untuk nilai td 71,6 detik dan tr 143,2 detik, dan untuk sore hari didapatkan hasil td 27,65 detik dan tr 55,3 detik.

4.5 Pengujian Aplikasi

Pengujian aplikasi dilakukan menggunakan android yang berfungsi sebagai antarmuka dari sistem ke pengguna, pembuatan aplikasi menggunakan *MIT app inventor 2*. Pengguna dapat menjalankan aplikasi yang sudah diinstall kemudian dapat melakukan monitoring akuaponik dan dapat melihat status *actuator* yaitu pompa nutrisi pada android. Tampilan halaman utama aplikasi dapat dilihat pada Gambar 4.16.



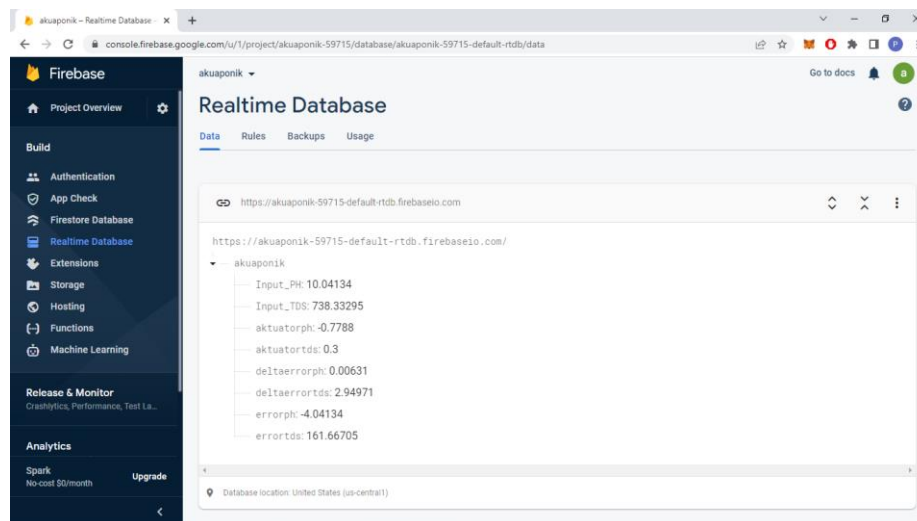
Gambar 4.16 Tampilan Halaman Utama Aplikasi

Pada gambar 4.16 tampilan halaman utama aplikasi terdapat data nilai pH dan nilai TDS terhadap air akuaponik, dan terdapat 2 tombol yaitu pH dan TDS. Masing-masing berfungsi untuk masuk ke halaman yang menampilkan data dari

masing-masing nutrisi. Data yang ditampilkan berupa data nutrisi, *error* dan *delta error* dan terdapat status *actuator* yaitu *pump up* dan *pump down*.

4.6 Pengujian Database

Pengujian *database* dilakukan dengan melihat data yang telah kirim oleh ESP 32 kedalam *database*, data dapat dilihat dengan cara membuka halaman *database* pada <https://akuaponik-59715-default-rtbd.firebaseio.com/>, data yang telah disimpan diberinama akuaponik. Gambar 4.17 merupakan tampilan dari penyimpanan *database*, data yang ditampilkan yaitu *input pH*, *input TDS*, *actuator pH*, *actuator TDS*, *deltaerror pH*, *deltaerror TDS*, *error pH* dan *error TDS*. Dari hasil pengamatan ini dapat disimpulkan bawasanya *server* berhasil menyimpan data sensor.



Gambar 4.17 Tampilan Penyimpanan Database

Pada gambar 4.17 adalah tampilan penyimpanan real time data dari firebase. Data dari firebase akan ditampilkan pada aplikasi yang sudah dibuat menggunakan *MIT App Inventor*.

4.7 Laju Pertumbuhan Akuaponik

Pengujian sistem kendali pH dan TDS air akuaponik menggunakan metode logika Fuzzy menghasilkan pertumbuhan terhadap ikan mas dan tanaman pakcoy. Laju pertumbuhan pada ikan mas dan tanaman pakcoy dilakukan dengan

mengukur panjang ikan mas dan tanaman pakcoy. Laju pertumbuhan akuaponik pada tanaman pakcoy dan ikan mas dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil Pertumbuhan Akuaponik

| Plant Akuaponik | Pertumbuhan Ikan dan Tanaman (cm) | | | |
|--------------------|-----------------------------------|-----------|------------|------------|
| | Hari ke-1 | Hari ke-7 | Hari ke-14 | Hari ke-30 |
| Ikan 1 | 6 | 7 | 8 | 10 |
| Ikan 2 | 7 | 8 | 10 | 12 |
| Ikan 3 | 7 | 8 | 10 | 11 |
| Tanaman 1 | 4 | 5 | 7 | 11 |
| Tanaman 2 | 4 | 5 | 8 | 12 |
| Tanaman 3 | 5 | 6 | 8 | 12 |

Pada tabel 4.15 menunjukkan pertumbuhan akuaponik pada tanaman dan ikan. Pertumbuhan rata-rata pada ikan mas pada plan akuaponik yang sudah dibuat yaitu 3 cm sampai 4 cm dalam 30 hari. Sedangkan rata-rata pertumbuhan pada tanaman pakcoy pada plan akuaponik yang sudah dibuat adalah 7 cm dalam 30 hari. Sistem dapat mengendalikan nutrisi pH dan TDS mendekati kebutuhan .

BAB V

PENUTUP

5.1 Kesimpulan

Dari pembahasan dan pengujian yang dilakukan pada sistem kendali pH dan TDS air akuaponik menggunakan metode logika Fuzzy, dapat disimpulkan bahwa.

1. Sistem kendali pH dan TDS pada akuaponik telah berhasil dibuat dengan *setpoint* 6,5 pH dan TDS sebesar 1100 PPM.
2. Sistem kendali pH dan TDS pada air akuaponik menggunakan metode logika Fuzzy berhasil mendapatkan performa terbaik untuk pH dengan *settling time* 300 detik dengan *error steady-state* sebesar 0,31% dan performa terbaik untuk TDS dengan *settling time* 210 detik dengan *error steady-state* sebesar 0,68%.

5.2 Saran

Adapun saran yang diajukan agar dapat menjadi masukan dalam pengembangan sistem ini adalah:

1. Menambahkan pilihan metode sistem kendali pada akuaponik, agar dapat menghasilkan pengendalian yang lebih baik.
2. Menambahkan variabel nutrisi pada kualitas air lainnya supaya lebih lengkap dan lebih optimal.

DAFTAR PUSTAKA

- [1] Burlian A., Y. Rahmanto., S. Samsugi., and A. Sucipto, “Sistem Kendali Otomatis Pada Akuaponik Berbasis Mikrokontroler Arduino Uno R3,” Universitas Teknologi Indonesia, Lampung, 2021.
- [2] Monica O. “Pemanfaatan Sistem Resirkulasi Limbah Budidaya Ikan Lele (*Clarias Batrachus*) Dengan Sistem Akuaponik Tanaman Sawi Pakcoy (*Brassica Rapa L.*) Terhadap Padat Tebar Ikan Nila Merah (*Oreochromis Niloticus*),” Sumatra: Universitas Sumatra Utara, 2020.
- [3] Krastanova M., I Sirakov., S. I Kirilova., D. Yarkov, and Pety, “*Aquaponic systems: biological and technological Parameters*,” Biotechnology & Biotechnological Equipment, 2022.
- [4] Sotyohadi, W., Surya., and I. Komang, “Perancangan Pengatur Kandungan Tds Dan Ph Pada Larutan Nutrisi Hidroponik Menggunakan Metode Fuzzy Logic,” Malang: Institut Teknologi Nasional Malang.
- [5] Rahmanto Y., A. Rifaini., S. Samsugi., Sampurna, Riskiono, “Sistem Monitoring pH Air Pada Aquaponik Menggunakan Mikrokontroler Arduino Uno,” Universitas Teknologi Indonesia, 2020.
- [6] Irchasandro S., Sungkono, and Agus P. “Sistem Otomatisasi Perawatan Hidroponik Padatanaman Kangkung Berbasis Iot,” Malang: Politeknik Negri Malang, 2021.
- [7] Jiwa K., I. Wirama., and I. Wayan. “Implementasi *Iot* Cerdas Berbasis *Inference* Fuzzy Tsukamoto Pada Pemantauan Kadar Ph Dan Ketinggian Air Dalam Akuaponik (*Smart Iot Based Inference Fuzzy Tsukamoto Implementation On Monitoring Ph Levels And Water Height In Aquaponics*),” Mataram: Universitas Mataram. 2019.
- [8] Megawati D., Masykuroh K., Kurnianto D., and Kunci K. “Rancang Bangun Sistem Monitoring Ph Dan Suhu Air Pada Akuaponik Berbasis *Internet Of Thing (Iot) Design Of Monitoring System For Ph And Water Temperature In Aquaponic Base On Internet Of Thing (Iot)*,” Purwokerto: Telkom Purwokerto, 2020.
- [9] Bracino A. A., S. Ronnie., Grace D., Rhay R. P, and S. Dadios, “*Fuzzy Logic-Based Automated pH and Temperature Control System for Biofilter in Smart Aquaponics*,” *Journal of Computational Innovations and Engineering Applications*, 2020.
- [10] Ronnie S., Lauguico S. C., Jonnel D., Alejandrino., Bandala A. A., Sybingco E., Vicerra R. R. P., Dadios P. E, and Cuello J, “*Adaptive Fertigation System Using Hybrid Vision-Based Lettuce Phenotyping and Fuzzy Logic Valve Controller Towards Sustainable Aquaponics*,” *Concepcion*, 2021.
- [11] Ronnie S., Lauguico S. C., James M. P., Loresco M., Valemzuela I. C., Dadios E. P, and Bandala A. A, “*Automated Nutrient Solution Control*

System Using Embedd Fuzzy Logic Controller for Smart Nutrient Film Technique Auaponics,” Journal of Computational Innovations and Engineering Applications, 2020.

- [12] Jonathan S., Priskila K., Angelicha T., and Salaki R. “Penentuan Beasiswa Dengan Metode Fuzzy Tsukamoto Berbasis Web Scholarship Determination Using Web Based Fuzzy Tsukamoto Method,”. Universitas Sam Raitulangi. 2022.
- [13] Bijan D., Imam M., and Soleha. “Himpunan Fuzzy dan *Rough Sets*,”. Institut Teknologi Sepuluh November. 2021.
- [14] Axmadjonov M. F, and Mirzaraximov M. A, “*Firestore in Real Time Systems Based on Client Server Technology*,” *Scientific Journal Impact Factor*, 2021
- [15] Defa R., M. Ramdhani., R. Priramadhi., and B. Aprillia. “*Automatic Controlling System and Iot Based Monitoring For Ph Rate On The Aquaponics System*,”. Bandung: Telkom Bandung. 2019.
- [16] Maniagasi R., S. Tumembouw., and Y. Mundeng. “Analisis Kualitas Fisika Kimia Air di Areal Budidaya Ikan Danau Tondano Provinsi Sulawesi Utara (*Analysis of Physical-Chemical Quality Of Waters At Aquaculture Area In Lake Tondano, North Sulawesi Province*),”. Sulawesi Utara. 2013.
- [17] Faysal M.,. “*Assessment Of Ph And Total Dissolved Substances (Tds) In The Commercially Available Bottled Drinking Water*,”. Bangladesh. 2017
- [18] Lucien M., and L. Amanda. “Analisa Kualitas Air Alkalinitas Dan Kesadahan (*Hardness*) Pada Pembesaran Udang Putih (*Litopenaeus Vannamei*) Di Laboratorium Animal Health Service Binaan Pt. Central Proteina Prima Tbk. Medan,”. Sekolah Tinggi Perikanan Sibolga. 2016.
- [19] Adi F., B. Setya., and A. Manan. “Pengaruh Pemberian Probiotik Berbeda Dalam Sistem Akuaponik Terhadap Fcr (*Feed Conversion Ratio*) Dan Biomassa Ikan Lele (*Clarias Sp.*) *Effect Addition Of Different Probiotics In Aquaponics System Toward Fcr (Feed Conversion Ratio) And Biomass Of Catfish (Clarias Sp.)*,”. Surabaya: Universitas Airlangga. 2018.
- [20] Wahyu P. “*Removal Klorida, Tds Dan Besi Pada Air Payau Melalui Penukar Ion Dan Filtrasi Campuran Zeolit Aktif Dengan Karbon Aktif*,”. Surabaya: Universitas Adi Buana. 2020.
- [21] Wijayanti M., H. Khotimah., A. Sasanti., S. Dwinanti., D. Madyasta., and A. Rarassari. “Pemeliharaan Ikan Nila (*Oreochromis Niloticus*) Dengan Sistem Akuaponik Di Desa Karang Endah, Kecamatan Gelumbang, Kabupaten Muara Enim Sumatra Selatan,” Sumatra: Universitas Sriwijaya. 2019.
- [22] Prahesti, “Penggunaan Sistem Akuaponik Dengan Jenis Tanaman Yang Berbeda Terhadap Pertumbuhan Dan Kelangsungan Hidup Ikan Mas (*Cyprinus Carpio*),”. Gresik: Universitas Muhammadiyah Gresik. 2019.

- [23] Ali B., A. Manggala., Hamdani., and Hastuti “Sistem Kendali Kekерuhan Dan Ph Air Kolam Budidaya Ikan Nila,”. 2021.
- [24] Bidarti A., M. Eka., M. Indri., M. Erni., and M. Arbi. “Percontohan Metode Aquaponik Dalam Budidaya Bayam (*Amaranthus Tricolor*) Sebagai Sumber Pendapatan Sampingan Bagi Ibu-Ibu Rumah Tangga Di Kecamatan Gandus Kota Palembang,” 2017.
- [25] Kukuћ S., Ibrahim., and R. Rahmadewi. “Sistem Kontrol Ph Dan Nutrisi Otomatis Pada Tanaman Hidroponik Bayam Berbasis Fuzzy Logic,”. 2021.
- [26] Wahyuningsih A., S. Fajriani., and N. Aini. “Komposisi Nutrisi Dan Media Tanam Terhadap Pertumbuhan Dan Hasil Tanaman Pakcoy (*Brassica Rapa L.*) Sistem Hidroponik *The Nutrition And Growth Media Composition On The Growth And Yield Of Pakcoy (Brassica Rapa L.) Using Hydroponics System,*”. 2017.
- [27] Wati D., and W. Sholihah. “Pengontrol Ph Dan Nutrisi Tanaman Selada Pada Hidroponik Sistem Nft Berbasis Arduino,”. 2021.
- [28] Pangaribowo T. “Perancangan Simulasi Kendali *valve* dengan Algoritma logika Fuzzy Menggunakan Bahasa *Visual Basic,*”. 2017.
- [29] Utama Y., W. Yonatan., S. Tri., and K. Hendra. “Sistem Pengaturan Dasar,”. 2018.
- [30] Nurkarima I., H. Priyatman., and B. Kurniawan. “Pemodelan Sistem Level Air Head Tank Menggunakan Pengendali PID Pada Pembangkit Listrik Tenaga Mini Hidro Merasap,”. 2019.
- [31] Wiharja U., and G. Herlambang. “Sistem Pengendali Kecepatan Putar Motor Dc Dengan Arduino Berbasis Labview,”. 2019.
- [32] Waspada I., and Sutikno. “Perbandingan Metode Defuzzifikasi Sistem Kendali Logika Fuzzy Model Mamdani Pada Motor Dc,”. 2020.
- [33] Setiadi B., and Kartono W. “Desain Dan Implementasi Fuzzy Logic Pada Sistem Pengendalian Beban Ac Berbasis *Mikrokontroler Design And Implementation Fuzzy Logic For Ac Load Control System Based On Microcontroller,*”. 2019.
- [34] Julianto Y. “Sistem Pengendalian Pompa Filter Pada Aquarium Menggunakan *Logika* Fuzzy Berbasis Arduino Skripsi Teknik Elektro Konsentrasi Teknik Kontrol,” 2020.
- [35] Kamsyakawuni A. “Aplikasi Sistem Pakar Untuk Diagnosa Penyakit Hipertiroid Dengan Metode Inferensi Fuzzy Mamdani,” 2012.
- [36] Widaningsih S. “Analisis Perbandingan Metode Fuzzy Tsukamoto, Mamdani Dan Sugeno Dalam Pengambilan Keputusan Penentuan Jumlah Distribusi Raskin Di Bulog Sub. Divisi *Regional (Divre) Cianju,r*”. 2017.

- [37] Haerani E. Analisa “Kendali Logika Fuzzy Dengan Metode Defuzzifikasi *Coa (Center Of Area)*, *Bisektor*, *Mom (Mean Of Maximum)*, *Lom (Largest Of Maximum)*, Dan *Som (Smallest Of Maximum)*,”. 2020.
- [38] Adella A., M. Fardika., F. Taufiqurrahman., and A. Baso “Pintu Otomatis Berbasis *Ultrasonic Internet Of Things*,” 2020
- [39] Kinasih R. “Rancang Bangun Alat Pengukur Kekeruhan Air Berbasis Mikrokontroler Atmega8535 Dengan Menggunakan Sensor Photodiode,”. 2020.
- [40] Selviyani S. “*Dc Current, And Voltage Monitoring System Design With Microcontroller Atmega32 On Wind Turbine Horizontal Axis*,”. 2017.
- [41] Deska A., F. Supegina., and T. Maya. ”Sistem Kontrol Dan Monitor Suplai Nutrisi Hidroponik Sistem *Deep Flow Technique (Dft)* Berbasis Arduino Nodemcu Dan Aplikasi Android,”. 2019.
- [42] Syafiqoh U. “Pengembangan *Wireless Sensor Network* Berbasis *Internet Of Things* Untuk Sistem Pemantauan Kualitas Air Dan Tanah Pertanian,”. 2018.
- [43] Kusumah H., and R. Pradana, “Penerapan *Trainer Interfacing Mikrokontroler Dan Internet Of Things* Berbasis Esp32 Pada Mata Kuliah *Interfacing*,”, 2018.
- [44] Samosir A. “Rancang Bangun Catu Daya Digital Menggunakan Buck Converter Berbasis Mikrokontroler Arduino,”. 2017.
- [45] Saleh M., and M. Haryanti. “Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay,” 2017.
- [46] Nugrahanto I. “Pembuatan Water Level Sebagai Pengendali Water Pumpotomatis Berbasis Transistor,”. 2017.
- [47] Suha L. “Rancang Bangunalat Penyiramanotomatis Padapertanian Vertikalberbasisarduinomenggunakan Algoritma Adaline,”. 2018.
- [48] Putri R. “Sistem Informasi Akuntansi-Pengaplikasian Dan Implementasi Konsep Basis Data Relasional Pada Sistem Pelaporan Dan Buku Besar,”. 2019
- [49] Mahali M. “*Smart Door Locks Based on Internet of Things Concept With Mobile Backend As A Service*,” 2017.
- [50] Tahtawi A., and R. Kurniawan. “*Ph Control for Deep Flow Technique Hydroponic Iot Systems Based on Fuzzy Logic Controller*,” 2020.
- [51] Pratikel D. “Respon Pertumbuhan Berbagai Jenis Tanaman Selada (*Lactuca Sativa*) Menggunakan Sistem Akuaponik Dengan Padat Tebar Berbeda Ikan Nila Merah (*Oreochromis Sp*) Pada Teknologi Bioflok,”. 2021.
- [52] Sholihah A., T. Tohir., and A. Tahtawi. “Kendali Tds Nutrisi Hidroponik *Deep Flow Technique*berbasis *Iot* MenggunakanFuzzy Logic,”, 2021.

- [53] Fikri M., A. Musthafa., and F. Reza. “*Design And Build Smart Aquascape Based on Ph and Tds With Iot System Using Fuzzy Logic*”*rancang Bangun Smart Aquascape Berd/asarkan Ph Dan Tds Dengan Sistem Iot Menggunakan Logika Fuzzy,*” 2021.
- [54] Kurnia S. “Implementasi Sistem Pengaturan Ph Otomatis Terhadap Air Akuarium Ikan Gurami Dengan Media Aquaponik Menggunakan Fuzzy Logic Control,”. 2021.

LAMPIRAN

Lampiran A Listing Code

```
#include <EEPROM.h>
#include "Sensor_TDS.h"
#include "fis_header.h"
#include <WiFi.h>
#include <FirebaseESP32.h>
#define WIFI_SSID "kepo1"
#define WIFI_PASSWORD "kepoajalu"
#define FIREBASE_HOST "https://akuaponik-59715-default-
rtadb.firebaseio.com"
#define FIREBASE_KEY "f9sSCQ5bVxssszmNhYPkFrN9Oo7f8MfFQtmqS7ey"
#define TDS_SENSOR_PIN 32
#define PH_SENSOR_PIN 33 //pH meter Analog output to
Arduino Analog Input 0
#define Offset 2.2 //deviation compensate
#define LED 13
#define ArrayLenth 40 //times of collection
int pHArray[ArrayLenth]; //Store the average value of the sensor
feedback
int pHArrayIndex = 0;
float setPointpH = 6;
float errorpH;
float deltaerrorpH;
float errorpHsebelum;
float setPointtds = 900;
float errortds;
float deltaerrortds;
float errortdssebelum;
float f_pH;
float f_tdss;
float Input_PH;
float Input_TDS;
float temperature = 25, tdsValue = 0;
float read_ph;
float read_tds;

Sensor_TDS tds_sensor;
FirebaseData firebaseData;
FirebaseJson firebaseJson;
int Relay1 = 3; //Pompa pH up
int Relay2 = 21; //Pompa pH down
int Relay3 = 19; //Pompa tds up
int Relay4 = 18; //Pompa tds down
unsigned long ulangi;
int waktu = 20000;
// Number of inputs to the Fuzzy inference system
const int fis_gcI = 4;
// Number of outputs to the Fuzzy inference system
const int fis_gcO = 2;
// Number of rules to the Fuzzy inference system
const int fis_gcR = 50;
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
// Setup routine runs once when you press reset:
void setup()
```

```

{
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Firebase.begin(FIREBASE_HOST, FIREBASE_KEY);
  Firebase.reconnectWiFi(true);

  //Set database read timeout to 1 minute (max 15 minutes)
  Firebase.setReadTimeout(firebaseData, 1000 * 60);
  //tiny, small, medium, large and unlimited.
  //Size and its write timeout e.g. tiny (1s), small (10s), medium
(30s) and large (60s).
  Firebase.setwriteSizeLimit(firebaseData, "small");
  Serial.println("Firebase Connected.");
  tds_sensor.setPin(TDS_SENSOR_PIN);
  tds_sensor.setAref(3.3);
  tds_sensor.setAdcRange(4096);
  tds_sensor.setKvalueAddress(16);
  tds_sensor.begin();
  pinMode(33, INPUT);
  pinMode(Relay1, OUTPUT);
  pinMode(Relay2, OUTPUT);
  pinMode(Relay3, OUTPUT);
  pinMode(Relay4, OUTPUT);
}
void hitungeror()
{
  //Rumus Fuzzy pH
  Serial.println("=====");
  Serial.print("pH = ");
  Serial.println(Input_PH);
  errorpH = setPointpH - Input_PH;
  Serial.print("errorpH = ");
  Serial.println(errorpH);
  deltaerrorpH = errorpH - errorpHsebelum;
  Serial.print("deltaerrorpH = ");
  Serial.println(deltaerrorpH);
  errorpHsebelum = errorpH;
  Serial.print("errorpHsebelum = ");
  Serial.println(errorpHsebelum);
  Serial.println("=====");
  //Rumus Fuzzy tds
  Serial.print("tds = ");
  Serial.println(Input_TDS);
  errortds = setPointtds - Input_TDS;
  Serial.print("errortds = ");
  Serial.println(errortds);
  deltaerrortds = errortds - errortdssebelum;
  Serial.print("deltaerrortds = ");
  Serial.println(deltaerrortds);
}

```

```

    errortdssebelum = errortds;
    Serial.print("errortdssebelum = ");
    Serial.println(errortdssebelum);
    Serial.println("=====");
    Serial.print("g_fisOutput[0]= ");
    Serial.println(g_fisOutput[0]);
    Serial.print("g_fisOutput[1]= ");
    Serial.println(g_fisOutput[1]);
}
/*
void serial_Logger()
{
Serial.println(F("=== Input ==="));
  Serial.print(F("PH: "));
  Serial.print(Input_PH);
  Serial.print(F("\t"));
  Serial.print(F("TDS: "));
  Serial.println(Input_TDS);
  Serial.println(F("=== Output ==="));
  Serial.print(F("PH_Motor: "));
  Serial.print(Output_Motor_PH);
  Serial.print(F("\t"));
  Serial.print(F("TDS_Motor: "));
  Serial.println(Output_Motor_TDS);
  Serial.println();
  Serial.println(F("=== Status ==="));
  Serial.print(F("ErrorpH : "));
  Serial.print(F(errorpH));
  Serial.print(F("\t\t| "));
  Serial.print(F("DeltaErrorpH: "));
  Serial.print(deltaerrorpH);
  Serial.print(F("ErrorTDS : "));
  Serial.print(F(errortds));
  Serial.print(F("\t| "));
  Serial.print(F("DeltaErrorTDS"));
  Serial.print(deltaerrortds);
  Serial.println();
  Serial.println();
}
*/
// Loop routine runs over and over again forever:
void loop(void)
{
  unsigned long currentMillis = millis();
  if ((currentMillis - ulangi) > waktu) {
    ulangi = millis();
    static float pHValue, voltage;
    {
      pHArray[pHArrayIndex++] = analogRead(PH_SENSOR_PIN);
      if (pHArrayIndex == ArrayLenth)pHArrayIndex = 0;
      voltage = avergearray(pHArray, ArrayLenth) * 3.3 / 4096;
      pHValue = 3.5 * voltage + Offset;
      Input_PH = pHValue;
    }
    {
      tds_sensor.setTemperature(temperature); // set the
temperature and execute temperature compensation
      tds_sensor.update(); //sample and calculate
      Input_TDS = tds_sensor.getTdsValue();
    }
  }
}

```

```

    }
    hitungeror();
    firebase_logger();
    keputusan();
}
}
void keputusan ()
{
    // Read Input: ErrorPh
    g_fisInput[0] = errorpH;
    // Read Input: DeltaErrorPh
    g_fisInput[1] = deltaerrorpH;
    // Read Input: ErrorTds
    g_fisInput[2] = errortds;
    // Read Input: DeltaErrorTds
    g_fisInput[3] = deltaerrortds;
    g_fisOutput[0] = 0;
    g_fisOutput[1] = 0;
    fis_evaluate();
    if ( (act(g_fisOutput[0]) == 0) && (act(g_fisOutput[1]) == 0) ) {
        off(3); //Semua RELAY Mati
        off(21);
        off(19);
        off(18);
    }
    if ( (act(g_fisOutput[0]) == 0) && (act(g_fisOutput[1]) == 1) ) {
        off(3);
        off(21);
        on(19), delay(5000), off(19);
        off(18);
    }
    if ( (act(g_fisOutput[0]) == 0) && (act(g_fisOutput[1]) == 2) ) {
        off(3);
        off(21);
        on(19), delay(7000), off(19);
        off(18);
    }
    if ( (act(g_fisOutput[0]) == 0) && (act(g_fisOutput[1]) == 3) ) {
        off(3);
        off(21);
        off(19);
        on(18), delay(5000), off(18);
    }
    if ( (act(g_fisOutput[0]) == 0) && (act(g_fisOutput[1]) == 4) ) {
        off(3);
        off(21);
        off(19);
        on(18), delay(7000), off(18);
    }
}
if ( (act(g_fisOutput[0]) == 1) && (act(g_fisOutput[1]) == 0) ) {
    on(3), delay(5000), off(3);
    off(21);
    off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 1) && (act(g_fisOutput[1]) == 1) ) {
    on(3), delay(5000), off(3);
    off(21);
}

```



```

    on(19), delay(5000), off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 1) && (act(g_fisOutput[1]) == 2)) {
    on(3), delay(5000), off(3);
    off(21);
    on(19), delay(7000), off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 1) && (act(g_fisOutput[1]) == 3)) {
    on(3), delay(5000), off(3);
    off(21);
    off(19);
    on(18), delay(5000), off(18);
}
if ( (act(g_fisOutput[0]) == 1) && (act(g_fisOutput[1]) == 4)) {
    on(3), delay(5000), off(3);
    off(21);
    off(19);
    on(18), delay(7000), off(18);
}
if ( (act(g_fisOutput[0]) == 2) && (act(g_fisOutput[1]) == 0)) {
    on(3), delay(7000), off(3);
    off(21);
    off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 2) && (act(g_fisOutput[1]) == 1)) {
    on(3), delay(7000), off(3);
    off(21);
    on(19), delay(5000), off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 2) && (act(g_fisOutput[1]) == 2)) {
    on(3), delay(7000), off(3);
    off(21);
    on(19), delay(7000), off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 2) && (act(g_fisOutput[1]) == 3)) {
    on(3), delay(7000), off(3);
    off(21);
    off(19);
    on(18), delay(5000), off(18);
}
if ( (act(g_fisOutput[0]) == 2) && (act(g_fisOutput[1]) == 4)) {
    on(3), delay(7000), off(3);
    off(21);
    off(19);
    on(18), delay(7000), off(18);
}
if ( (act(g_fisOutput[0]) == 3) && (act(g_fisOutput[1]) == 0)) {
    off(3);
    on(21), delay(5000), off(21);
    off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 3) && (act(g_fisOutput[1]) == 1)) {
    off(3);
}

```

```

    on(21), delay(5000), off(21);
    on(19), delay(5000), off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 3) && (act(g_fisOutput[1]) == 2)) {
    off(3);
    on(21), delay(5000), off(21);
    on(19), delay(7000), off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 3) && (act(g_fisOutput[1]) == 3)) {
    off(3);
    on(21), delay(5000), off(21);
    off(19);
    on(18), delay(5000), off(18);
}
if ( (act(g_fisOutput[0]) == 3) && (act(g_fisOutput[1]) == 4)) {
    off(3);
    on(21), delay(5000), off(21);
    off(19);
    on(18), delay(7000), off(18);
}
if ( (act(g_fisOutput[0]) == 4) && (act(g_fisOutput[1]) == 0)) {
    off(3);
    on(21), delay(7000), off(21);
    off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 4) && (act(g_fisOutput[1]) == 1)) {
    off(3);
    on(21), delay(7000), off(21);
    on(19), delay(5000), off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 4) && (act(g_fisOutput[1]) == 2)) {
    off(3);
    on(21), delay(7000), off(21);
    on(19), delay(7000), off(19);
    off(18);
}
if ( (act(g_fisOutput[0]) == 4) && (act(g_fisOutput[1]) == 3)) {
    off(3);
    on(21), delay(7000), off(21);
    off(19);
    on(18), delay(5000), off(18);
}
if ( (act(g_fisOutput[0]) == 4) && (act(g_fisOutput[1]) == 4)) {
    off(3);
    on(21), delay(7000), off(21);
    off(19);
    on(18), delay(7000), off(18);
}
}
}
void on(int p) {
    digitalWrite(p, LOW);
}
void off(int p) {
    digitalWrite(p, HIGH);
}
}

```

```

bool nol(float n) {
    if (n < 0.001 && n > -0.001) return true;
    else false;
}
int act(float n)
{
    // { 0 = MATI | 1 = Relay 3&4 | 2 = Relay 5&6 }
    if(n >= -0.1 && n <= 0.1) return 0;
    if(n <= -0.5) return 2;
    if(n >= -0.5 && n <= -0.1) return 1;
    if(n >= 0.1 && n <= 0.5) return 3;
    if(n >= 0.5) return 4;
}
//*****
// Support functions for Fuzzy Inference System
//*****
// Trapezoidal Member Function
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d -
x) / (d - c)) : 0)));
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x -
a) / (b - a)) : 0)));
    return (FIS_TYPE) min(t1, t2);
}
FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}
FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}
FIS_TYPE fis_array_operation(FIS_TYPE *array, int size,
_FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;

    if (size == 0) return ret;
    if (size == 1) return array[0];

    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);
    }
    return ret;
}
//*****
// Data for Fuzzy Inference System
//*****
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =

```

```

{
    fis_trapmf
};
// Count of member function for each Input
int fis_gIMFCount[] = { 5, 5, 5, 5 };
// Count of member function for each Output
int fis_gOMFCount[] = { 5, 5 };
// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { -4, -3, -1, 0 };
FIS_TYPE fis_gMFI0Coeff2[] = { -1, 0, 0, 1 };
FIS_TYPE fis_gMFI0Coeff3[] = { 3, 4, 6, 6 };
FIS_TYPE fis_gMFI0Coeff4[] = { -6, -6, -4, -3 };
FIS_TYPE fis_gMFI0Coeff5[] = { 0, 1, 3, 4 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2,
fis_gMFI0Coeff3, fis_gMFI0Coeff4, fis_gMFI0Coeff5 };
FIS_TYPE fis_gMFI1Coeff1[] = { -1, -1, -0.6, -0.4 };
FIS_TYPE fis_gMFI1Coeff2[] = { -0.1, 0, 0, 0.1 };
FIS_TYPE fis_gMFI1Coeff3[] = { 0.4, 0.6, 1, 1 };
FIS_TYPE fis_gMFI1Coeff4[] = { -0.5947, -0.3947, -0.1947, 0.005291
};
FIS_TYPE fis_gMFI1Coeff5[] = { 0, 0.2, 0.4, 0.6 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2,
fis_gMFI1Coeff3, fis_gMFI1Coeff4, fis_gMFI1Coeff5 };
FIS_TYPE fis_gMFI2Coeff1[] = { -1100, -1100, -733.3, -488.9 };
FIS_TYPE fis_gMFI2Coeff2[] = { -122.2, 0, 0, 122.2 };
FIS_TYPE fis_gMFI2Coeff3[] = { 488.9, 733.3, 1100, 1100 };
FIS_TYPE fis_gMFI2Coeff4[] = { -733.3, -488.9, -244.4, 0 };
FIS_TYPE fis_gMFI2Coeff5[] = { 0, 244.4, 488.9, 733.3 };
FIS_TYPE* fis_gMFI2Coeff[] = { fis_gMFI2Coeff1, fis_gMFI2Coeff2,
fis_gMFI2Coeff3, fis_gMFI2Coeff4, fis_gMFI2Coeff5 };
FIS_TYPE fis_gMFI3Coeff1[] = { -100, -100, -60, -40 };
FIS_TYPE fis_gMFI3Coeff2[] = { -10, 0, 0, 10 };
FIS_TYPE fis_gMFI3Coeff3[] = { 40, 60, 100, 100 };
FIS_TYPE fis_gMFI3Coeff4[] = { -60, -40, -20, 0 };
FIS_TYPE fis_gMFI3Coeff5[] = { 0, 20, 40, 60 };
FIS_TYPE* fis_gMFI3Coeff[] = { fis_gMFI3Coeff1, fis_gMFI3Coeff2,
fis_gMFI3Coeff3, fis_gMFI3Coeff4, fis_gMFI3Coeff5 };
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff,
fis_gMFI2Coeff, fis_gMFI3Coeff };
// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { -1, -1, -0.6, -0.5 };
FIS_TYPE fis_gMFO0Coeff2[] = { -0.1, 0, 0, 0.1 };
FIS_TYPE fis_gMFO0Coeff3[] = { 0.5, 0.6, 1, 1 };
FIS_TYPE fis_gMFO0Coeff4[] = { -0.5, -0.4, -0.2, -0.1 };
FIS_TYPE fis_gMFO0Coeff5[] = { 0.1, 0.2, 0.4, 0.5 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
fis_gMFO0Coeff3, fis_gMFO0Coeff4, fis_gMFO0Coeff5 };
FIS_TYPE fis_gMFO1Coeff1[] = { -1, -1, -0.6, -0.5 };
FIS_TYPE fis_gMFO1Coeff2[] = { -0.1, 0, 0, 0.1 };
FIS_TYPE fis_gMFO1Coeff3[] = { 0.5, 0.6, 1, 1 };
FIS_TYPE fis_gMFO1Coeff4[] = { -0.5, -0.4, -0.2, -0.1 };
FIS_TYPE fis_gMFO1Coeff5[] = { 0.1, 0.2, 0.4, 0.5 };
FIS_TYPE* fis_gMFO1Coeff[] = { fis_gMFO1Coeff1, fis_gMFO1Coeff2,
fis_gMFO1Coeff3, fis_gMFO1Coeff4, fis_gMFO1Coeff5 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff, fis_gMFO1Coeff };
// Input membership function set
int fis_gMFI0[] = { 0, 0, 0, 0, 0 };
int fis_gMFI1[] = { 0, 0, 0, 0, 0 };
int fis_gMFI2[] = { 0, 0, 0, 0, 0 };

```



```

int fis_gRI44[] = { 0, 0, 5, 5 };
int fis_gRI45[] = { 0, 0, 3, 1 };
int fis_gRI46[] = { 0, 0, 3, 4 };
int fis_gRI47[] = { 0, 0, 3, 2 };
int fis_gRI48[] = { 0, 0, 3, 5 };
int fis_gRI49[] = { 0, 0, 3, 3 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3,
fis_gRI4, fis_gRI5, fis_gRI6, fis_gRI7, fis_gRI8, fis_gRI9,
fis_gRI10, fis_gRI11, fis_gRI12, fis_gRI13, fis_gRI14, fis_gRI15,
fis_gRI16, fis_gRI17, fis_gRI18, fis_gRI19, fis_gRI20, fis_gRI21,
fis_gRI22, fis_gRI23, fis_gRI24, fis_gRI25, fis_gRI26, fis_gRI27,
fis_gRI28, fis_gRI29, fis_gRI30, fis_gRI31, fis_gRI32, fis_gRI33,
fis_gRI34, fis_gRI35, fis_gRI36, fis_gRI37, fis_gRI38, fis_gRI39,
fis_gRI40, fis_gRI41, fis_gRI42, fis_gRI43, fis_gRI44, fis_gRI45,
fis_gRI46, fis_gRI47, fis_gRI48, fis_gRI49 };
// Rule Outputs
int fis_gRO0[] = { 1, 0 };
int fis_gRO1[] = { 1, 0 };
int fis_gRO2[] = { 1, 0 };
int fis_gRO3[] = { 1, 0 };
int fis_gRO4[] = { 1, 0 };
int fis_gRO5[] = { 4, 0 };
int fis_gRO6[] = { 4, 0 };
int fis_gRO7[] = { 4, 0 };
int fis_gRO8[] = { 4, 0 };
int fis_gRO9[] = { 4, 0 };
int fis_gRO10[] = { 2, 0 };
int fis_gRO11[] = { 2, 0 };
int fis_gRO12[] = { 2, 0 };
int fis_gRO13[] = { 2, 0 };
int fis_gRO14[] = { 2, 0 };
int fis_gRO15[] = { 5, 0 };
int fis_gRO16[] = { 5, 0 };
int fis_gRO17[] = { 5, 0 };
int fis_gRO18[] = { 5, 0 };
int fis_gRO19[] = { 5, 0 };
int fis_gRO20[] = { 3, 0 };
int fis_gRO21[] = { 3, 0 };
int fis_gRO22[] = { 3, 0 };
int fis_gRO23[] = { 3, 0 };
int fis_gRO24[] = { 3, 0 };
int fis_gRO25[] = { 0, 1 };
int fis_gRO26[] = { 0, 1 };
int fis_gRO27[] = { 0, 1 };
int fis_gRO28[] = { 0, 1 };
int fis_gRO29[] = { 0, 1 };
int fis_gRO30[] = { 0, 4 };
int fis_gRO31[] = { 0, 4 };
int fis_gRO32[] = { 0, 4 };
int fis_gRO33[] = { 0, 4 };
int fis_gRO34[] = { 0, 4 };
int fis_gRO35[] = { 0, 2 };
int fis_gRO36[] = { 0, 2 };
int fis_gRO37[] = { 0, 2 };
int fis_gRO38[] = { 0, 2 };
int fis_gRO39[] = { 0, 2 };
int fis_gRO40[] = { 0, 5 };
int fis_gRO41[] = { 0, 5 };
int fis_gRO42[] = { 0, 5 };

```

```

int fis_gRO43[] = { 0, 5 };
int fis_gRO44[] = { 0, 5 };
int fis_gRO45[] = { 0, 3 };
int fis_gRO46[] = { 0, 3 };
int fis_gRO47[] = { 0, 3 };
int fis_gRO48[] = { 0, 3 };
int fis_gRO49[] = { 0, 3 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3,
fis_gRO4, fis_gRO5, fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9,
fis_gRO10, fis_gRO11, fis_gRO12, fis_gRO13, fis_gRO14, fis_gRO15,
fis_gRO16, fis_gRO17, fis_gRO18, fis_gRO19, fis_gRO20, fis_gRO21,
fis_gRO22, fis_gRO23, fis_gRO24, fis_gRO25, fis_gRO26, fis_gRO27,
fis_gRO28, fis_gRO29, fis_gRO30, fis_gRO31, fis_gRO32, fis_gRO33,
fis_gRO34, fis_gRO35, fis_gRO36, fis_gRO37, fis_gRO38, fis_gRO39,
fis_gRO40, fis_gRO41, fis_gRO42, fis_gRO43, fis_gRO44, fis_gRO45,
fis_gRO46, fis_gRO47, fis_gRO48, fis_gRO49 };
// Input range Min
FIS_TYPE fis_gIMin[] = { -6, -1, -1100, -100 };
// Input range Max
FIS_TYPE fis_gIMax[] = { 6, 1, 1100, 100 };
// Output range Min
FIS_TYPE fis_gOMin[] = { -1, -1 };
// Output range Max
FIS_TYPE fis_gOMax[] = { 1, 1 };
/*****
*****
// Data dependent support functions for Fuzzy Inference System
/*****
*****
FIS_TYPE fis_MF_out(FIS_TYPE** FuzzyRuleSet, FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;
    for (r = 0; r < fis_gcR; ++r)
    {
        int index = fis_gRO[r][o];
        if (index > 0)
        {
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
        }
        else if (index < 0)
        {
            index = -index - 1;
            mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
        }
        else
        {
            mfOut = 0;
        }
        FuzzyRuleSet[0][r] = fis_min(mfOut, FuzzyRuleSet[1][r]);
    }
    return fis_array_operation(FuzzyRuleSet[0], fis_gcR, fis_max);
}
FIS_TYPE fis_defuzz_centroid(FIS_TYPE** FuzzyRuleSet, int o)
{

```

```

    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) /
(FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;
    // calculate the area under the curve formed by the MF outputs
    for (i = 0; i < FIS_RESOLUTION; ++i){
        dist = fis_gOMin[o] + (step * i);
        slice = step * fis_MF_out(FuzzyRuleSet, dist, o);
        area += slice;
        momentum += slice*dist;
    }
    return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) :
(momentum / area));
}
//*****
// Fuzzy Inference System
//*****
void fis_evaluate()
{
    FIS_TYPE FuzzyInput0[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE FuzzyInput1[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE FuzzyInput2[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE FuzzyInput3[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE* FuzzyInput[fis_gcI] = { FuzzyInput0, FuzzyInput1,
FuzzyInput2, FuzzyInput3, };
    FIS_TYPE FuzzyOutput0[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE FuzzyOutput1[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE* FuzzyOutput[fis_gcO] = { FuzzyOutput0, FuzzyOutput1,
};
    FIS_TYPE FuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE FuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* FuzzyRuleSet[] = { FuzzyRules, FuzzyFires };
    FIS_TYPE sW = 0;

    // Transforming input to Fuzzy Input
    int i, j, r, o;
    for (i = 0; i < fis_gcI; ++i)
    {
        for (j = 0; j < fis_gIMFCount[i]; ++j)
        {
            FuzzyInput[i][j] =
                (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i],
fis_gMFICoeff[i][j]);
        }
    }
    int index = 0;
    for (r = 0; r < fis_gcR; ++r)
    {
        if (fis_gRType[r] == 1)
        {
            FuzzyFires[r] = FIS_MAX;
            for (i = 0; i < fis_gcI; ++i)
            {
                index = fis_gRI[r][i];
                if (index > 0)

```



```

        FuzzyFires[r] = fis_min(FuzzyFires[r],
FuzzyInput[i][index - 1]);
        else if (index < 0)
            FuzzyFires[r] = fis_min(FuzzyFires[r], 1 -
FuzzyInput[i][-index - 1]);
        else
            FuzzyFires[r] = fis_min(FuzzyFires[r], 1);
    }
}
else
{
    FuzzyFires[r] = FIS_MIN;
    for (i = 0; i < fis_gcI; ++i)
    {
        index = fis_gRI[r][i];
        if (index > 0)
            FuzzyFires[r] = fis_max(FuzzyFires[r],
FuzzyInput[i][index - 1]);
        else if (index < 0)
            FuzzyFires[r] = fis_max(FuzzyFires[r], 1 -
FuzzyInput[i][-index - 1]);
        else
            FuzzyFires[r] = fis_max(FuzzyFires[r], 0);
    }
}
FuzzyFires[r] = fis_gRWeight[r] * FuzzyFires[r];
sW += FuzzyFires[r];
}
if (sW == 0)
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
    }
}
else
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = fis_defuzz_centroid(FuzzyRuleSet, o);
    }
}
}
double avergearray(int* arr, int number) {
    int i;
    int max, min;
    double avg;
    long amount = 0;
    if (number <= 0) {
        Serial.println("Error number for the array to avraging!/n");
        return 0;
    }
    if (number < 5) { //less than 5, calculated directly statistics
        for (i = 0; i < number; i++) {
            amount += arr[i];
        }
        avg = amount / number;
        return avg;
    } else {

```

```

    if (arr[0] < arr[1]) {
        min = arr[0]; max = arr[1];
    }
    else {
        min = arr[1]; max = arr[0];
    }
    for (i = 2; i < number; i++) {
        if (arr[i] < min) {
            amount += min;          //arr<min
            min = arr[i];
        } else {
            if (arr[i] > max) {
                amount += max;      //arr>max
                max = arr[i];
            } else {
                amount += arr[i]; //min<=arr<=max
            }
        }
    }
    avg = (double)amount / (number - 2);
}
// Firebase Logger
void firebase_logger()
{
    firebaseJson.set("/Input_TDS", Input_TDS);
    firebaseJson.set("/deltaerrortds", deltaerrortds);
    firebaseJson.set("/errortds", errortds);
    firebaseJson.set("/Input_PH", Input_PH);
    firebaseJson.set("/deltaerrorph", deltaerrorph);
    firebaseJson.set("/errorph", errorph);
    firebaseJson.set("/aktuatorTds", g_fisOutput[1]);
    firebaseJson.set("/aktuatorph", g_fisOutput[0]);
    Firebase.updateNode(firebaseData, "/akuaponik", firebaseJson);
}

```

Lampiran B Hasil Pengujian

