



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE141599

**SISTEM PENGATURAN OKSIGEN TERLARUT
MENGUNAKAN METODE LOGIKA *FUZZY*
BERBASIS MIKROKONTROLER *TEENSY BOARD***

Luthfi Riadhi
NRP 2213100069

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Fajar Budiman, ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE141599

**CONTROL SYSTEM OF DISSOLVED OXYGEN USING
FUZZY LOGIC BASED ON TEENSY BOARD
MICROCONTROLLER**

Luthfi Riadhi
NRP 2213100069

Advisor
Dr. Muhammad Rivai, ST., MT.
Fajar Budiman, ST., M.Sc.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017


PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tugas akhir saya dengan judul “SISTEM PENGATURAN OKSIGEN TERLARUT MENGGUNAKAN METODE LOGIKA *FUZZY* BERBASIS MIKROKONTROLER *TEENSY BOARD*” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2017



Luthfi Riadhi
NRP. 2213100069

[Halaman ini sengaja dikosongkan]

**SISTEM PENGATURAN OKSIGEN TERLARUT
MENGUNAKAN METODE LOGIKA FUZZY
BERBASIS MIKROKONTROLER TEENSY BOARD**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

(Pada)

**Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro**

Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I



Dr. Muhammad Rivai, ST., MT.
NIP. 196904261994031003

Dosen Pembimbing II



Fajar Budiman, ST., M.Sc.
NIP. 198607072014041001



[Halaman ini sengaja dikosongkan]

SISTEM PENGATURAN OKSIGEN TERLARUT MENGUNAKAN METODE LOGIKA FUZZY BERBASIS MIKROKONTROLER TEENSY BOARD

Nama : Luthfi Riadhi
Pembimbing I : Dr. Muhammad Rivai, ST., MT.
Pembimbing II : Fajar Budiman, ST., M.Sc.

ABSTRAK

Oksigen terlarut merupakan faktor kritis dalam budidaya pembenihan ikan dan menentukan tingkat keberhasilan dan kegagalan dalam proses meningkatkan kualitas serta kuantitas pembenihan ikan. Kadar oksigen terlarut yang rendah menyebabkan proses penguraian, reproduksi, dan pertumbuhan di dalam kolam tidak berjalan dengan baik sehingga menyebabkan kematian pada ikan.

Pada tugas akhir ini penulis membuat sistem yang dapat mengatur kadar oksigen terlarut dalam air dengan memanfaatkan sensor *Dissolved Oxygen Meter AZ-8403*. *Output* data analog dari sensor dikonversi oleh *analog to digital converter (ADC)* menjadi data digital yang mempresentasikan kadar oksigen terlarut di air. Presentase kadar oksigen dikendalikan dengan *Fuzzy Logic* di dalam mikrokontroler *Teensy board*. Ketika level oksigen di bawah nilai *setting point* maka aerator diaktifkan untuk menghasilkan kadar oksigen terlarut di dalam air.

Hasil pengujian yang dilakukan pada akuarium dengan panjang 27cm, lebar 16cm dan tinggi 17cm yang berisi ikan di dalamnya. didapatkan nilai rata – rata kesalahan pembacaan sensor oksigen terlarut sebesar 0.075%. Pengujian respon sistem dilakukan terhadap variasi *setting point*. Dari data yang diperoleh menunjukkan bahwa respon sistem membutuhkan waktu 1 menit untuk mencapai nilai *setting point* dengan pembacaan *error* positif dan aerator dimatikan ketika pembacaan *error* negatif, karena sistem tidak dilengkapi dengan proses reduksi oksigen. Sistem yang dirancang telah mampu untuk mengontrol oksigen terlarut dengan menggunakan metode logika *fuzzy*.

Kata kunci: Kontrol Logika *Fuzzy*, Oksigen terlarut, *Teensy Board*

Halaman ini sengaja dikosongkan

CONTROL SYSTEM OF DISSOLVED OXYGEN USING FUZZY LOGIC BASED ON TEENSY BOARD MICROCONTROLLER

Name : Luthfi Riadhi
Advisor : Dr. Muhammad Rivai, ST., MT.
Co-Advisor : Fajar Budiman, ST., M.Sc.

ABSTRACT

Dissolved oxygen is a critical factor in the cultivation of fish hatcheries and determines the success rate and failure in the process of improving the quality and quantity of fish hatcheries. Low dissolved oxygen levels cause problems of process decomposition, reproduction, and growth in the pond does not work well resulting fish die.

In this final project author creates a system that can control dissolved oxygen levels in the water by utilizing sensor Dissolved Oxygen Meter AZ-8403. Analog output from sensor is converted by analog to digital converter (ADC) into digital data which presents dissolved oxygen levels in water. The percentage of oxygen levels controlled by fuzzy logic inside the teensy board microcontroller. When the oxygen levels are below the setting point, the aerator is activated to produce dissolved oxygen in the water.

The results test on the aquarium with 27cm length, 16cm wide, and 17cm high with fish in it obtained error value for the reading of dissolved oxygen sensor is 0.075%. The system was evaluated with different setting points. From the data shows that system response takes 1 minutes to reach setting point when the error value is positive and the aerator is stopped when the error value is negative, because the system does not have dissolved oxygen reduction process. The designed system has been able to control dissolved oxygen using fuzzy logic method.

Key words: Dissolved oxygen, Fuzzy Logic Controller, *Teensy Board*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan judul :

Sistem Pengaturan Oksigen Terlarut Menggunakan Metode Logika Fuzzy Berbasis Mikrokontroler

Tugas Akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu pembuatan tugas akhir ini, khususnya kepada:

1. Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing 1 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
2. Fajar Budiman, ST., M.Sc. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
3. Dr. Tri Arief Sardjono, ST., MT., Rachmad Setiawan, ST., MT., Astria Nur Irfansyah, ST., M.Eng., Ir. Siti Halimah Baki, MT. Selaku dosen penguji yang memberikan banyak masukan dan pengarahan sehingga penulis dapat menyelesaikan buku tugas akhir ini dengan baik
4. Seluruh dosen bidang studi Elektronika.
5. Dr. Eng. Ardyono Priyadi, ST., M.Eng. Selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
6. Karsidianto, Fitriani, Iudha Fakhreza, dan teman-teman laboratorium Elektronika serta seluruh keluarga yang memberikan dukungan baik moril maupun materil.

Terakhir, penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi banyak pihak. Penulis juga berharap Tugas Akhir ini dapat membantu pengembangan mikrokontroler *teensy board*.

Surabaya, 23 Juli 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	2
1.5. Metodologi	2
1.6. Sistematika Penulisan	3
1.7. Relevansi dan Manfaat	4
BAB II	5
2.1. Faktor Kualitas Kolam Air Tawar	5
2.1.1. <i>Temperature</i> Air	5
2.1.2. Oksigen Terlarut	6
2.1.3. Derajat Keasaman.....	6
2.1.4. Salinitas	7
2.2. Oksigen Terlarut (<i>Dissolved Oxygen</i>)	8
2.3. Kontrol Logika <i>Fuzzy</i>	12
2.4. <i>Teensy Board 3.2</i>	18
2.4.1. PWM (<i>Pulse Width Modulation</i>)	21
2.4.2. Protokol Komunikasi I2C	21
2.5. Penguat Operasional	22
2.5.1. <i>Differential Amplifier</i>	22
2.5.2. <i>Non Inverting Amplifier</i>	23
2.6. Sensor	24
2.7. ADC (<i>Analog to Digital Converter</i>)	25
2.8. SSR (<i>Solid State Relay</i>)	27
BAB III.....	31
3.1. Desain Sistem Keseluruhan	31
3.2. Sensor <i>Dissolved Oxygen</i> AZ-8403.....	34
3.3. Perancangan Rangkaian Penguat Operasional.....	38
3.4. Kontrol <i>Fuzzy Logic</i>	40
3.4.1. Himpunan Masukan <i>Fuzzy</i>	43

3.4.2	Himpunan Keluaran <i>Fuzzy</i>	44
3.4.3	<i>Fuzzy Inference System</i> (FIS).....	45
3.4.4	Fuzzy Rule	46
3.5.	Aktuator	46
3.5.1	Pompa Udara.....	46
3.5.2	SSR (<i>Solid State Relay</i>)	47
BAB IV	49
4.1.	Realisasi Desain Sistem	49
4.2.	Pengujian ADC	50
4.3.	Pengujian dan Kalibrasi Sensor	51
4.3.1	Pengujian dan Kalibrasi Oksigen Terlarut	51
4.3.2	Pengujian Awal Terhadap Saturasi Udara	53
4.3.3	Pengujian Awal Terhadap Air.....	54
4.3.4	Pengujian Terhadap Pompa Udara.....	55
4.3.5	Pengujian Ketika Pompa Udara Dimatikan	55
4.4.	Pengujian <i>Set Point</i>	56
BAB V	59
5.1.	Kesimpulan	59
5.2.	Saran	59
DAFTAR PUSTAKA	61
LAMPIRAN	63
BIODATA PENULIS	75

DAFTAR GAMBAR

Gambar 2. 1 Prinsip kerja pengukuran oksigen terlarut	10
Gambar 2. 2 Kurva kalibrasi pembacaan DO meter	11
Gambar 2. 3 Diagram blok fuzzy logic	13
Gambar 2. 4 Out single tone	14
Gambar 2. 5 Diagram blok kontrol logika fuzzy	15
Gambar 2. 6 Fungsi himpunan fuzzy	15
Gambar 2. 7 Sistem inferensi mamdani	17
Gambar 2. 8 Defuzzifikasi metode centroid	17
Gambar 2. 9 Tampak atas Teensy Board 3.2	19
Gambar 2. 10 Tampak bawah Teensy Board 3.2	19
Gambar 2. 11 Diagram Blok ARM Cortex-M4	20
Gambar 2. 12 Schematic Teensy Board 3.2	20
Gambar 2. 13 Modulasi PWM	21
Gambar 2. 14 Timing diagram I2C	22
Gambar 2. 15 Rangkaian dasar operational amplifier	22
Gambar 2. 16 Differential Amplifier	23
Gambar 2. 17 Non Inverting Amplifier	24
Gambar 2. 18 Rangkaian Opto triac	27
Gambar 2. 19 Sinyal output zero crossing detector circuit dan ssr beban resistif	28
Gambar 2. 20 Rangkaian Snubber komponen switching	29
Gambar 3. 1 Diagram blok sistem	31
Gambar 3. 2 Diagram blok perangkat keras sistem	32
Gambar 3. 3 Gambar alur kerja sistem	33
Gambar 3. 4 Desain fisik Sistem dilihat dari samping	34
Gambar 3. 5 Desain Fisik Sistem dilihat dari depan	34
Gambar 3. 6 Dissolved Oxygen AZ-8403	35
Gambar 3. 7 Dissolved oxygen probe	36
Gambar 3. 8 Rangkaian penguat operasional	39
Gambar 3. 9 Rangkaian pengaman	39
Gambar 3. 10 Diagram blok kontrol sistem	40
Gambar 3. 11 Diagram blok kontrol logika fuzzy	40
Gambar 3. 12 Tipe keanggotaan segitiga	41
Gambar 3. 13 Program fungsi himpunan segitiga	42
Gambar 3. 14 Tipe keanggotaan trapesium	42
Gambar 3. 15 Program fungsi himpunan trapesium	43

Gambar 3. 16 Grafik fungsi keanggotaan nilai error	43
Gambar 3. 17 Program fungsi input error	43
Gambar 3. 18 Grafik Fungsi keanggotaan nilai Δ Error	44
Gambar 3. 19 Program fungsi input Δ error.....	44
Gambar 3. 20 Out put Fuzzy Logic	45
Gambar 3. 21 Program fungsi output duty cycle (%).	45
Gambar 3. 22 pompa udara.....	46
Gambar 3. 23 Mekanisme pompa menghisap udara	47
Gambar 3. 24 Mekanisme pompa mengeluarkan udara.....	47
Gambar 3. 25 OMRON G3MB-202P	47
Gambar 4. 1 Tampak depan sistem.....	49
Gambar 4. 2 Pembacaan error ADC	51
Gambar 4. 3 Perbandingan output sensor dan amplifier	52
Gambar 4. 4 Grafik pengujian dan kalibrasi sensor oksigen terlarut	53
Gambar 4. 5 Output data di udara hingga masuk ke dalam air	54
Gambar 4. 6 Output data di beri pompa udara	55
Gambar 4. 7 Output data ketika didiamkan tanpa pompa udara	56
Gambar 4. 8 Oksigen Terlarut pada Set Point 3.5	57
Gambar 4. 9 Oksigen Terlarut pada Set Point 4.00	57
Gambar 4. 10 Oksigen Terlarut Pada Set Point 4.50	58
Gambar 4. 11 Oksigen Terlarut Pada Set Point 5.00	58

DAFTAR TABEL

Tabel 2. 1 Temperature ikan air tawar.....	5
Tabel 2. 2 Oksigen terlarut ikan air tawar	6
Tabel 2. 3 Keasaman ikan air tawar	7
Tabel 2. 4 Salinitas ikan air tawar	7
Tabel 2. 5 Rule tabel kontrol logika fuzzy	16
Tabel 3. 1 Spesifikasi DO meter 8403.....	38
Tabel 3. 2 Rules.....	46
Tabel 3. 3 Spesifikasi SSR OMRON G3MB-202P	48
Tabel 4. 1 Pembacaan ADC	50
Tabel 4. 2 Nilai pengujian oksigen terlarut	51
Tabel 4. 3 Pengujian terhadap saturasi udara	53

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam pembudidayaan ikan, keberhasilan di bidang pembenihan sangat ditentukan oleh beberapa faktor antara lain: kualitas benih, kualitas air, pengolahan, dan sebagainya. Faktor kualitas air dalam hal ini meliputi: suhu air, salinitas, pH, dan oksigen terlarut (DO).

Salah satu masalah dalam *aquaculture* yaitu mengetahui dan mengatasi masalah kadar oksigen dalam ekosistem air. Pada tahun 2012 terdapat kasus kematian ikan pada budidaya ikan di Wukisari kecamatan Cangkringan, Sleman, Yogyakarta. Target pembibitan pembenihan ikan lele sekitar 943200 ekor dan hasil yang didapat hanya sekitar 419200 ekor [1]. Dari data tersebut dapat disimpulkan bahwa tingkat kelangsungan hidup bibit ikan lele sebesar 34,22% atau tingkat kematian larva sebesar 65,78% dalam satu siklus pembenihan [2]. Kualitas air menjadi faktor yang sangat menentukan tingkat keberhasilan budidaya ikan. Salah satu faktor kualitas air yang menyebabkan tingginya kematian ikan adalah kadar oksigen terlarut [3]. Kadar oksigen terlarut yang rendah menyebabkan proses penguraian, reproduksi, dan pertumbuhan di dalam kolam tidak berjalan dengan baik. Nilai minimum kadar oksigen terlarut untuk budidaya ikan adalah 3 ppm [4]. Kebutuhan oksigen terlarut pada ikan dipengaruhi oleh umur, aktivitas dan kondisi perairan [5].

Seperti yang telah dipaparkan diatas kandungan oksigen berpengaruh pada proses oksidasi dan reduksi bahan organik dan anorganik[6]. Kekurangan oksigen akan menyebabkan ikan kurang nafsu makan dan berkembangnya bakteri yang menyebabkan kematian pada ikan. Kekurangan oksigen terlarut juga terjadi pada tipe kolam air tenang yang menjadi fokus utama alat ini akan dipasang.

Masalah diatas melatarbelakangi penulis untuk melakukan penelitian ini menggunakan metode *fuzzy* berbasis mikrokontroler. Sistem ini diharapkan dapat digunakan sebagai alat kontrol oksigen terlarut yang efektif, murah, dan mudah penggunaannya.

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah, antara lain :

1. Bagaimana mendeteksi oksigen terlarut pada air.
2. Metode apa yang digunakan untuk menkontrol oksigen terlarut.
3. Bagaimana cara mengimplementasikan metode ini agar lebih ringkas.

1.3. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah

1. Penelitian ini dibatasi pada akuarium dengan panjang 27cm, lebar 16cm, dan tinggi 17cm dengan menggunakan ikan di dalamnya.
2. Objek pengujian adalah air PAM.
3. Kontrol *fuzzy logic* dilakukan oleh mikrokontroler.
4. Sistem ini hanya diaplikasikan untuk pembenihan ikan di dalam wadah akuarium.

1.4. Tujuan

Tujuan yang ingin dicapai dalam perancangan ini adalah:

1. Penggunaan sensor *Dissolved Oxygen Meter AZ-8403*.
2. Penerapan metode *Fuzzy Logic* sebagai kontrol oksigen terlarut.
3. Penggunaan *Teensy USB Development Board*.

1.5. Metodologi

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, dan artikel-artikel di internet dan forum-forum diskusi internet.

2. Perancangan *Software*

Pada tahap ini dilakukan perancangan *source code* untuk sistem tertanam sebagai unit pemroses data dan pengekseskusi metode logika *fuzzy*.

3. Perancangan *Hardware*

Pada tahap ini dilakukan perancangan perangkat keras yang digunakan dalam penelitian tugas akhir ini.

4. Pengujian Sistem

Pengujian alat dilakukan untuk menentukan kelebihan dan kekurangan dari sistem yang telah dirancang. Pengujian dilakukan untuk melihat apakah *software* dan *hardware* dapat bekerja secara baik.

Untuk pengujian dapat dilakukan dalam tiga tahap. Pertama adalah pengambilan dan pengolahan data dari sensor DO meter menggunakan *fuzzy logic*. Kedua adalah *output* data dari *fuzzy logic* akan digunakan untuk mengatur pompa udara. Dan terakhir adalah data akan diteruskan ke komputer untuk ditampilkan grafiknya.

5. Analisa

Analisa dilakukan terhadap hasil dari pengujian sehingga dapat ditentukan karakteristik dari *software* dan *hardware* yang telah dibuat. Penarikan kesimpulan diketahui dengan kemampuan sistem dalam mengatur kadar oksigen di dalam air dengan metode *fuzzy logic*.

6. Penyusunan Laporan Tugas Akhir

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka, teori penunjang, perancangan sistem, pengujian, dan penutup.

1.6. Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

➤ **BAB I : PENDAHULUAN**

Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

➤ **BAB II : TINJAUAN PUSTAKA DAN TEORI PENUNJANG**

Bab ini menjelaskan tentang teori penunjang dan *literature* yang dibutuhkan dalam pengerjakan tugas akhir ini. Dasar teori yang menunjang meliputi faktor kualitas air yang baik untuk ikan, metode kontrol *fuzzy logic* dan *flowchart* cara kerja sistem. Bagian ini memaparkan mengenai beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Tugas Akhir ini.

➤ **BAB III : PERANCANGAN SISTEM**

Bab ini menjelaskan secara rinci tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk kontrol oksigen terlarut di dalam air.

➤ **BAB IV : PENGUJIAN**

Pada bab ini akan menjelaskan hasil uji coba sistem beserta analisisnya.

➤ **BAB V : PENUTUP**

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangan lebih lanjut.

1.7. Relevansi dan Manfaat

Hasil yang diharapkan dari tugas akhir ini diharapkan mampu meringankan pekerjaan manusia pada bidang pembudidayaan ikan khususnya pada pembenihan ikan yang memerlukan kondisi air yang baik, mulai dari awal ikan bertelur hingga penetasan telur. Memberikan analisa tentang penggunaan aerator, menggunakan mikrokontroler *teensy board*, memberikan analisa tentang metode kontrol *fuzzy logic* yang diterapkan untuk aerator penambah oksigen pada akuarium.

BAB II

TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Teori penunjang dalam bab ini menjelaskan tentang hal yang berhubungan dengan keseluruhan sistem yang akan dibuat pada tugas akhir ini. Sedangkan tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan pernah diimplementasikan oleh penulis-penulis sebelumnya.

2.1. Faktor Kualitas Kolam Air Tawar

Ada 4 faktor penting yang mempengaruhi kualitas air pada pembudidayaan ikan air tawar:

2.1.1. *Temperature Air*

Suhu air mempengaruhi densitasnya. Semakin tinggi suhu air, densitasnya semakin rendah (gr/cm^3). Perbedaan densitas air di lapisan atas dan di lapisan bawah dapat menyebabkan stratifikasi. Suhu yang mematikan untuk hampir semua semua jenis ikan adalah 10 - 11 derajat celsius selama beberapa hari. Nafsu makan ikan menurun pada suhu di bawah 16 derajat celsius, sementara reproduksi ikan mengalami penurunan pada suhu di bawah 21 derajat celsius. Berikut adalah tabel suhu perairan yang optimum untuk pertumbuhan ikan air tawar.

Tabel 2. 1 Temperature ikan air tawar

No	Jenis Ikan	Temperature Optimum ($^{\circ}\text{C}$)
1.	Nilem	18 – 28
2.	Mas/Tombro	20 – 25
3.	Patin	28 – 32
4.	Bawal	25 – 30
5.	Gurame	24 – 28
6.	Nila	25 – 30
7.	Sidat	28 – 29
8.	Lele	25 – 30
9.	Gabus	25 – 30

2.1.2. Oksigen Terlarut

Oksigen diperlukann oleh makhluk hidup seperti ikan dan organisme perairan lainya untuk pernafasan dan metabolisme tubuh. Oksigen diperlukan untuk pembakaran pakan sehingga menghasilkan energi untuk melakukan aktifitas gerak, pertumbuhan, dan reproduksi. Laju pertumbuhan ikan dan konversi pakan sangat dipengaruhi oleh kandungan oksigen dalam air. Berikut ini tabel DO yang sesuai untuk budidaya ikan air tawar.

Tabel 2. 2 Oksigen terlarut ikan air tawar

No	Jenis Ikan	Oksigen Terlarut (ppm)
1.	Patin	4.5 – 6.5
2.	Nila	4 - 6
3.	Sidat	5 - 6
4.	Gabus	4.2 - 5.6
5.	Nilem	5 - 7
6.	Mas	4 - 5
7.	bawal	4 - 6
8.	lele	3 -5
9.	gurame	4 - 6

2.1.3 Derajat Keasaman

Derajat keasaman air dibagi menjadi tiga yaitu, pH rendah (asam), pH netral dan pH tinggi (basa). Derajat keasaman air dipengaruhi oleh ion Hidrogen (H⁺). Air menjadi asam apabila pH <7 dan dikatakan basa bila PH >7. Derajat keasaman air budidaya yang memenuhi syarat adalah 5 - 8,5. Untuk budidaya ikan air tawar pH yang cocok adalah 6,5 - 7,5. Syarat lain yang penting adalah fluktuasi atau perbedaan pH pagi dan siang tidak lebih dari 1. Misalnya, pagi hari pH air pada kolam atau karamba jaring apung adalah 6,5 maka, pH pada siang hari tidak boleh mencapai angka 8.

Tabel 2. 3 Keasaman ikan air tawar

No	Jenis Ikan	Keasaman (pH)
1.	Nilem	6.5 – 7.5
2.	Mas/Tombro	7 – 8
3.	Patin	6 – 7
4.	Bawal	7 – 8
5.	Gurame	6.5 – 8
6.	Nila	6.5 – 8.5
7.	Sidat	7 – 8
8.	Lele	6.5 – 8
9.	Gabus	6 – 7.5

2.1.4 Salinitas

Salinitas atau kadar garam adalah jumlah kandungan bahan padat dalam satu kilogram air laut. Seluruh karbonat telah diubah menjadi oksida, brom, dan yodium telah disetarakan dengan klor dan bahan organik yang telah dioksidasi. Secara langsung salinitas media akan mempengaruhi tekanan osmotik cairan tubuh ikan. Apabila osmotik lingkungan (salinitas) berbeda jauh dengan tekanan osmotik cairan tubuh (kondisi tidak ideal) maka, osmotikmedia akan menjadi beban bagi ikan sehingga dibutuhkan energi yang relatif besar untuk mempertahankan osmotik tubuhnya agar tetap berada pada keadaan yang ideal. Pembelanjaan energi untuk osmoregulasi akan mempengaruhi tingkat konsumsi pakan dan konversi menjadi berat tubuh.

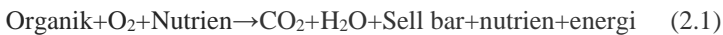
Tabel 2. 4 Salinitas ikan air tawar

No	perairan	Salinitas (ppt)
1.	Air tawar	
	<i>Fresh water</i>	<0.5
	oligohaline	0.5 – 3.0
2.	Air laut	
	<i>Marine</i>	30 - 40

2.2. Oksigen Terlarut (*Dissolved Oxygen*)

Kadar oksigen diukur menurut satuan ppm (*part per million*). Kadar oksigen dibawah 2 ppm atau lebih rendah dapat menyebabkan makhluk hidup di laut menjadi stress dan mati. Namun beberapa makhluk hidup seperti tiram dapat bertahan hidup pada kadar oksigen yang rendah untuk beberapa hari (1 ppm untuk 5 hari). Oksigen merupakan unsur yang dapat dengan cepat berubah di dalam air walaupun kadarnya lebih kecil bila dibandingkan di udara. Pada air laut, oksigen masuk ke dalam air melalui proses fotosintesis dan penyerapan dari udara seperti gerakan gelombang air maupun perputaran arus air.

Di dalam air, oksigen memainkan peranan dalam menguraikan komponen-komponen kimia menjadi komponen yang lebih sederhana. Oksigen memiliki kemampuan untuk beroksidasi dengan zat pencemar seperti komponen organik sehingga zat pencemar tersebut tidak membahayakan. Oksigen juga diperlukan oleh mikroorganisme, baik yang bersifat aerob maupun anaerob dalam proses metabolisme. Dengan adanya oksigen dalam air, mikroorganisme semakin giat dalam menguraikan kandungan dalam air. Reaksi yang terjadi dalam penguraian tersebut adalah komponen.



Jika reaksi penguraian komponen kimia dalam air terus berlaku, maka kadar oksigen akan menurun. Pada klimaksnya, oksigen yang tersedia tidak cukup untuk menguraikan komponen kimia tersebut. Keadaan yang demikian merupakan pencemaran berat pada air.

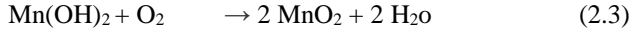
A Metode Pengukuran Oksigen Terlarut

Terdapat 2 metode untuk menganalisa oksigen terlarut:

a. Metode titrasi

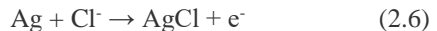
Prinsipnya dengan menggunakan titrasi iodometri. Sampel yang akan dianalisis terlebih dahulu ditambahkan larutan MnCl_2 , NaOH , atau KI , sehingga akan terjadi endapan MnO_2 . Dengan menambahkan H_2SO_4 atau HCl maka endapan yang terjadi akan larut kembali dan juga akan membebaskan molekul iodium (I_2) yang ekuivalen dengan oksigen terlarut. Iodium yang dibebaskan ini selanjutnya

dititrasi dengan larutan standar natrium tiosulfat ($\text{Na}_2\text{S}_2\text{O}_3$) dan menggunakan indikator larutan amilum (kanji). Reaksi kimia yang terjadi dapat dirumuskan:

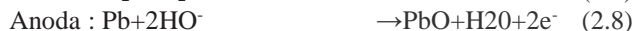


b. Metode elektrokimia

DO meter tersusun atas beberapa komponen utama yang disketsakan pada gambar di bawah ini. Terdapat dua elektrode utama yang masing-masing berfungsi sebagai katode dan anode. Batang katode terbuat dari logam mulia seperti emas atau platina, sedangkan batang anode terbuat dari bahan perak. Kedua elektrode ini terselimuti cairan elektrolit KCl yang memiliki pH netral. Permukaan elektrode perak akan membentuk senyawa AgCl yang sifatnya stabil dan membuat elektrode ini memiliki beda potensial yang tetap. Oleh karena itu, anode pada DO meter ini berfungsi sebagai elektrode referensi.

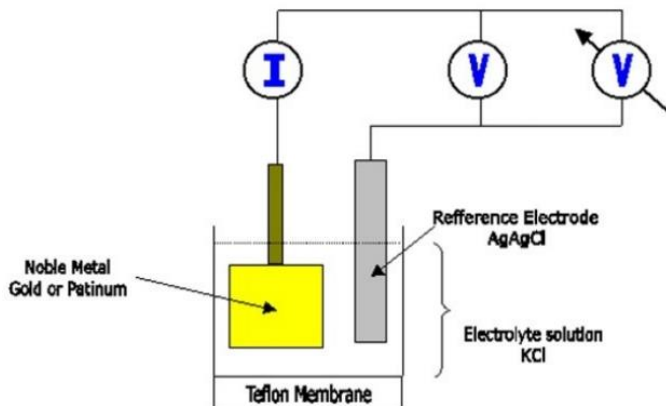


Cara penentuan oksigen terlarut dengan metoda elektrokimia adalah cara langsung untuk menentukan oksigen terlarut. Prinsip kerjanya adalah menggunakan *probe* oksigen yang terdiri dari katoda dan anoda yang direndam dalam larutan elektrolit. Pada alat DO meter ini, *probe* biasanya menggunakan katoda perak (Ag) dan anoda timbal (Pb). Secara keseluruhan, elektrode ini dilapisi dengan membran plastik yang bersifat semi *permeable* terhadap oksigen. Reaksi kimia yang akan terjadi adalah:



Aliran reaksi yang terjadi tersebut tergantung dari aliran oksigen pada katoda. Difusi oksigen dari sampel ke elektroda berbanding lurus terhadap konsentrasi oksigen terlarut.

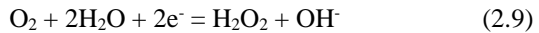
Penentuan oksigen terlarut (DO) dengan cara titrasi berdasarkan metoda *WINKLER* lebih analitis apabila dibandingkan dengan cara alat DO meter. Hal yang perlu diperhatikan dalam titrasi iodometri ialah penentuan titik akhir titrasinya, standarisasi larutan tiosulfat, dan pembuatan larutan standar kaliumbikromat yang tepat. Dengan mengikuti prosedur penimbangan kaliumbikromat dan standarisasi tiosulfat secara analitis, akan diperoleh hasil penentuan oksigen terlarut yang lebih akurat. Sedangkan penentuan oksigen terlarut dengan H⁺24 cara DO meter, harus diperhatikan suhu dan salinitas sampel yang akan diperiksa. Peranan suhu dan salinitas ini sangat vital terhadap akurasi penentuan oksigen terlarut dengan cara DO meter. Disamping itu, peranan kalibrasi alat sangat menentukan akurasinya hasil penentuan. Berdasarkan pengalaman di lapangan, penentuan oksigen terlarut dengan cara titrasi lebih dianjurkan untuk mendapatkan hasil yang lebih akurat. Alat DO meter masih dianjurkan jika sifat penentuannya hanya bersifat kisaran.



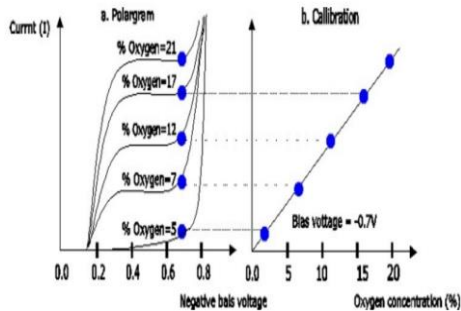
Gambar 2. 1 Prinsip kerja pengukuran oksigen terlarut [7]

Kedua elektrode DO meter yang diselimuti larutan KCl tersebut dibungkus oleh sebuah wadah kedap yang pada bagian ujung adalah berupa komponen penting lainnya yaitu membran teflon. Membran ini hanya bisa dilewati oleh gas terlarut yang ada di dalam cairan terukur, ia tidak akan bisa dilewati oleh material lain termasuk ion, senyawa lain, dan tentu saja padatan pengotor.

Prinsip kerja DO meter adalah berdasarkan fenomena polarografi yang terjadi di antara dua elektrode katode dan anode. Tegangan listrik negatif diberikan kepada elektrode katode. Adanya tegangan negatif ini akan mengakibatkan reaksi kimia terjadi secara cepat antara air dan oksigen terlarut pada permukaan katode. Berikut adalah reaksi kimia yang terjadi pada elektrode katode:



Tegangan listrik akan terus naik mencapai nilai jenuh yang setara dengan sudah bereaksinya seluruh oksigen terlarut pada permukaan elektrode katode. Tegangan listrik jenuh ini ditandai dengan hampir naiknya pembacaan arus listrik, setelah beberapa saat diam di satu nilai meskipun nilai tegangan dinaikkan. Setelah melewati tegangan jenuh ini, arus listrik terus naik jika tegangan terus ditambah. Naiknya nilai arus ini terjadi karena reaksi kimia lain telah terjadi, terutama adalah reaksi pecahnya molekul air H₂O menjadi ion H⁺ dan OH⁻.



Gambar 2. 2 Kurva kalibrasi pembacaan DO meter [7]

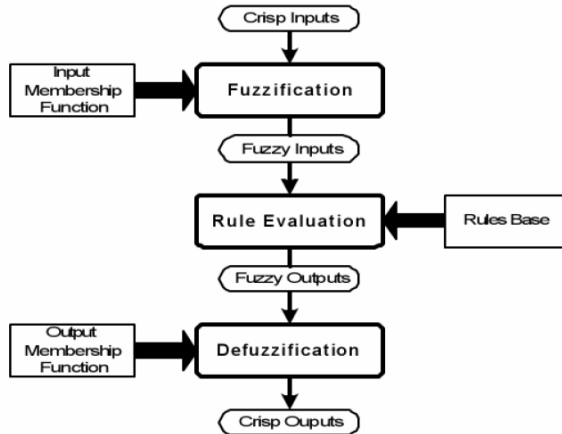
Pembacaan nilai oksigen terlarut didapatkan dari nilai arus listrik pada saat semua oksigen terdifusi pada permukaan elektrode katode. Dengan kata lain, arus listrik yang terbaca pada saat sistem mencapai tegangan jenuh, setara dengan besaran oksigen terlarut. Dengan menggunakan metode kalibrasi linier seperti kurva di atas, didapatkan nilai oksigen terlarut yang dicari.

2.3. Kontrol Logika Fuzzy

Fuzzy Logic adalah suatu cabang ilmu *Artificial Intelligence*, yaitu suatu pengetahuan yang membuat komputer dapat meniru kecerdasan manusia sehingga diharapkan komputer dapat melakukan hal-hal yang apabila dikerjakan manusia memerlukan kecerdasan. *Fuzzy Logic* dapat memecahkan masalah dalam pengendali yang tersimpan dan pemrosesan informasi. bisa diimplementasikan pada sistem tertanam pada mikrokontroler [8]. *fuzzy logic* menyerupai pembuatan keputusan pada manusia dengan kemampuannya untuk bekerja dari data yang ditafsirkan dan mencari solusi yang tepat. *Fuzzy logic* pada dasarnya merupakan logika bernilai banyak (*multivalued logic*) yang dapat mendefinisikan nilai diantara keadaan konvensional seperti ya atau tidak. Penalaran *fuzzy* menyediakan cara untuk memahami kinerja dari sistem dengan cara menilai *input* dan *output* *system* dari hasil pengamatan.

Logika *fuzzy* (logika samar) itu sendiri merupakan logika yang berhadapan dengan konsep kebenaran sebagian, dimana logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah *binary* (0 atau 1). Logika *fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1. Logika *fuzzy* dianggap mampu untuk memetakan suatu *input* kedalam suatu *output* tanpa mengabaikan faktor-faktor yang ada. Logika *fuzzy* diyakini dapat sangat fleksibel dan memiliki toleransi terhadap data-data yang ada. Dengan berdasarkan logika *fuzzy*, akan dihasilkan suatu model dari suatu sistem yang mampu memperkirakan jumlah produksi. Faktor-faktor yang mempengaruhi dalam menentukan jumlah produksi logika *fuzzy* antara lain jumlah permintaan dan jumlah persediaan.

Berikut merupakan diagram blok dari *fuzzy logic control*:



Gambar 2. 3 Diagram blok *fuzzy logic* [9]

1. *Fuzzification*

proses untuk mengubah variabel *non fuzzy* (variabel numerik) menjadi variabel *fuzzy* (variabel linguistik). Prosedur fuzzifikasi merupakan proses untuk mengubah variabel *non fuzzy* (variabel numerik) menjadi variabel *fuzzy* (*variable linguistik*). Nilai *error* dan $\Delta error$ yang dikuantisasi sebelumnya diolah oleh kontroler *fuzzy logic*, kemudian diubah terlebih dahulu ke dalam *variable fuzzy*. Melalui *membership function* (fungsi keanggotaan) yang telah disusun, maka dari nilai *error* dan $\Delta error$ kuantisasi akan didapatkan derajat keanggotaan bagi masing-masing nilai *error* dan $\Delta error$. Pada unit fuzzifikasi ini terjadi proses transformasi, yang dilakukan dengan cara pemetaan ruang masukan, dari variabel masukan domain *non-fuzzy* (*crisp*) ke dalam domain *fuzzy*, dengan bantuan faktor penskala (*scaling factor*). Faktor penskala menggunakan metode heuristik, diatur sedemikian rupa sehingga seluruh variabel masukan terpetakan dalam semesta pembicaraan yang dirancang. Perbedaan kecepatan dalam diskala dengan *gain* (G1), sedangkan perbedaan perubahan kecepatan ($\Delta error$) diskala dengan *gain* (G2).

2. Rules

Menentukan aturan kondisi *output* sesuai dengan kondisi *input*. Basis pengetahuan *fuzzy* terdiri dari beberapa aturan *fuzzy* yang dikelompokkan kedalam suatu basis aturan, disebut basis aturan *fuzzy* (*fuzzy rule base*). *Rule base* merupakan dasar dari pengambilan keputusan atau *inference process*, untuk mendapatkan aksi keluaran sinyal kontrol dari suatu kondisi masukan yaitu *error* dan $\Delta error$, dengan berdasarkan *rules* yang telah ditetapkan. Proses inferensi menghasilkan sinyal keluaran yang masih dalam bentuk bilangan *fuzzy*, yaitu derajat keanggotaan dari sinyal kontrol. Pendefinisian *rule base* tergantung dari data yang telah disesuaikan dan dalam bentuk tabel. Sebagai contoh *Rule base* dengan 5 tingkat kuantisasi untuk variabel *input* (*error* (E) dan $\Delta error$ (dE))

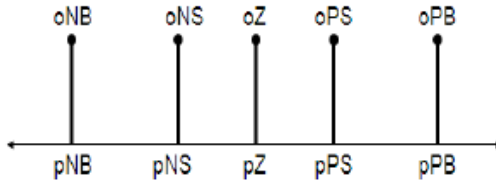
3. Defuzzification

Proses pengubahan data-data *fuzzy* tersebut menjadi data-data numerik yang dapat dikirimkan ke peralatan pengendalian [10]. Defuzzifikasi merupakan kebalikan dari proses transformasi sebuah himpunan *fuzzy* kedalam himpunan tegas. Metode defuzzifikasi yang umum digunakan ada 2 macam yaitu: [11]

1. Metode Mamdani *center of gravity*.

$$y = \frac{\sum_{j=1}^R c^j \int \mu_{0j}(u) du}{\sum_{j=1}^R \int \mu_{0j}(u) du} \quad (2.11)$$

2. Metode Sugeno, *single tone*

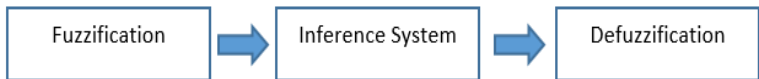


Gambar 2. 4 Out single tone

$$Out = \frac{oNB.pNB + oNS.pNS + oZ.pZ + oPS.pPS + oPB.pPB}{oNB + oNS + oZ + oPS + oPB} \quad (2.12)$$

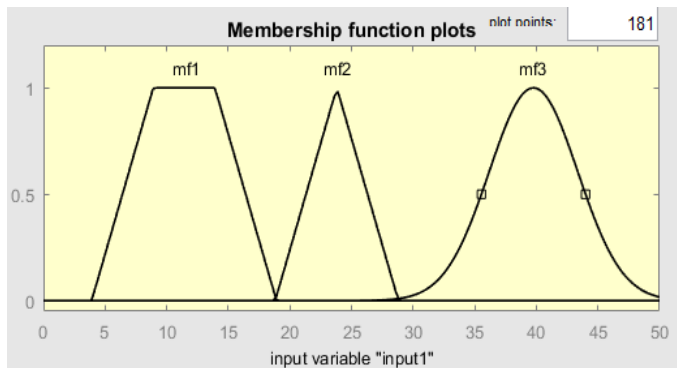
$$Out = \frac{\sum_{i=1}^N O_i \cdot p_i}{\sum_{i=1}^N 10_i} \quad (2.13)$$

Kontrol logika *fuzzy* menggunakan *Fuzzy Inference System* (FIS) dengan menggunakan logika *fuzzy* yang memiliki nilai derajat keanggotaan himpunan kontinu dari 0 hingga 1 [12]. FIS menerima *input* yang berupa himpunan *crisp* yang lalu dirubah menjadi himpunan *fuzzy* yang disebut dengan fuzzifikasi. Hasil fuzzifikasi menjadi masukan himpunan *fuzzy* akan diproses oleh algoritma inferensi *fuzzy* yang berupa tabel peraturan yang akan menghasilkan nilai keluaran himpunan *fuzzy*. Keluaran himpunan *fuzzy* harus dikembalikan ke dalam bentuk himpunan *crisp* agar dapat diaplikasikan di dunia nyata dengan proses yang dinamakan dengan defuzzifikasi



Gambar 2. 5 Diagram blok kontrol logika *fuzzy*

Pada gambar 2.6 adalah contoh bentuk himpunan yang digunakan dalam fungsi matlab. Untuk setiap bentuk trapesium, segitiga, dan sebagainya memiliki rumus yang berbeda ketika dikonversikan ke dalam bahasa pemrograman.



Gambar 2. 6 Fungsi himpunan *fuzzy*

Pada Tabel 2.5 Merupakan contoh tabel rule yang terdiri dari 12 rule. Rule ini digunakan untuk mengontrol output yang diinginkan.

Rule bisa dibuat sesuai kebutuhan *user* untuk mencapai nilai *setting point*.

Tabel 2. 5 Rule tabel kontrol logika *fuzzy*

		Error				
		NB	NK	Z	PK	PB
Δ Error	NEGB	OFF	OFF	OFF	MED	HIGH
	NEGS	OFF	OFF	OFF	MED	HIGH
	NOL	OFF	OFF	OFF	HIGH	HIGH
	POSS	OFF	OFF	OFF	MED	HIGH
	POSB	OFF	OFF	OFF	HIGH	HIGH

Fuzzifikasi adalah proses perubahan himpunan *crisp* ke himpunan *fuzzy*. Pada umumnya fungsi himpunan *fuzzy* yang digunakan adalah segitiga, *gaussian*, dan trapesium. Fungsi himpunan *fuzzy* merepresentasikan nilai derajat keanggotaan *fuzzy* sebagai fungsi dari nilai himpunan *crisp*.

Sistem inferensi dalam *fuzzy* berfungsi sebagai pengolah masukan *input* himpunan *fuzzy* dengan tabel peraturan untuk menghasilkan keluaran yang berupa himpunan *fuzzy*. Tabel peraturan adalah koleksi pernyataan linguistik yang menjelaskan bagaimana FIS akan menentukan keluaran himpunan *fuzzy*. Secara linguistik peraturan *fuzzy* ditulis dengan *if-then statement* seperti berikut *if* (masukan himpunan *fuzzy* 1) *and/or* (masukan himpunan *fuzzy* 2), *then* (keluaran himpunan *fuzzy*). Atau secara linguistik dapat dicontohkan, jika suhu tinggi dan kelembapan rendah, maka ruangan berudara panas.

Terdapat dua jenis sistem inferensi dalam FIS, yaitu metode Mamdani dan Sugeno. Perbedaan keduanya terletak pada bagaimana hasil keluaran himpunan *fuzzy* dihasilkan. Metode Mamdani menghasilkan keluaran himpunan *fuzzy* dengan operasi *MIN-MAX* pada pernyataan *if*- pada masukan himpunan *fuzzy* untuk memotong derajat keanggotaan himpunan *fuzzy* keluaran yang lalu akan dilakukan operasi *and* pada semua himpunan *fuzzy* keluaran yang memenuhi tabel peraturan *fuzzy*.

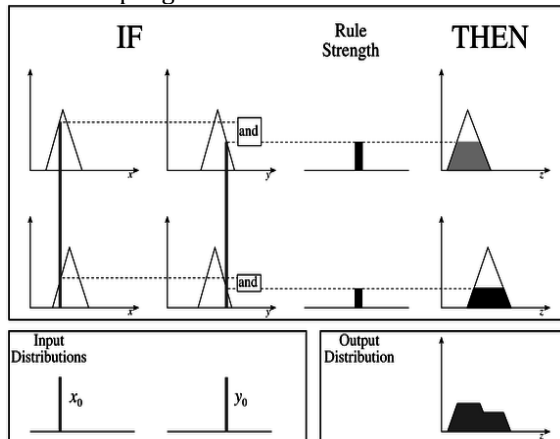
Metode Sugeno tidak memiliki himpunan *fuzzy* keluaran, sehingga nilai keluaran himpunan *crisp* tidak akan didapatkan melalui proses defuzzifikasi. Nilai keluaran dihitung melalui perkalian dari masing-masing *input* dengan konstanta dan dijumlahkan. Berikut adalah perhitungan untuk keluaran dari metode Sugeno.

$$x^* = \frac{Y_1 X_1 + Y_2 X_2}{Y_1 + Y_2} \quad (2.14)$$

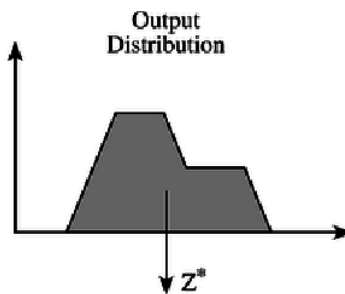
x^* = nilai hasil keluaran *crisp*

x = nilai keluaran *if-then statement*

γ = konstanta pengali keluaran



Gambar 2. 7 Sistem inferensi mamdani [12]



Gambar 2. 8 Defuzzifikasi metode *centroid* [12]

Defuzzifikasi merupakan proses untuk merubah sebuah himpunan *fuzzy* menjadi nilai *crisp*. Proses ini digunakan pada FIS metode Mamdani untuk mendapatkan hasil keluaran *crisp* dari keluaran himpunan *fuzzy*. Terdapat banyak metode defuzzifikasi yang dapat digunakan dan metode *centroid* merupakan metode yang paling sering digunakan. Metode *centroid* menghitung titik massa dari himpunan *fuzzy* keluaran untuk didapatkan titik yang menjadi nilai keluaran *crisp*.

$$Z^* = \frac{\sum_n^{\infty} Z_n \mu_c(Z_n)}{\sum_n^{\infty} \mu_c(Z_n)} \quad (2.15)$$

Z^* merupakan nilai keluaran *crisp* dan Z_n adalah derajat keanggotaan pada himpunan *fuzzy* keluaran pada μ_c .

2.4. *Teensy Board 3.2*

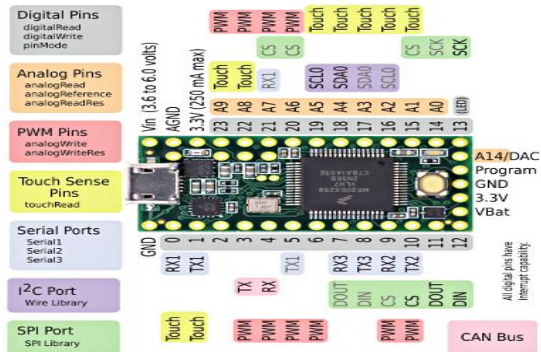
Teensy 3.2 adalah *development board* berbentuk kecil dan dapat dipasang pada *breadboard* yang di desain oleh Paul Stoffregen dan PJRC. *Teensy 3.2* membawa 32-bit ARM Cortex-M4 yang murah kepada *hobbyist*, siswa, dan *engineer* menggunakan versi adaptasi dari arduino IDE (*Teensyduino*) atau diprogram secara langsung di bahasa C.

Pemilihan *Teensy* berdasar pada kebutuhan pemrosesan sinyal yang akan memakan *resources* yang besar namun tetap memperhatikan keringkasan. Hal ini bertujuan agar mudah diimplementasikan pada alat sensor dan aktuator

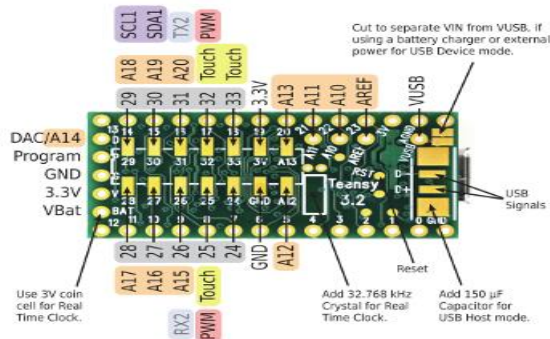
Berikut adalah spesifikasi teknis dari *Teensy 3.2* :

- 32 bit ARM Cortex-M4 72MHz CPU (M4 = DSP extensions)
- 256K Flash Memory, 64K RAM, 2K EEPROM
- 21 buah High Resolution Analog Inputs (13 bits usable, 16 bit hardware)
- 34 buah Digital I/O Pins (21 shared with analog)
- 12 buah PWM outputs
- 1 buah 12-bit DAC output
- 8 Timers for intervals/delays, separate from PWM
- USB with dedicated DMA memory transfers
- CAN bus

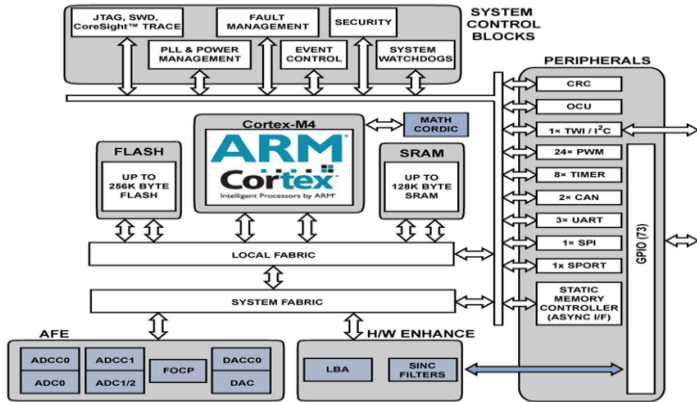
- 3 buah UARTs (serial ports)
- SPI, I2C, I2S, IR modulator
- I2S (for *high quality audio interface*)
- *Real Time Clock (with user-added 32.768 crystal and battery)*
- 16 *general purpose DMA channels (separate from USB)*
- *Touch Sensor Inputs*



Gambar 2. 9 Tampak atas Teensy Board 3.2 [13]

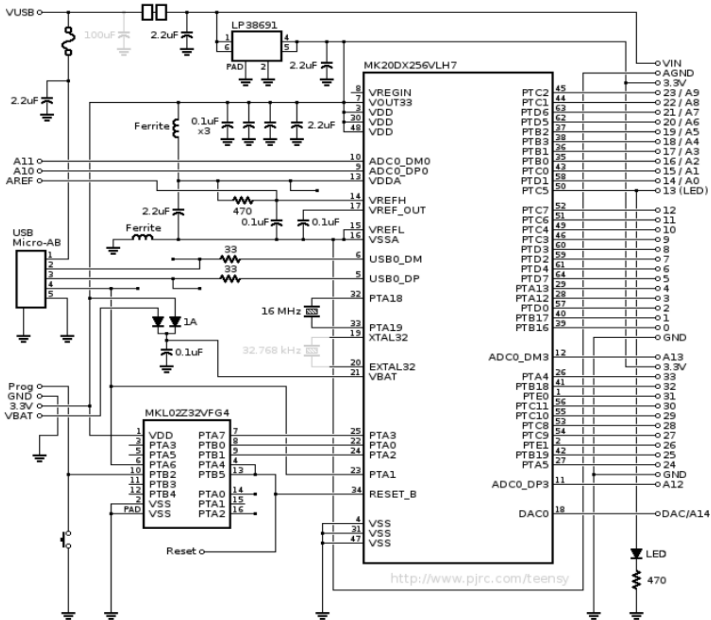


Gambar 2. 10 Tampak bawah Teensy Board 3.2 [13]



Gambar 2. 11 Diagram Blok ARM Cortex-M4 [13]

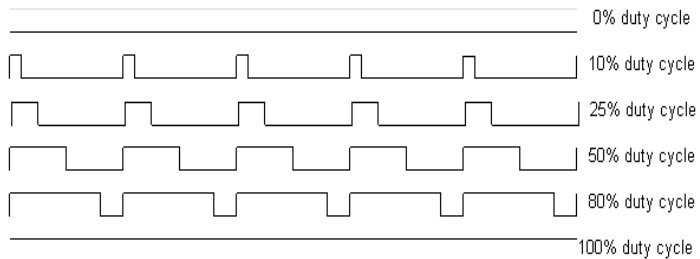
Berikut adalah minimum sistem dari teensy board 3.2:



Gambar 2. 12 Schematic Teensy Board 3.2 [13]

2.4.1 PWM (*Pulse Width Modulation*)

PWM adalah suatu teknik modulasi sinyal dengan mengubah *duty cycle* atau rasio waktu dari pulsa sinyal yang “ON” pada suatu periode pulsa. Sinyal PWM membutuhkan sinyal pembawa yang disebut dengan PWM *switching frequency* yang nilainya bergantung dari aplikasi yang dibutuhkan. Kelebihan modulasi sinyal PWM adalah rugi daya akibat proses *switching* yang rendah dan implementasi digital yang mudah. PWM pada umumnya digunakan sebagai sinyal penggerak motor, protokol komunikasi, hingga sebagai sinyal kontrol.

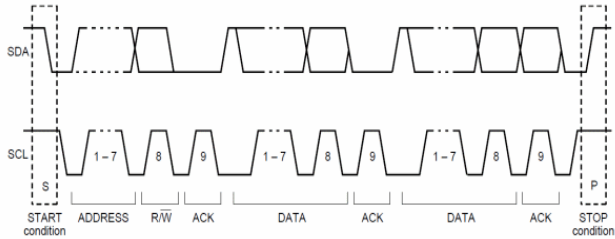


Gambar 2. 13 Modulasi PWM [14]

2.4.2 Protokol Komunikasi I2C

I2C (*Inter-Integrated Circuit*) merupakan sebuah protokol komunikasi serial elektronik yang mempunyai kemampuan untuk *multi-master* dan *multi-slave*. Mode *multi-master* dan *multi-slave* dapat digunakan untuk berkomunikasi dengan banyak perangkat dengan menggunakan *slave address* untuk masing-masing perangkat yang digunakan sebagai *slave*. Perangkat yang berfungsi sebagai *master* mengirimkan *clock* dan dapat memilih untuk berkomunikasi dengan menggunakan alamat perangkat pada perangkat *slave*, namun tidak sebaliknya.

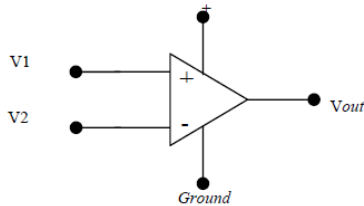
I2C berkomunikasi dengan dua buah *port* SDA (*Serial Data Line*) dan SCL (*Serial Clock Line*) yang masing-masing berfungsi sebagai jalur data dan kecepatan *bus* komunikasi yang pada umumnya mempunyai kecepatan 100 kbit dan 400 kbit per detik.



Gambar 2. 14 *Timing diagram I2C* [15]

2.5. Penguat Operasional

Penguat operasional adalah rangkaian terpadu (IC) yang mempunyai 5 buah terminal dasar. Dua terminal untuk catu daya, 2 yang lain digunakan untuk isyarat masukan yang berupa masukan membalik (-) dan masukan tak membalik (+) serta 1 terminal untuk keluaran.



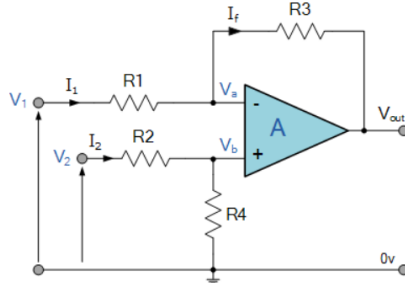
Gambar 2. 15 Rangkaian dasar *operational amplifier* [16]

2.5.1 *Differential Amplifier*

Penguat differensial dalam suatu penguat operasional (operational amplifier) dibuat menggunakan kopling langsung (*DC kopling*) yang bertujuan untuk menghilangkan efek yang ditimbulkan akibat penambahan atau pemasangan kapasitor bypass maupun kapasitor kopling. Penggunaan kopling DC pada penguat differensial ini bertujuan untuk menghindari permasalahan perlambatan yang terjadi akibat pengisian muatan pada kapasitor-kapasitor kopling (penggandeng) oleh tegangan sumber DC, dengan demikian titik kerja DC untuk mencapai titik stabil diperlukan juga waktu tunda (*time constant*). Sehingga mengakibatkan terjadinya efek kenaikan

batas frekuensi bawah (f_L) karena adanya kenaikan waktu untuk mencapai stabil (*time constant*) yang lebih lambat.

Karena penguat pasangan differensial didalamnya terdiri dari dua buah transistor, maka untuk mendapatkan titik kerja DC yang simetris, diperlukan dua buah transistor yang mempunyai konfigurasi bentuk phisis dengan karakteristik yang sama. Sedangkan untuk menghindari akibat pengaruh adanya perubahan temperatur yang berbeda pada kedua transistor tersebut, sebaiknya cara pemasangan kedua transistor adalah dibuat sedemikian rupa agar sedapat mungkin berpasangan-berhimpit satu sama lainnya.



Gambar 2. 16 *Differential Amplifier* [16]

$$V_{out} = -\frac{R_2}{R_1} \cdot V_1 + \left(1 + \frac{R_2}{R_1}\right) \cdot \left(\frac{R_4}{R_3 + R_4}\right) \cdot V_2 \quad (2.16)$$

Ketika nilai resistansi $R_1 = R_3$ dan $R_2 = R_4$ maka:

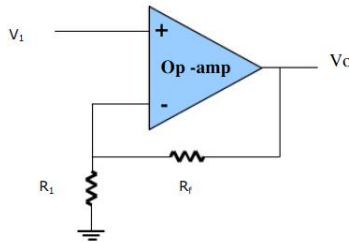
$$V_{out} = \frac{R_2}{R_1} \cdot (V_2 - V_1) \quad (2.17)$$

Penguat differensial pada Op Amp mempunyai karakteristik yang sama dengan penguat tunggal emitor bersama (*common emitter*), maka didalam analisa titik kerja DC maupun analisa sinyal bolak balik pada dasarnya mengacu pada rangkaian emitor bersama.

2.5.2 *Non Inverting Amplifier*

Non-Inverting amplifier merupakan penguat sinyal dengan karakteristik dasat sinyal *output* yang dikuatkan

memiliki fasa yang sama dengan sinyal *input*. Penguat *non inverting amplifier* dapat dibangun menggunakan penguat operasional, karena penguat operasional memang didesain untuk penguat sinyal baik membalik ataupun tak membalik. Rangkaian penguat tak membalik ini dapat digunakan untuk memperkuat isyarat AC maupun DC dengan keluaran yang tetap sefase dengan sinyal *input*. Impedansi masukan dari rangkaian penguat tak membalik (*non inverting amplifier*) berharga sangat tinggi dengan nilai impedansi sekitar 100 Mohm.



Gambar 2. 17 *Non Inverting Amplifier* [16]

$$V_1 = \frac{R_1}{R_1 + R_f} \cdot V_o \quad (2.18)$$

$$\frac{V_o}{V_1} = \frac{R_1 + R_f}{R_1} = 1 + \frac{R_f}{R_1} \quad (2.19)$$

2.6. Sensor

Sensor adalah alat untuk mendeteksi atau mengukur sesuatu, yang digunakan untuk mengubah variasi mekanis, magnetis, panas, sinar dan kimia menjadi tegangan dan arus listrik. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh kontroler sebagai otaknya.

Sensor dalam teknik pengukuran dan pengaturan secara elektronik berfungsi mengubah besaran fisik (misalnya: temperatur, gaya, kecepatan putaran) menjadi besaran listrik yang proposional. Sensor dalam teknik pengukuran dan pengaturan ini harus memenuhi persyaratan-persyaratan kualitas yaitu:

- a. Linieritas

konversi harus benar – benar proposional, jadi karakteristik konversi harus linear.

b. Tidak tergantung temperatur

keluaran konverter tidak boleh tergantung pada temperatur sekelilingnya, kecuali sensor suhu.

c. kepekaan

kepekaan sensor harus dipilih sedemikian rupa, sehingga pada nilai – nilai masukan yang ada dapat diperoleh tegangan listrik keluaran yang cukup besar.

d. Waktu tanggapan

Waktu tanggapan adalah waktu yang diperlukan keluaran sensor untuk mencapai nilai akhirnya pada nilai masukan yang berubah secara mendadak. Sensor harus dapat berubah cepat bila nilai masukan pada sistem tempat sensor tersebut berubah.

e. batas frekuensi terendah dan tertinggi

batas – batas tersebut adalah nilai frekuensi masukan periodik terendah dan tertinggi yang masih dapat dikonversi oleh sensor secara benar. Pada kebanyakan aplikasi disyaratkan bahwa frekuensi terendah adalah 0Hz.

f. Stabilitas waktu

Untuk nilai masukan tertentu sensor harus dapat memberikan keluaran yang nilainya tetap dalam waktu yang lama.

g. Histerisis

Gejala histerisis yang ada pada magnetisasi besi dapat pula dijumpai pada sensor. Misalnya, pada suatu temperatur tertentu sebuah sensor dapat memberikan keluaran yang berlainan.

Empat sifat diantara syarat – syarat diatas, yaitu linieritas, ketergantungan pada temperatur, stabilitas waktu dan histerisis menentukan ketelitian sensor.

2.7. ADC (*Analog to Digital Converter*)

ADC adalah suatu piranti yang dirancang untuk mengubah sinyal - sinyal analog menjadi bentuk sinyal digital. Atau dapat pula disimpulkan ADC dapat merubah nilai suatu masukan yang berupa tegangan listrik dalam bentuk voltase atau sinyal analog lainnya menjadi keluaran berupa nilai digital. Dalam penggunaannya ADC

adalah tegangan maksimum yang dikonversikan dari rangkaian pengkondisian sinyal, resolusi, ketepatan dan waktu konversinya.

ADC internal pada *microcontroller Teensy Board 3.2* merupakan ADC 12-bit yang bernilai 4.096. ADC internal ini digunakan untuk *input* dari sensor serta masukan atau data lain yang digunakan, seperti SSR (*Solid State Relay*). Prinsip kerja ADC sensiri adalah memberi tegangan masukan pada *port* yang tersedia pada *Teensy Board*. ADC memerlukan tegangan sebesar 3.3v pada pin Vcc(Vref) dan (GND) dihubungkan dengan *ground*. Dari beberapa indikasi yang dimiliki oleh ADC kita dapat mengetahui berapa tegangan yang dibutuhkan oleh ADC untuk menaikkan 1 *byte*. Persamaan untuk resolusi ADC adalah:

$$\text{Resolusi} = Q = \frac{V_{ref}}{2^n} \quad (2.20)$$

Dengan:

- Q : Resolusi
- Vref : Tegangan referensi
- N : jumlah bit keluaran

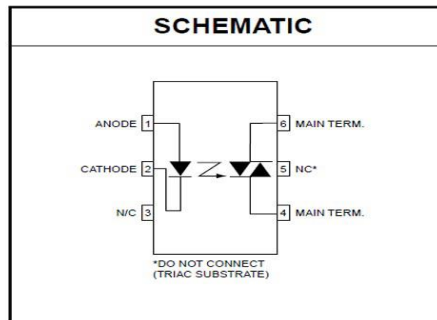
Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Sebagai contoh: ADC 8 *bit* akan memiliki *output 8bit* data digital, ini berarti sinyal *input* dapat dinyatakan dalam 255 ($2^n - 1$) nilai diskrit. ADC 12 *bit* memiliki 12*bit output* data digital, ini berarti sinyal *input* dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 *bit* akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 *bit*.

ADC (*Analog to Digital Converter*) memiliki 2 karakter prinsip, yaitu kecepatan sampling dan resolusi. Kecepatan sampling suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan sampling biasanya dinyatakan dalam *sample per second* (SPS).

Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal *input* dan tegangan referensi. Sebagai contoh, bila tegangan referensi (Vref) 5 volt, tegangan *input* 3volt, rasio *input* terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar $60\% \times 255 = 153$ (bentuk decimal) atau 10011001 (bentuk biner).

2.8. SSR (*Solid State Relay*)

Berbeda dengan relai elektromagnetis, SSR (*solid state relay*) merupakan relai yang tidak memiliki kumparan dan kontak sesungguhnya, sebagai penggantinya digunakan alat penghubung semikonduktor seperti transistor *bipolar*, *mosfet*, *scr*, atau *triac*. Relai *solid state* tidak memiliki bagian yang berputar, relai tersebut tahan terhadap guncangan dan getaran serta tertutup rapat terhadap kotoran dan kelembaban.



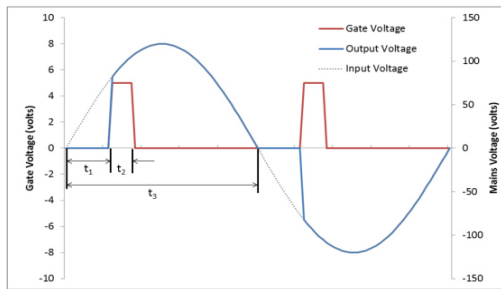
Gambar 2. 18 Rangkaian Opto triac [17]

Tegangan kontrol untuk SSR dapat arus searah atau bolak balik, dan biasanya berkisar antara 3 sampai 32 *volt* untuk versi dc dan 80 sampai 280V untuk versi ac. Pada sebagian besar aplikasi, SSR digunakan sebagai perantara antara rangkaian kontrol tegangan rendah dengan tegangan line ac yang lebih tinggi.

SSR memiliki beberapa keuntungan dibandingkan dengan relai elektromagnetis. SSR lebih dipercaya mempunyai umur pemakaian yang lebih panjang karena SSR tidak memiliki bagian yang bergerak, dapat digabungkan dengan rangkaian transistor dan sikuit IC, serta tidak banyak menimbulkan interferensi elektromagnetis. SSR lebih tahan terhadap guncangan dan terhadap getaran serta memiliki waktu respon yang lebih cepat.

Zero crossing detector merupakan rangkaian elektronika yang berfungsi untuk mendeteksi titik persilangan nol pada suatu sinyal ac baik sinus maupun sinyal ac lainnya. Rangkaian *zero crossing detector* sering diaplikasikan pada peralatan yang digunakan untuk

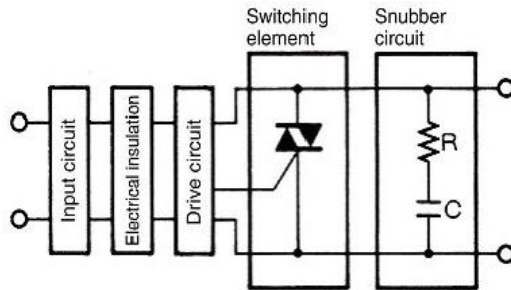
mengendalikan beban pada jaringan listrik AC dan menggunakan eksekutor berupa *TRIAC* atau *SCR*. Titik persilangan dengan nol tegangan sumber untuk beban yang dikendalikan dengan komponen saklar berupa *TRIAC* atau *SCR* diperlukan untuk menentukan waktu mulai pemberian *trigger* atau sinyal kontrol pada *SCR* atau *TRIAC* tersebut. Pemberian sinyal *input* pada *SCR* atau *TRIAC* yang tepat pada titik persilangan nol akan meningkatkan efektifitas dan efisiensi daya *output* dari pengendalian beban listrik AC.



Gambar 2. 19 Sinyal output zero crossing detector circuit dan srr beban resistif. [17]

Dalam pengaplikasiannya pada rangkaian elektronika dengan beban induktif akan menimbulkan lonjakan tegangan yang berasal dari proses induksi beban induktor tersebut. Lonjakan yang berasal dari proses induksi dari beban induktor memiliki level tegangan yang cukup besar sehingga lonjakan tegangan dari beban induktif ini dapat mengganggu kinerja rangkaian yang lain. Terjadinya lonjakan tegangan induktif pada beban induktor ini terjadi pada saat beban induktor tersebut pertama kali diberikan sumber tegangan (ON) dan pada saat tegangan sumber diputuskan (OFF).

Pada rangkaian dengan beban induktif diperlukan komponen atau rangkaian untuk memproteksi dari terjadinya lonjakan tegangan induktif tersebut agar tidak mempengaruhi kinerja rangkaian. Untuk memproteksi rangkaian dari kondisi ini (terjadinya tegangan induksi) dapat dilakukan dengan memasang dioda atau 18 kapasitor dan resistor paralel pada beban induktif tersebut seperti terlihat pada gambar berikut:



Gambar 2. 20 Rangkaian Snubber komponen *switching* [17]

Output dari optoisolator akan mengendalikan *TRIAC* yang dihubungkan secara seri dengan beban pada tiap fasa. Optoisolator ini hanya akan beroperasi pada posisi *zero voltage crossing* bahkan jika *switching* berada pada tegangan puncak, beban akan kembali *ON* setelah melewati *zero voltage crossing*. Dari penerapan *ZVS* tersebut akan menghasilkan efisiensi yang lebih baik saat proses *switching*. *ZVS* akan meningkatkan umur pemakaian dari kontroler serta beban dan mengurangi kemungkinan pembentukan busur di relai. Tanpa *ZVC*, optoisolator dapat beroperasi, namun pada tegangan yang kuat dapat mengakibatkan busur api (*spark*).

SSR dapat diaplikasikan dalam berbagai peralatan yaitu antara lain mesin kemasan, perkakas mesin, peralatan elektronik, peralatan *manufacturing*, sistem uji dan keamanan, sistem instrumentasi, peralatan proteksi, kontrol *power on-board*, kontrol lalu lintas, kontrol elevator, trafo, *valve*, motor, solenoid.

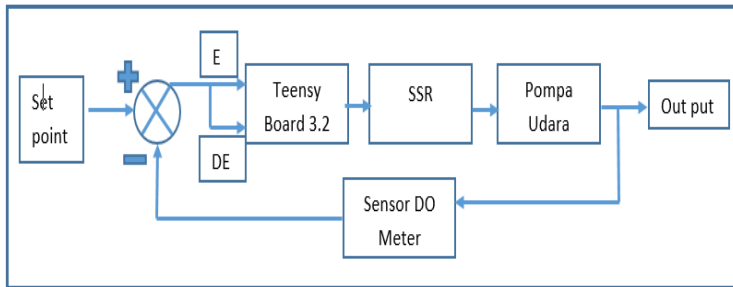
Halaman ini sengaja dikosongkan

BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai tahapan dalam rancangan desain sistem melingkupi perancangan perangkat keras dan lunak sistem secara detail.

3.1. Desain Sistem Keseluruhan

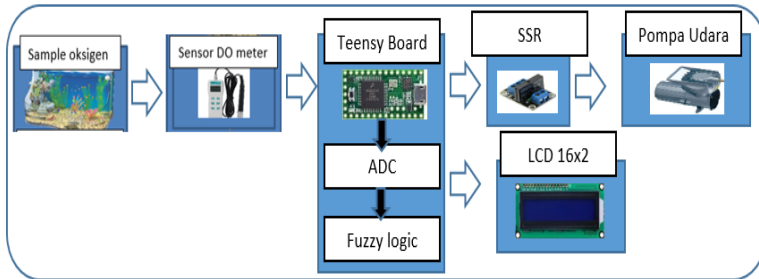
Perancangan sistem dapat dibagi menjadi 3 buah bagian yaitu sensor, kontroler *fuzzy*, dan aktuator. Pada gambar 3.1 menunjukkan diagram blok sistem. *input* dari sistem berupa nilai *setting point*, diagram blok terdiri dari beberapa bagian yaitu teensy board sebagai pengolah data, proses ADC dan *fuzzy logic* dilakukan didalam mikrokontroler teensy board. *Output* data dari *teensy board* akan memberi sinyal PWM pada modul SSR yang berguna untuk mengaktifkan pompa udara untuk menghasilkan kadar oksigen terlarut. Ketika level oksigen terlarut di bawah nilai *setting point* maka aerator akan dihidupkan untuk menghasilkan kadar oksigen terlarut. Ketika level oksigen di atas nilai *setting point* maka aerator akan di nonaktifkan. Berikut di bawah ini adalah diagram blok sistem keseluruhan:



Gambar 3. 1 Diagram blok sistem

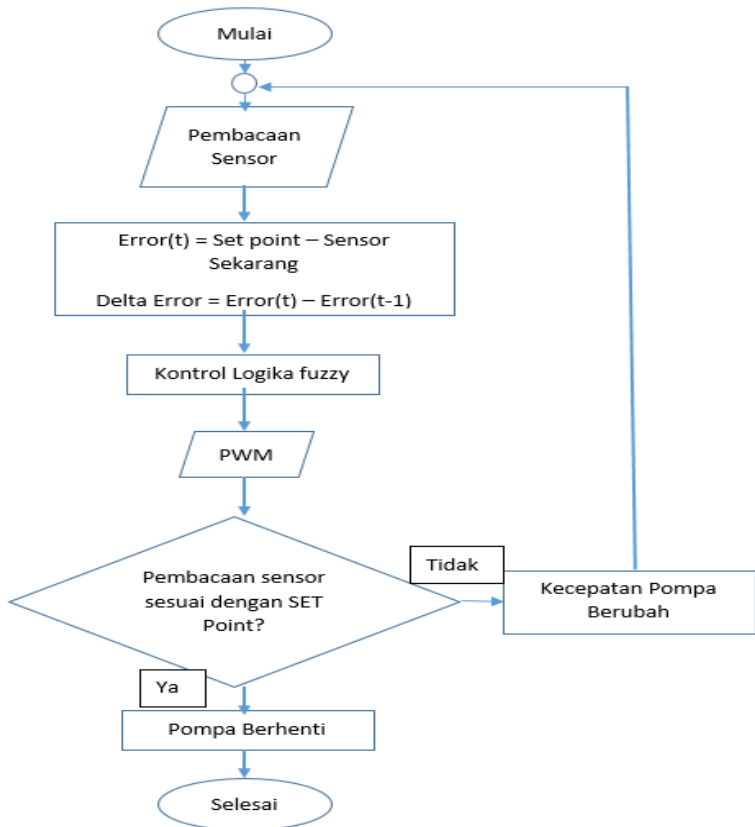
Sistem yang diajukan untuk penelitian Tugas Akhir ini menggunakan *Teensy Board 3.2* sebagai unit akuisisi data, pemrosesan data, dan pengekseskusi kontrol logika *fuzzy*. Pada gambar 3.2 merupakan diagram blok perangkat keras sistem. Perangkat keras sistem terdiri dari sensor DO-AZ8403 yang berfungsi mengukur kadar oksigen di dalam air, yang kedua terdapat *teensy board 3.2* sebagai

alat untuk mengolah data, yang ketiga modul SSR sebagai *relay* dengan *input* PWM untuk mengontrol pompa udara, yang keempat adalah LCD 16x2 berfungsi sebagai monitor untuk menampilkan kadar oksigen terlarut di dalam air, dan yang terakhir adalah pompa udara sebagai alat untuk menghasilkan kadar oksigen terlarut di dalam air. Berikut adalah diagram blok perangkat keras sistem:



Gambar 3. 2 Diagram blok perangkat keras sistem

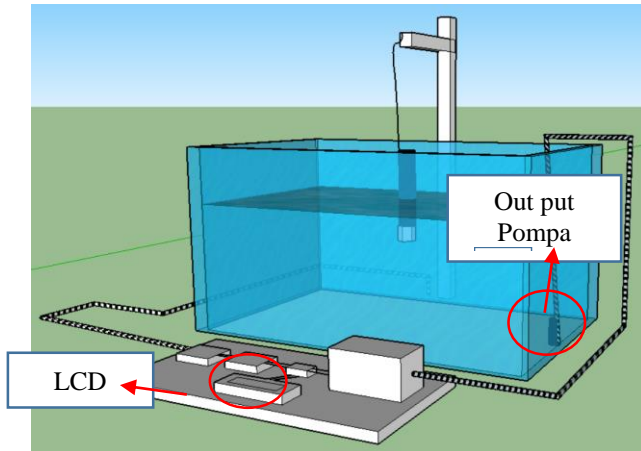
Sistem yang diajukan akan mengontrol kadar oksigen dalam air untuk menghindari kekurangan oksigen pada pembudidayaan ikan. Sistem akan mengontrol kadar oksigen dengan menghidupkan pompa udara sebagai aktuator yang menghasilkan gelembung udara di dalam air. Pada gambar 3.3 merupakan gambar alur kerja sistem. Langkah awal yang dilakukan sistem adalah membaca kadar oksigen terlarut, kadar oksigen terlarut yang berupa data analog akan diolah oleh analog to digital *converter* pada mikrokontroler *teensy board* yang memiliki resolusi 10bit. Data digital yang dihasilkan ADC akan masuk pada proses *fuzzy* dimana *error* dan $\Delta error$ menjadi himpunan masukan. *Error* adalah nilai *setting point* dikurang pembacaan sensor sekarang dan nilai $\Delta error$ adalah nilai *error* sekarang dikurang dengan *error* sebelumnya. nilai *error* dan $\Delta error$ akan masuk pada proses fuzzifikasi dimana nilai *crisp* diubah dalam bentuk himpunan *fuzzy*, selanjutnya himpunan *fuzzy* akan ditentukan oleh *rule* yang sudah dibuat, dan tahap akhir dari *fuzzy* adalah proses defuzzifikasi yang merupakan *output* dari *membership function*. Berikut adalah alur kerja sistem:



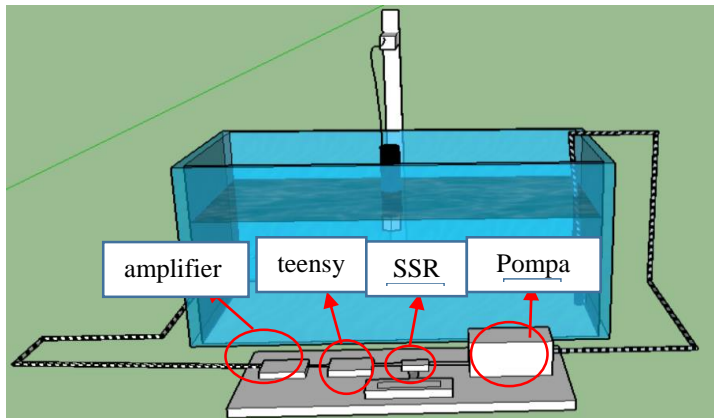
Gambar 3. 3 Gambar alur kerja sistem

A Desain Fisik Sistem

Sistem kontrol oksigen yang terdiri dari akuarium dan rangkaian elektronik yang di letakkan pada sebuah wadah akrilik. gambar 3.4 dan 3.5 adalah desain fisik sistem yang terdiri dari 5 perangkat keras. Berikut adalah desain fisik dari sistem:



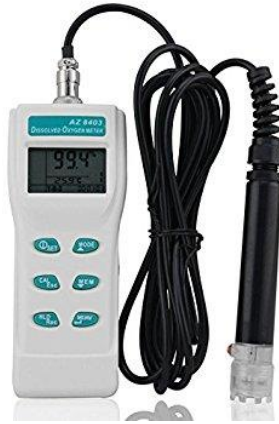
Gambar 3. 4 Desain fisik Sistem dilihat dari samping



Gambar 3. 5 Desain Fisik Sistem dilihat dari depan

3.2. Sensor *Dissolved Oxygen AZ-8403*

Sensor digunakan untuk mengetahui besaran parameter yang akan menjadi masukan untuk kontrol logika *fuzzy* dan pengukuran kadar oksigen di dalam air. Sensor DO AZ8403 untuk mengukur kadar oksigen terlarut.



Gambar 3. 6 Dissolved Oxygen AZ-8403

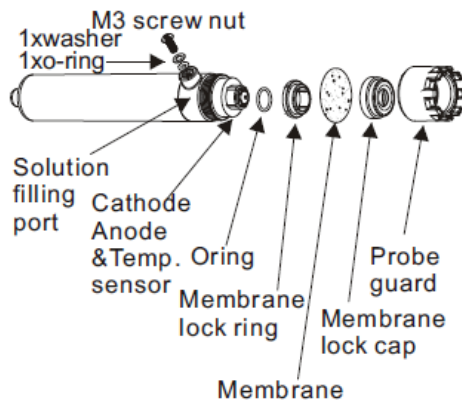
A Probe Dissolved Oxygen

Tingkat oksigen di dalam air harus tetap dijaga. Jika level oksigen terlarut terlalu rendah, maka ikan akan sulit untuk bernafas, bakteri pengurai akan mati, dan penguraian akan berhenti. Jika oksigen terlalu tinggi energi akan dibuang dalam aerasi air. Di lain pihak aplikasi dalam industri seperti *boiler* dibutuhkan tingkat oksigen terlarut yang rendah untuk menghindari dari korosi. *Dissolved oxygen* tidak dapat disahkan dari besar pH dan salinitasi.

Ada 2 teknik dasar dalam pengukuran oksigen terlarut yaitu *galvanic* dan *polarographic*, kedua *probe* menggunakan sistem *electrode* dimana DO bereaksi dengan katoda untuk menghasilkan arus. Jika bahan *electrode* dipilih beda potensial -0.5 volt atau lebih besar katoda dan anoda. Sumber eksternal tidak dibutuhkan dan disebut *galvanic*. Jika sumber luar dipakai maka disebut *polarographic*.

1. *Galvanic probe* lebih stabil dan lebih akurat DO level rendah
2. *Polarographic probes* memerlukan pengisian ulang setiap beberapa minggu

Sensor *galvanic probe* terdiri dari dua elektrode, yaitu anoda dan katoda yang keduanya dicelupkan ke dalam elektrolit (berada di dalam bodi sensor). Oksigen menembus membran memisahkan anoda dan katoda dari air. Oksigen berdifusi melewati membran dan berinteraksi dengan *probe* internal untuk menghasilkan arus, besar tekanan oksigen yang berdifusi melewati membran berbanding lurus dengan arus yang dihasilkan. Dalam kenyatannya keluaran dari sensor adalah 0.6 volt dengan kenaikan 1mv untuk setiap kenaikan 0.01 ppm. Keluarannya juga akan menghasilkan arus yang melewati thermistor resistor yang merubah keluaran menjadi *temperature*.



Gambar 3. 7 *Dissolved oxygen probe*

Katoda adalah elektrode yang bermuatan negatif. *Electrolyte* mengelilingi pasangan elektrode dan dimasukkan oleh membran. Dengan tanpa oksigen katoda berpolarisasi dengan hidrogen dan tahanan aliran arus. Ketika oksigen melewati membran, katoda dipolarisasi dan elektron diambil. Elektrokimia pada katoda mengurangi oksigen dijadikan ion hidroksi.

Pasangan elektrode mengalirkan arus secara langsung secara proporsional ke sistem, sejumlah oksigen akan masuk sehingga memberikan secara langsung ukuran jumlah oksigen yang masuk ke *probe*. Karena oksigen yang masuk adalah berupa kimia, tekanan parsial yang ada melintasi membran dan rata – rata oksigen yang masuk ke dalam *probe* adalah fungsi tekanan parsial oksigen di udara atau air terukur.

Karena tekanan parsial *dissolved oxygen* adalah fungsi *temperature* dari sampel, *probe* harus dikalibrasi sesuai dengan *temperature* sampel yang terukur. Perhitungan kadar oksigen terlarut nya:

$$DO = \frac{B}{B-2} \times 5,6 \times 10 \times N \times V \quad (3.1)$$

Dimana:

B = volume botol sampel BOD = 250ml

B-2 = volume air dalam botol sampel setelah ditambah 1ml larutan MnCl₂ dan 1 ml NaOH - KI

5,6 = konstanta yang sama dengan ml oksigen ~ 1 mgrek tiosulfat

10 = volume K₂Cr₂O₇ 0,01 N yang ditambahkan

N = normalitas tiosulfat

V = volume tiosulfat yang dibutuhkan untuk titrasi

A Karakteristik Membran *Probe Dissolved Oxygen*

- Ph tidak mempengaruhi kemampuan DO *probes*.
- *Chlorine* dan *hydrogen sulfide*(H₂S) dapat menyebabkan pembacaan yang salah.
- Tekanan atmosfer mempengaruhi saturasi oksigen oleh DO *probes* dan harus dikalibrasi untuk tekanan barometric ketika pembacaan dalam ppm atau mg/L.
- Ketebalan membran menentukan tingkatan *output* pembacaan dan alat ukur.

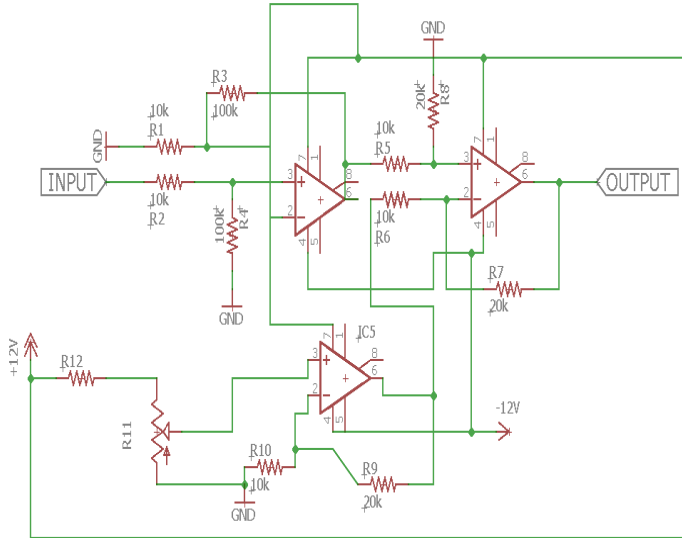
Spesifikasi DO Meter 8403:

Tabel 3. 1 Spesifikasi DO meter 8403

Do. %	0.0 ~ 199.9%
Res.:	0.1%
Accuracy:	<1.5% Of F. S
D.O. Ppm:	0.00 ~ 19.99ppm
Res.:	0.01ppm
Accuracy:	<1.5% Of F. S
Temp.:	0~30 °c,0~50 °c (8402/8403)
Res.:	0.1 °c
Accuracy:	<5 °c, <3 °c (8402/8403)
Salinity:	0.0 ~ 50.0 Ppt (8402/8403)
Res. :	0.1ppt
Pressure.:	500~1499mmhg/L 66.6~199.9kpa (8402/8403)
Res.:	1mmhg/0.1kpa
Memory:	99point (8403)
Irda PORT:	YES (8403)
Rs232:	YES (8401/8402/8403)
Real Time	Yes (8403)

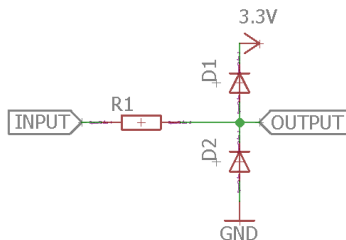
3.3. Perancangan Rangkaian Penguat Operasional

Rangkaian penguat operasional berfungsi untuk mengolah sinyal dari sensor DO meter AZ-8403 berupa tegangan sebesar 0,6 volt dengan kenaikan tegangan 1mv untuk setiap 0,01ppm. Untuk memperbesar *range* pengukuran dengan *output* tegangan maksimal 3.3v diperlukan 3 rangkaian penguat operasional.



Gambar 3. 8 Rangkaian penguat operasional

Selanjutnya adalah rangkaian pengaman yang berfungsi untuk memastikan keamanan rangkaian dan keamanan penggunaannya. Untuk itu diperlukan beberapa pengaman yang dibuat dari dioda yang dipasang seperti gambar 3.9.



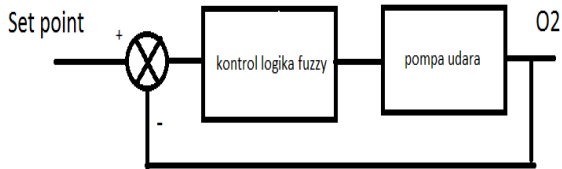
Gambar 3. 9 Rangkaian pengaman

Rangkaian ini terdiri dari satu resistor dan dua dioda, dimana resistor digunakan untuk membatasi arus, sedangkan dioda berfungsi sebagai pembatas tegangan. Apabila tegangan masukan berada diantara batas atas yaitu 3.3volt dan batas

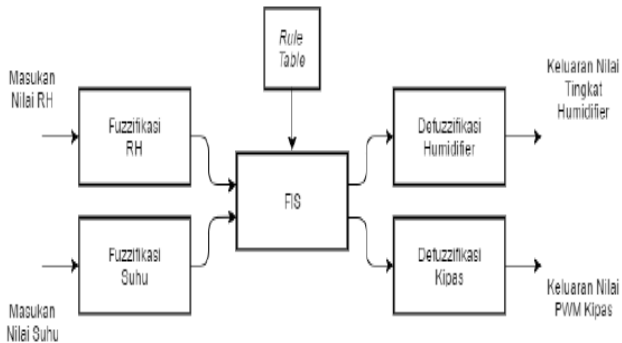
bawah 0.7volt , maka sinyal akan dilewatkan. Apabila sinyal berada diatas batas atas maka dioda positif menjadi *forward bias* dan dioda negatif menjadi menjadi *reverse bias* sehingga arus mengalir ke dioda positif. Sebaliknya, apabila sinyal berada pada batas bawah maka dioda positif menjadi *reverse bias* dan dioda negatif menjadi *forward bias* sehingga arus mengalir ke dioda negatif.

3.4. Kontrol Fuzzy Logic

Kontroler *logika fuzzy* digunakan sebagai metode kontrol untuk penghasil udara pada pompa air yang menggunakan parameter kadar oksigen terlarut dari sensor DO AZ-8403. Berikut pada gambar 3.10 adalah diagram blok sistem kontrol dengan kontrol logika *fuzzy*:



Gambar 3. 10 Diagram blok kontrol sistem



Gambar 3. 11 Diagram blok kontrol logika *fuzzy*

Terdapat satu masukan *fuzzy* yang berupa bilangan *crisp*, yaitu kadar oksigen di dalam air. Nilai masukan yang berupa bilangan *crisp* akan dirubah menjadi bilangan *fuzzy* dengan proses fuzzifikasi.

Pada Gambar 3.1 *input fuzzy* berupa *error* dan $\Delta error$. Nilai diperoleh dari:

$$Error \quad : \text{Set point} - \text{Input sensor} \quad (3.2)$$

$$\Delta error \quad : \text{Error sekarang} - \text{Error Sebelumnya} \quad (3.3)$$

Pada fuzzifikasi ini, input *error* dan $\Delta error$ masing – masing dibagi dalam fungsi keanggotaan sebagai berikut:

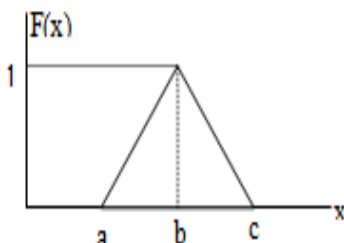
- *Error* : NB (Negatif besar), Zero (nilai nol), PK (Positif kecil), PB (Positif besar).
- $\Delta error$: NEG (Negatif), NOL, POS (Positif)

Sedangkan tipe keanggotaan *fuzzy* yang digunakan dalam proyek akhir ini ada 2 yaitu segitiga dan trapesium. Berikut ini persamaan dari segitiga dan trapesium:

a. Tipe keanggotaan segitiga:

$$F(x; a, b, c) = \begin{cases} 0 & x < a \\ (x - a)/(b - a) & a \leq x < b \\ (c - x)/(c - b) & b \leq x < c \\ 0 & x \geq c \end{cases} \quad (3.4)$$

Dengan nilai x, a, b, dan c diperoleh dari gambar 3.12 berikut:



Gambar 3. 12 Tipe keanggotaan segitiga

Pada gambar 3.13 merupakan *list program teensy* pada arduino hasil konversi rumus segitiga himpunan *fuzzy*.

```

// Triangular Member Function
FIS_TYPE fis_trmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}

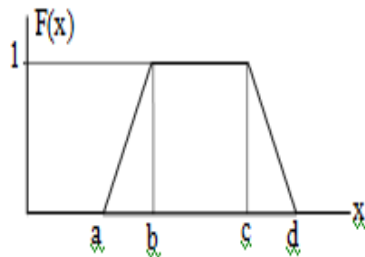
```

Gambar 3. 13 Program fungsi himpunan segitiga

b. Tipe keanggotaan trapesium:

$$F(x; a, b, c) = \begin{cases} 0 & x < a \\ (x - a)/(b - a) & a \leq x < b \\ 1 & b \leq x < c \\ (c - x)/(c - d) & c \leq x < d \\ 0 & x \geq d \end{cases} \quad (3.5)$$

Dengan nilai x, a, b, dan c diperoleh dari gambar 3.14 berikut:



Gambar 3. 14 Tipe keanggotaan trapesium

Pada gambar 3.15 merupakan *list program teensy* pada arduino hasil konversi rumus trapesium himpunan *fuzzy*.

```

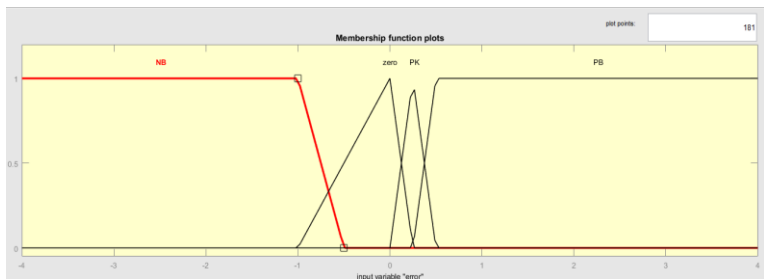
// Trapezoidal Member Function
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) / (d - c)) : 0)));
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) / (b - a)) : 0)));
    return (FIS_TYPE) min(t1, t2);
}

```

Gambar 3. 15 Program fungsi himpunan trapesium

3.4.1 Himpunan Masukan *Fuzzy*

Terdapat dua masukan *fuzzy* yang berupa bilangan *crisp*, yaitu nilai *error* dan $\Delta error$. Nilai masukan yang berupa bilangan *crisp* akan dirubah menjadi bilangan *fuzzy* dengan proses fuzzifikasi. Nilai masukan *error* dan $\Delta error$ *fuzzy* merupakan representasi nilai kesalahan pembacaan sensor dari nilai *set point* yang telah ditentukan.



Gambar 3. 16 Grafik fungsi keanggotaan nilai *error*

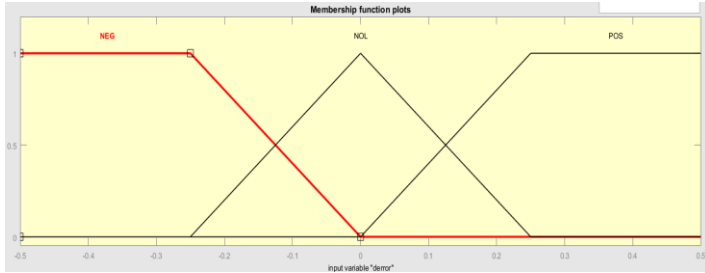
Pada gambar 3.17 merupakan *list program teensy* pada arduino untuk himpunan masukan.

```

FIS_TYPE fis_gMPIOcoeff1[] = { -5, -5, -1, -0.5 };
FIS_TYPE fis_gMPIOcoeff2[] = { -1, 0, 0.25 };
FIS_TYPE fis_gMPIOcoeff3[] = { 0, 0.25, 0.5 };
FIS_TYPE fis_gMPIOcoeff4[] = { 0.25, 0.5, 5, 5 };
FIS_TYPE* fis_gMPIOcoeff[] = { fis_gMPIOcoeff1, fis_gMPIOcoeff2, fis_gMPIOcoeff3, fis_gMPIOcoeff4 };

```

Gambar 3. 17 Program fungsi input *error*



Gambar 3. 18 Grafik Fungsi keanggotaan nilai $\Delta Error$

Pada gambar 3.19 merupakan *list program teensy* pada arduino untuk himpunan masukan.

```
FIS_TYPE fis_gMFI1Coeff1[] = { -0.5, -0.5, -0.25, 0 };
FIS_TYPE fis_gMFI1Coeff2[] = { -0.25, 0, 0.25 };
FIS_TYPE fis_gMFI1Coeff3[] = { 0, 0.25, 0.5, 0.5 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2, fis_gMFI1Coeff3 };
```

Gambar 3. 19 Program fungsi input $\Delta error$

Gambar 3.16 adalah grafik derajat keanggotaan himpunan *fuzzy*. Masukan *fuzzy* memiliki 4 himpunan:

- NB (Negatif Besar) : -5 - -0.5
- Zero (Nilai Nol) : -1 - 0.25
- PK (Positif Kecil) : 0 - 0.5
- PB (Positif Besar) : 0.25 - 5

Gambar 3.18 adalah grafik derajat keanggotaan himpunan *fuzzy*. Masukan *fuzzy* memiliki 3 himpunan:

- NEG: -0.5 - 0
- NOL: -0.25 – 0.25
- POS: 0 – 0.5

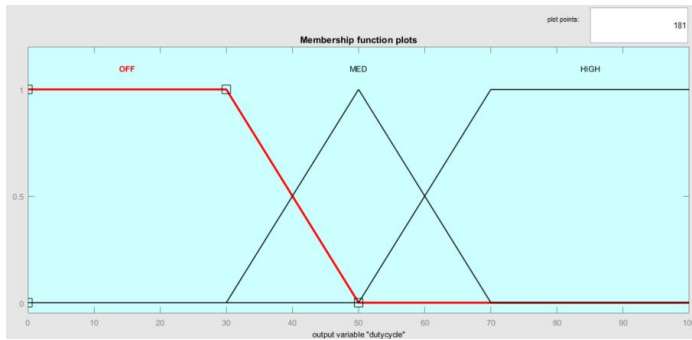
3.4.2 Himpunan Keluaran *Fuzzy*

Keluaran *fuzzy* berupa parameter kecepatan pompa air yang didapatkan dengan proses defuzzifikasi. Keluaran *fuzzy*

akan digunakan mikrokontroler untuk mengaktifkan aktuator berupa pompa udara.

Gambar adalah grafik keluaran yang berupa nilai dari kecepatan pompa. Keluaran *fuzzy* memiliki 3 himpunan, yaitu;

- *Off*
- *Medium*
- *High*



Gambar 3. 20 Out put Fuzzy Logic

Pada gambar 3.17 merupakan *list program teensy* pada arduino untuk himpunan keluaran.

```
FIS_TYPE fis_gMFO0Ccoeff1[] = { 0, 0, 30, 50 };
FIS_TYPE fis_gMFO0Ccoeff2[] = { 30, 50, 70 };
FIS_TYPE fis_gMFO0Ccoeff3[] = { 50, 70, 100, 100 };
FIS_TYPE* fis_gMFO0Ccoeff[] = { fis_gMFO0Ccoeff1, fis_gMFO0Ccoeff2, fis_gMFO0Ccoeff3 };
```

Gambar 3. 21 Program fungsi *output duty cycle* (%).

3.4.3 Fuzzy Inference System (FIS)

Sistem inferensi *fuzzy* merupakan sebuah sistem yang menggunakan bilangan *fuzzy* untuk mengolah masukan menjadi keluaran dengan menggunakan *rule table* untuk memetakan hubungan antara masukan dengan keluaran. Sistem inferensi yang digunakan pada penelitian Terdapat 12 buah kemungkinan masukan pada tahapan *if-then statement* dengan dua buah keluaran yang independen terhadap satu sama

lain. Masing-masing keluaran memiliki *rule table* yang berbeda-beda untuk memetakan keluaran yang berupa nilai PWM pompa. Tugas Akhir ini adalah berjenis Mamdani dengan proses defuzzifikasi dengan metode *centroid*. Inferensi jenis Mamdani digunakan karena bersifat intuitif, tabel aturan yang mudah dipahami, dan sederhana.

3.4.4 Fuzzy Rule

Tabel 3. 2 Rules

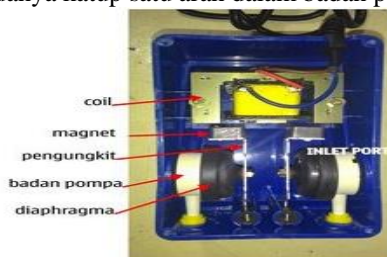
		Error			
		NB	Z	PK	PB
$\Delta Error$	NEG	OFF	OFF	MED	HIGH
	NOL	OFF	OFF	HIGH	HIGH
	POS	OFF	OFF	HIGH	HIGH

3.5. Aktuator

Aktuator digunakan untuk merubah parameter agar mencapai *set point* yang diinginkan.

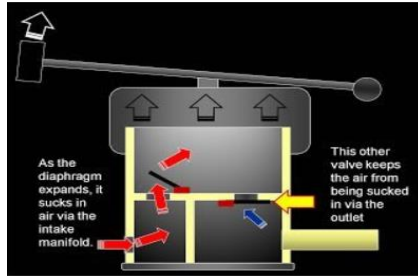
3.5.1 Pompa Udara

Cara kerja pompa udara ini adalah *coil* akan menghasilkan gaya elektromagnet yang akan menggerakkan magnet pada pengungkit yang bergerak ke kanan dan kiri sehingga akan membuat diafragma bergerak naik turun. Udara akan keluar melalui *outlet port*. Sistem ini berjalan satu arah karena adanya katup satu arah dalam badan pompa.

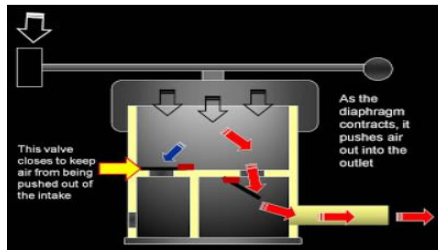


Gambar 3. 22 pompa udara

Mekanisme kerja pompa dapat digambarkan sebagai berikut:



Gambar 3. 23 Mekanisme pompa menghisap udara



Gambar 3. 24 Mekanisme pompa mengeluarkan udara

3.5.2 SSR (*Solid State Relay*)

Modul yang digunakan pada tugas akhir ini adalah OMRON G3MB-202P 2 channel sebanyak 1 buah. 1 channel menggerakkan 1 pompa udara. Gambar 3.20 menunjukkan bentuk dari modul SSR OMRON G3MB-202P.



Gambar 3. 25 OMRON G3MB-202P

Tabel 3. 3 Spesifikasi SSR OMRON G3MB-202P

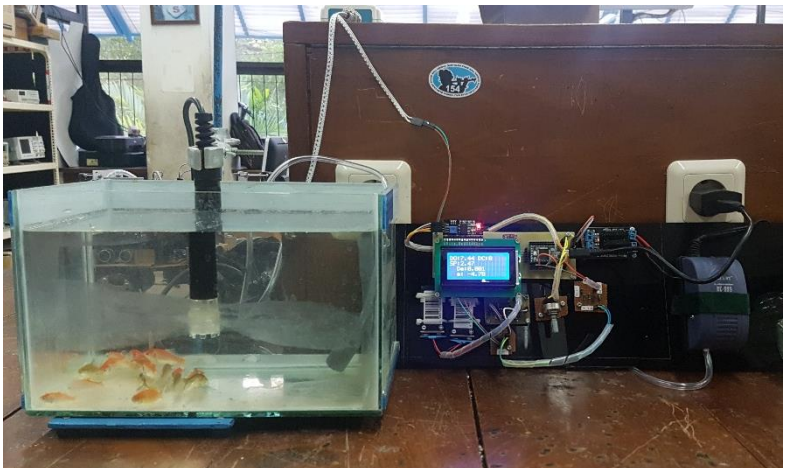
Working Voltage:	5V DC (160mA)
Channel:	2 channel
Size:	Approx.55 * 34 * 25mm / 2.2 * 1.3 * 1.0in
Weight:	Approx.21g / 0.8oz
Input control signal voltage:	(0-2.5V state low relay ON) (3.3-5V state high relay OFF)
Voltage Current:	1-Channel 12.5mA 2-Channel 22.5mA
Trigger Voltage:	0-2.5V
Trigger Current:	2mA
Quiescent Current:	0mA

BAB IV PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Bab ini bertujuan untuk mengetahui apakah tujuan dalam perancangan sistem pada Tugas Akhir ini telah terlaksana atau tidak. Cara pengujian dan pembahasan pada bab ini adalah dengan membahas tiap blok dari perancangan sistem secara keseluruhan dengan disertai tabel dan gambar yang mendukung pengujian dan pembahasan sistem.

4.1. Realisasi Desain Sistem

Pada gambar 4.1 adalah realisasi dari sistem. Terdapat sebuah akurium yang berisi ikan komet di dalamnya. Disebelah kanan akurium terdapat rangkaian yang terdiri dari 5 perangkat keras yaitu teensy 3.2, LCD 16 x 2, rangkaian amplifier, solid state relay, dan pompa udara. *Setting point* dapat diubah oleh pengguna dengan memutar potensio yang ada pada rangkaian. Sistem ini memiliki variasi *setting point* mulai dari 2.3ppm – 5.8ppm.



Gambar 4. 1 Tampak depan sistem

4.2. Pengujian ADC

Pengujian yang pertama kali adalah melakukan uji pada pembacaan data ADC pada mikrokontroler *teensy board 3.2* yang memiliki resolusi *10bit*. Hal ini di perlukan untuk keakuratan pengukuran pembacaan ADC. Dalam pengujian ini ADC diberikan *input* tegangan dengan nilai *0 – 3.3 volt*. Dan hasil pembacaan akan dicocokkan dengan dengan pembacaan multimeter

Dari pengujian didapat data pembacaan ADC sebagai berikut:

Tabel 4. 1 Pembacaan ADC

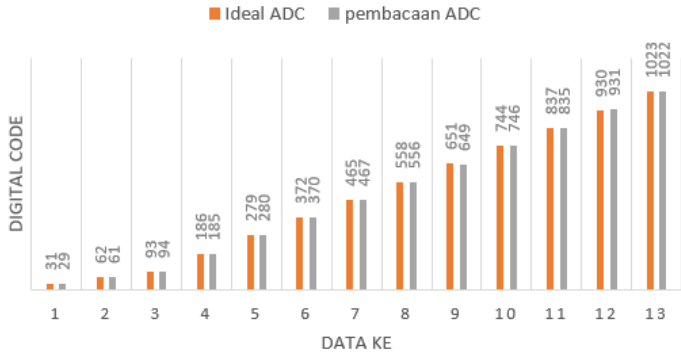
No	Vin (Volt)	Nilai ADC sebenarnya	Nilai ADC Teensy board	Error (bit)	Error (%)
1	0.1	31	29	2	0.064516
2	0.2	62	61	1	0.016129
3	0.3	93	94	1	0.010753
4	0.6	186	185	1	0.005376
5	0.9	279	280	1	0.003584
6	1.2	372	370	2	0.005376
7	1.5	465	466	1	0.002151
8	1.8	558	556	2	0.003584
9	2.1	651	649	2	0.003072
10	2.4	744	746	2	0.002688
11	2.7	837	835	2	0.002389
12	3	930	931	1	0.001075
13	3.3	1023	1022	1	0.000978

Dari persamaan ADC maka dapat diketahui *error* dari pembacaan:

$$\text{Data ADC} = \frac{V_{in} \times 2^{10-n}}{V_{ref}} \quad (4.1)$$

Diketahui *error* dari pembacaan ADC adalah 1.461 bit atau 0.0093 %

PEMBACAAN ERROR ADC



Gambar 4. 2 Pembacaan *error* ADC

4.3. Pengujian dan Kalibrasi Sensor

Pengujian *probe* sensor *dissolved oxygen* dilakukan untuk mengetahui karakteristik sensor, serta mencari nilai *output* sensor yang akan diakuisisi oleh ADC. Pengujian ini meliputi air PAM yang ditampung di akuarium dengan karakteristik sesuai dengan batasan masalah.

4.3.1 Pengujian dan Kalibrasi Oksigen Terlarut

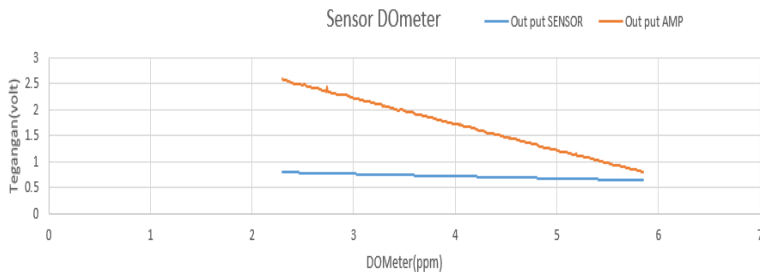
Pada pengujian ini dilakukan untuk mengetahui mengetahui nilai sensor DO AZ-8403 dan untuk proses kalibrasi. Berikut hasil pembacaan alat:

Tabel 4. 2 Nilai pengujian oksigen terlarut

Oksigen DO Meter (ppm)	Oksigen Alat (ppm)	<i>Error</i> Sensor (ppm)	<i>output</i> sensor (volt)	<i>output</i> amplifier (volt)
2.3	2.04	0.26	0.797	2.595
2.5	2.25	0.25	0.787	2.473

Oksigen DO Meter (ppm)	Oksigen Alat (ppm)	Error Sensor (ppm)	output sensor (volt)	output amplifier (volt)
2.7	2.43	0.27	0.778	2.368
3	2.72	0.28	0.765	2.224
3.5	3.17	0.33	0.744	1.978
3.7	3.35	0.35	0.735	1.875
4	3.74	0.26	0.722	1.726
4.5	4.26	0.24	0.7	1.467
4.7	4.39	0.31	0.691	1.371
5	4.76	0.24	0.678	1.224
5.2	4.87	0.33	0.669	1.116
5.5	5.22	0.28	0.657	0.971
5.85	5.59	0.26	0.642	0.814

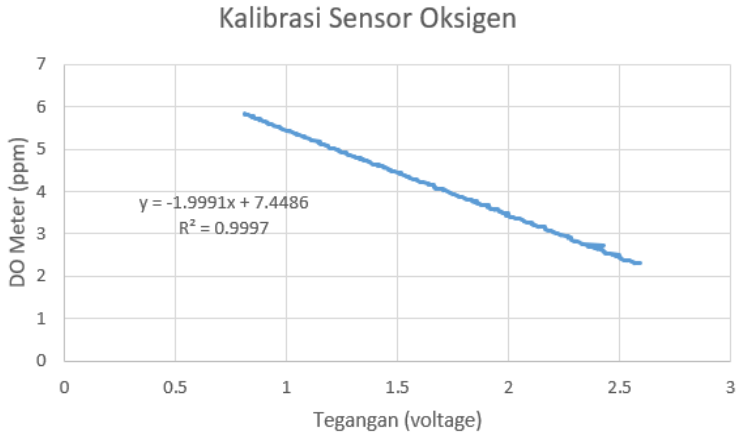
Dari pengujian pada tabel 4.2 di peroleh rata rata kesalahan pembacaan adalah 0.28ppm atau 0.075% dengan kesalahan pembacaan maksimum 0.39ppm.



Gambar 4.3 Perbandingan *output* sensor dan *amplifier*

Kalibrasi nilai oksigen terlarut dapat dilakukan dengan menggunakan regresi linier dari pembacaan data dengan nilai

minimum 2.3 ppm dan nilai maksimum 5.8ppm. Dari hasil pengujian didapatkan rumus kalibrasi sensor yang tertera pada gambar 4.4



Gambar 4. 4 Grafik pengujian dan kalibrasi sensor oksigen terlarut

4.3.2 Pengujian Awal Terhadap Saturasi Udara

Sebelum *probe* dimasukan ke dalam air, hal yang harus dilakukan adalah mencari nilai saturasi *probe* sensor terhadap udara, dengan tujuan untuk melakukan kalibrasi kadar oksigen terlarut di dalam air. Sensor *probe* memerlukan waktu 3 menit untuk mencapai nilai saturasi. Tabel 4.3 ini merupakan tampilan respon dari sensor *probe*. Presentase kadar oksigen terlarut adalah nilai yang digunakan untuk kalibrasi pada modul sensor, ketika pembacaan nilai presentase berkisar antara 90% - 100% dengan pembacaan yang stabil maka pembacaan data sudah akurat.

Tabel 4. 3 Pengujian terhadap saturasi udara

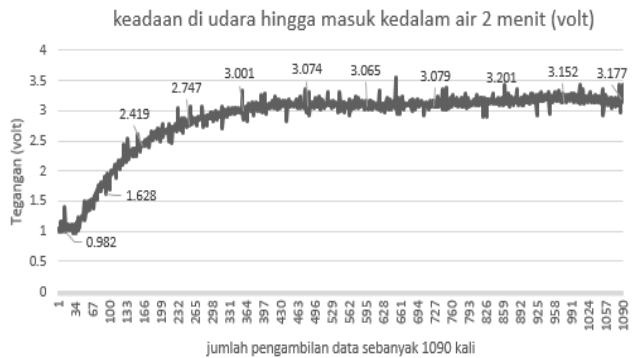
Waktu(detik)	Presentase (%)	Temperature (°C)
0	111	23
20	105	24
40	108	24

60	110	24
Waktu(detik)	Presentase (%)	Temperature (°C)
80	103	24
100	104	24
120	101	24
140	97	24
160	98	24
180	97	24

4.3.3 Pengujian Awal Terhadap Air

Pengujian ini dilakukan untuk mengetahui bagaimana respon sensor *probe* pada saat dimasukkan ke dalam air. Grafik ini menampilkan respon sensor *probe* selama 2 menit dengan periode sampling 110ms.

Pada gambar 4.4 dapat dilihat pembacaan yang stabil didapatkan selama 2 menit. Proses pembacaan cukup lama disebabkan proses difusi antara oksigen dan membran memerlukan waktu untuk ionisasi.



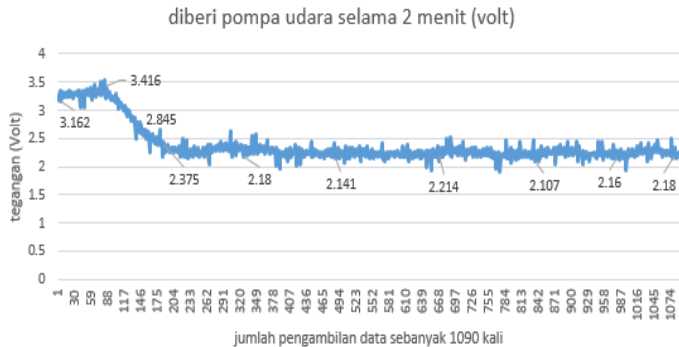
Gambar 4. 5 Output data di udara hingga masuk ke dalam air

Dari pengujian diatas diperlukan waktu 2 menit untuk mendapat data yang stabil yaitu 3.1 volt. Pada pengambilan

data ke 430, pembacaan do meter sudah mendekati nilai saturasi.

4.3.4 Pengujian Terhadap Pompa Udara

Pengujian ini dilakukan untuk mengetahui bagaimana respon sensor *probe* pada saat pompa udara dihidupkan. Grafik ini menampilkan respon sensor *probe* selama 2 menit dengan periode sampling 110ms.

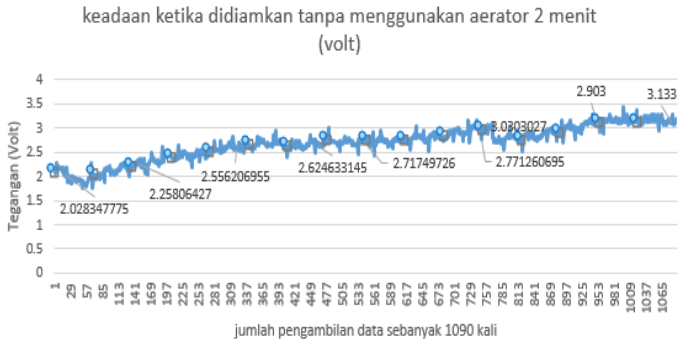


Gambar 4. 6 *Output* data di beri pompa udara

Dari pengujian diatas didapatkan hasil selama 2 menit di beri pompa udara oksigen terlarut bertambah signifikan pada menit awal dan mengalami saturasi pada tegangan 2.18 *volt*. Pada pembacaan data ke 204 nilai do meter sudah mendekati nilai saturasi. Dari data membuktikan penambahan oksigen terlarut menggunakan pompa udara bisa dilakukan dan memiliki respon yang cepat.

4.3.5 Pengujian Ketika Pompa Udara Dimatikan

Pengujian ini dilakukan untuk mengetahui bagaimana respon sensor *probe* terhadap keadaan air setelah diberi pompa udara selama 2 menit dengan periode sampling 110ms.



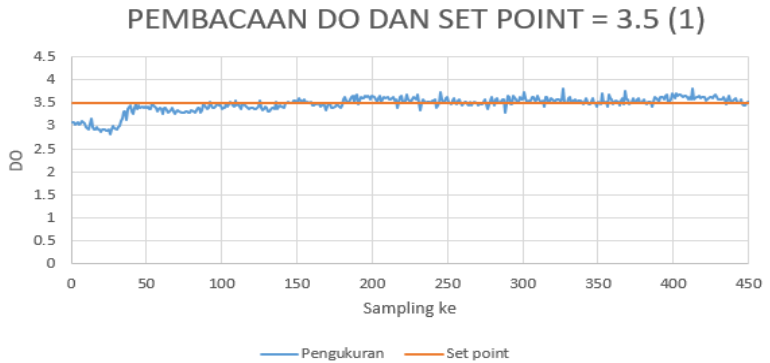
Gambar 4. 7 Output data ketika dидiamkan tanpa pompa udara

Dari pengujian di atas didapatkan hasil selama 2 menit bahwa oksigen terlarut memerlukan waktu yang cukup lama untuk hilang setelah diberi pompa udara. Dari pengambilan data awal sebesar tegangan 2 volt dan saturasi pada tegangan 3.1 volt.

4.4. Pengujian Set Point

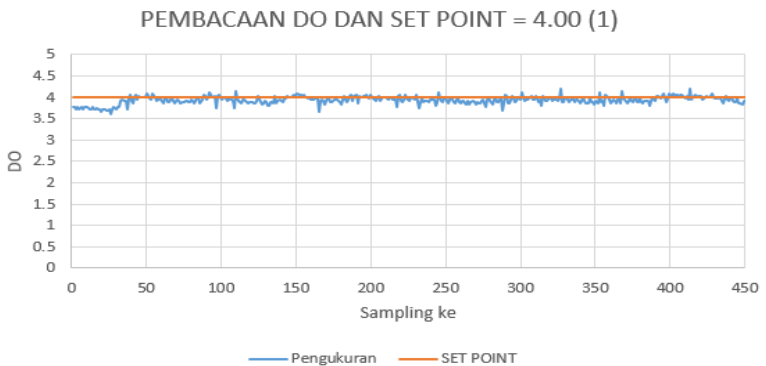
Pengujian ini dilakukan untuk mendapatkan hasil untuk kerja sistem kontrol oksigen terlarut dengan pompa udara. Pengujian ini dilakukan di dalam akuarium sesuai dengan batasan masalah. Nilai pembacaan oksigen terlarut dicatat selama 4.5 menit dengan nilai *set point* oksigen terlarut bernilai 3.5, 4.00, 4.5, 5.00 ppm.

Pada gambar 4.7 menunjukkan respon sistem dalam mencapai nilai *setting point*, dapat dilihat nilai oksigen terlarut mencapai *setting point* pada sampling ke 50 tetapi belum dalam keadaan *steady state* dikarenakan pembacaan pada alat. Keadaan *steady state* terjadi pada sampling ke 100 atau selama 1 menit. Periode sampling pada sistem ini adalah 600ms.



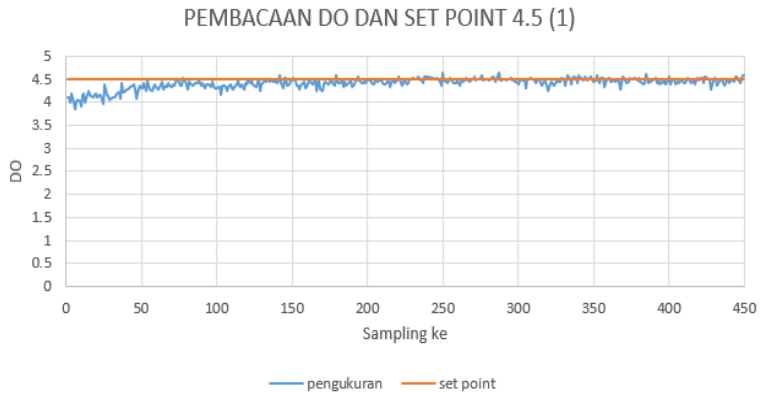
Gambar 4. 8 Oksigen Terlarut pada Set Point 3.5

Pada gambar 4.8 menunjukkan respon sistem dalam mencapai nilai *setting point*, dapat dilihat nilai oksigen terlarut mencapai *setting point* pada sampling ke 50 atau selama 30 detik. Pada percobaan ini sistem lebih cepat dalam mencapai keadaan *steady state*



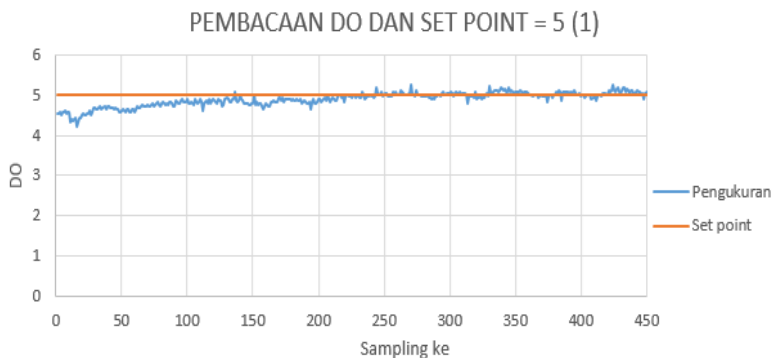
Gambar 4. 9 Oksigen Terlarut pada *Set Point* 4.00

Pada gambar 4.8 menunjukkan respon sistem dalam mencapai nilai *setting point*, dapat dilihat nilai oksigen terlarut mencapai *setting point* pada sampling ke 80 atau selama 48 detik tetapi keadaan *steady state* terjadi pada detik ke 72.



Gambar 4. 10 Oksigen Terlarut Pada *Set Point* 4.50

Pada gambar 4.8 menunjukkan respon sistem dalam mencapai nilai *setting point*, dapat dilihat nilai oksigen terlarut mencapai *setting point* pada sampling ke 150 atau selama 84 detik. Respon *steady state* terjadi pada sampling ke 160 atau selama 96 detik dikarenakan kadar oksigen terlarut mendekati nilai saturasi yaitu 5.8 ppm.



Gambar 4. 11 Oksigen Terlarut Pada *Set Point* 5.00

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah melakukan pengujian dari keseluruhan sistem pada tugas akhir ini dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan pengujian sensor dapat disimpulkan bahwa nilai rata – rata *error* dari pembacaan alat dengan periode sampling 600ms adalah 0.075%.
2. Pengujian karakteristik aktuator memberikan kesimpulan bahwa menambah pompa udara yang digunakan tidak menambah nilai saturasi oksigen sebesar 5.88 ppm.
3. Dari data yang diperoleh menunjukkan bahwa respon sistem membutuhkan waktu 2 menit untuk mengejar nilai *setting point* dengan nilai *error* positif dan membutuhkan waktu 1 jam untuk mengejar *setting point* dengan $\Delta error$ negatif dikarenakan proses reduksi oksigen terlarut membutuhkan waktu yang lama.
4. Dapat dibuat alat yang bisa mengontrol kadar oksigen terlarut di dalam air dengan mengimplementasikan metode *fuzzy logic* yang tertanam pada mikrokontroler
5. Pengaturan kadar oksigen terlarut dengan metode *fuzzy logic* sangat efisien untuk banyak *input* dan *output*.
6. *Fuzzy Logic Controller* merupakan suatu sistem kendali yang jauh lebih mudah penerapannya dibandingkan kendali konvensional, karena tidak perlu mencari model matematis dari sistem.
7. Fungsi keanggotaan dari *fuzzy* yang dibuat disesuaikan dengan kondisi *plan* yang cepat saat *error* besar dan melambat saat *error* mengecil.

5.2. Saran

Beberapa saran yang perlu disampaikan dalam laporan ini dalam rangka pengembangan penelitian ini antara lain adalah sebagai berikut:

1. Untuk penggunaan yang lebih efisien, penulis menyarankan mengganti baterai 9 volt dengan rangkaian power *supply* 12 volt untuk penggunaan yang lebih lama.

2. Untuk mendapatkan *output* sensor yang lebih presisi bisa digunakan rangkaian filter pada output rangkaian penguat operasional.
3. Untuk pengembangan selanjutnya penulis menyarankan membuat sistem dengan proses reduksi oksigen terlarut.

DAFTAR PUSTAKA

- [1] Wibisono, G.A., “Studi Kelayakan Investasi Pembuatan Perikanan Pembibitan Ikan Lele Dalam Perspektif *Supply Chain Management*”, Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, 2012
- [2] Yosanto, R.A., “Pembenihan dan Pembesaran Ikan Lele Sangkuriang *Clarias* sp. Di Unit Kerja Budidaya Air Tawar Wonocatur Cangkringan Sleman Yogyakarta”, Institut Pertanian Bogor, Bogor, 2015
- [3] Yusvarina, M., Sumarna., “Rancang Bangun Sistem Kontrol Kadar Oksigen di Dalam Air Pada Kolam Pembenihan Ikan Lele Mutiara di Unit Kerja Budidaya Air Tawar(UKBAT) Wonocatur Cangkringan, Sleman, Yogyakarta”, Jurnal Universitas Negeri Yogyakarta, Vol 5, No.7, 2016
- [4] Urbasa, P.A., “Dampak Kualitas Air Pada Budi Daya Ikan Dengan Jaring Tancap di Desa Toulimembet Danau Tondano”, Jurnal Budidaya Perairan, Vol. 3 No. 1: 59-67, jan 2015
- [5] Fujaya, Y., 2002. “Fisiologi ikan: dasar pengembangan teknologi perikanan”, Diterbitkan oleh Direktorat Jenderal Pendidikan Tinggi Depdiknas, Jakarta, 2002
- [6] Macqy, G., Pajarillo, J., Tenorio, J.E., Trambulo, E.M., Apsay, M.R.B., Chua, M.G., “Development of Dissolved Oxygen Monitoring System for Fish Ponds”, IEEE 3rd International Conference on System Engineering and Technology, Malaysia, 19 - 20 Aug 2013
- [7] Zulkarnain, M.R., “Sistem Monitoring Kualitas Air Sungai Yang Dilengkapi Dengan Data Logger Dan Komunikasi Wireless Sebagai Media Pengawasan Pencemaran Limbah Cair”, Prosiding Undergraduate Theses of Electrical Engineering, RSE 621.384 Zul s, Surabaya, Juni 2015
- [8] Resmana, Ferdinando., Thiang, Widagdo A.S., “Implementasi *Fuzzy Logic* Pada *Microcontroller* Untuk Kendali Putaran Motor DC”, Proceedings, Industrial Electronic Seminar, Surabaya, Oktober 1999
- [9] Masri'an., ” Pengendalian Orientasi *Webcam* Sebagai Pengawas Ruangan Dengan Metode Kontrol *Fuzzy*”, Prosiding Undergraduate Theses of Electrical Engineering, Semarang, Januari 2011

- [10] Hidayati, Q., Prasetyo, M.E., "Pengaturan Motor DC dengan Menggunakan Mikrokontroler Berbasis Fuzzy-PID" Jurnal Teknologi Terpadu, No.1 Vol. 4, Balikpapan, 2016
- [11] Kusumadewi, Sri dan Purnomo, Hari. 2010. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*, Yogyakarta, Indonesia.
- [12] Princeton University. "*Fuzzy Inference System*", <URL: <http://www.cs.princeton.edu/courses/archive/fall07/cos436/HI/HIDD/Knapp/fuzzy004.htm>>, Oktober, 2007
- [13] PJRC Electronic Project Components Available Worldwide, "Teensy 3.2 New Features", <URL: <https://www.pjrc.com/teensy31.html>>, 2014
- [14] Shirriff, K., Badger, P., "*Secrets of Arduino PWM*", <URL: <https://www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM>>, 2010
- [15] I2C BUS, "*I2C Bus Specification*", <URL: <http://i2c.info/i2c-bus-specification>>, 2010.
- [16] Pujiono," Rangkaian Elektronika Analog", Graha Ilmu, Yogyakarta, 2012
- [17] Erwanda, R., "Rancang Bangun Prototipe Pengendali Kecepatan Motor Induksi 3 Fasa Dengan Pengaturan Tegangan Berbasis Mikrokontroler Arduino Dan Android", Universitas Lampung, Lampung, 2016

LAMPIRAN

```
#include "fis_header.h"
#include "Wire.h"
#include "LiquidCrystal_I2C.h"

LiquidCrystal_I2C lcd(0x3f, 16, 3);
const int numReadings = 1 ;

int readings[numReadings];
int readIndex = 0;
float total = 0;
int setpoint = A1;
float SET=0;
int DOmeter = A0;
float nilaidometer = 0;
float errortadi=0;
float Derror;
float error;
float SETPOINT=0;

int PWM=255;
int z;
int y;

// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 12;

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);
  pinMode(3 , OUTPUT);
```

```

//LCD
lcd.begin();
lcd.setCursor (0,0);
lcd.print("DO:");
lcd.setCursor (8,0);
lcd.print("DC:");
lcd.setCursor (0,1);
lcd.print("SP:");
lcd.setCursor (-2,2);
lcd.print("De:");
lcd.setCursor (-2,3);
lcd.print("e:");

for (int thisReading = 0; thisReading < numReadings;
thisReading++) {
    readings[thisReading] = 0;
}
}

// Loop routine runs over and over again forever:
void loop()
{
    //PWM = 255-PWM
    analogWrite(3, PWM);

    total = total - readings[readIndex];
    readings[readIndex] = analogRead(DOMeter);
    total = total + readings[readIndex];
    readIndex = readIndex + 1;
    if (readIndex >= numReadings) {
        // ...wrap around to the beginning:
        readIndex = 0;
    }

    // put your main code here, to run repeatedly:
    nilaidometer = total / numReadings;
    SET=analogRead(setpoint);
    nilaidometer = -1.9991 * (nilaidometer * 0.0032258065)+ 7.4486 ;
    SET=(SET*0.0032258065*1.860)+0.2;

```

```

SETPOINT=SET-0.2;
if (SET<2.3) SET=2.3;
if (SET>5.85) SET=5.85;
error=SET - nilaidometer;
Derror=error-errortadi;

lcd.setCursor (3,0);
lcd.print(nilaidometer,2);
lcd.setCursor (3,1);
lcd.print(SETPOINT );
lcd.setCursor (1,2);
lcd.print(Derror);
lcd.setCursor (1,3);
lcd.print(error);
delay(500);

// Read Input: error
g_fisInput[0] = error;
// Read Input: derror
g_fisInput[1] = Derror;

g_fisOutput[0] = 0;

fis_evaluate();

// Set output vlaue: dutycycle/pwm

//SET pompa udara
z=map(g_fisOutput[0],20,79,0,100);
y=map(z,100,0,0,255);
PWM = y;
lcd.setCursor (11,0);
//if(z<10) lcd.print(" ");
lcd.print(z);
lcd.print(" ");
Serial.println(nilaidometer);
}

```

```

//*****
*****
// 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);
}

// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}

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, fis_trimf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 4, 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 3 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { -5, -5, -1, -0.5 };
FIS_TYPE fis_gMFI0Coeff2[] = { -1, 0, 0.25 };
FIS_TYPE fis_gMFI0Coeff3[] = { 0, 0.25, 0.5 };

```

```

FIS_TYPE fis_gMFI0Coeff4[] = { 0.25, 0.5, 5, 5 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1,
fis_gMFI0Coeff2, fis_gMFI0Coeff3, fis_gMFI0Coeff4 };
FIS_TYPE fis_gMFI1Coeff1[] = { -0.5, -0.5, -0.25, 0 };
FIS_TYPE fis_gMFI1Coeff2[] = { -0.25, 0, 0.25 };
FIS_TYPE fis_gMFI1Coeff3[] = { 0, 0.25, 0.5, 0.5 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1,
fis_gMFI1Coeff2, fis_gMFI1Coeff3 };
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff,
fis_gMFI1Coeff };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { 0, 0, 30, 50 };
FIS_TYPE fis_gMFO0Coeff2[] = { 30, 50, 70 };
FIS_TYPE fis_gMFO0Coeff3[] = { 50, 70, 100, 100 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1,
fis_gMFO0Coeff2, fis_gMFO0Coeff3 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff };

// Input membership function set
int fis_gMFI0[] = { 0, 1, 1, 0 };
int fis_gMFI1[] = { 0, 1, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };

// Output membership function set
int fis_gMFO0[] = { 0, 1, 0 };
int* fis_gMFO[] = { fis_gMFO0 };

// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 1, 1 };
int fis_gRI1[] = { 1, 2 };
int fis_gRI2[] = { 1, 3 };
int fis_gRI3[] = { 2, 1 };

```



```

int fis_gRI4[] = { 2, 2 };
int fis_gRI5[] = { 2, 3 };
int fis_gRI6[] = { 3, 1 };
int fis_gRI7[] = { 3, 2 };
int fis_gRI8[] = { 3, 3 };
int fis_gRI9[] = { 4, 1 };
int fis_gRI10[] = { 4, 2 };
int fis_gRI11[] = { 4, 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 };

```

```

// Rule Outputs

```

```

int fis_gRO0[] = { 1 };
int fis_gRO1[] = { 1 };
int fis_gRO2[] = { 1 };
int fis_gRO3[] = { 1 };
int fis_gRO4[] = { 1 };
int fis_gRO5[] = { 1 };
int fis_gRO6[] = { 2 };
int fis_gRO7[] = { 2 };
int fis_gRO8[] = { 2 };
int fis_gRO9[] = { 3 };
int fis_gRO10[] = { 3 };
int fis_gRO11[] = { 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 };

```

```

// Input range Min

```

```

FIS_TYPE fis_gIMin[] = { -4, -0.5 };

```

```

// Input range Max

```

```

FIS_TYPE fis_gIMax[] = { 4, 0.5 };

```

```

// Output range Min

```

```

FIS_TYPE fis_gOMin[] = { 0 };

```

```

// Output range Max

```

```

FIS_TYPE fis_gOMax[] = { 100 };

//*****
*****
// 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 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };
    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);
    }
}

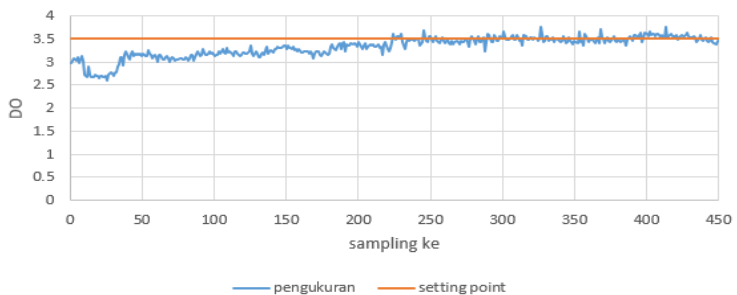
fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
sW += fuzzyFires[r];
}

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);
    }
}
errortadi=error;
}

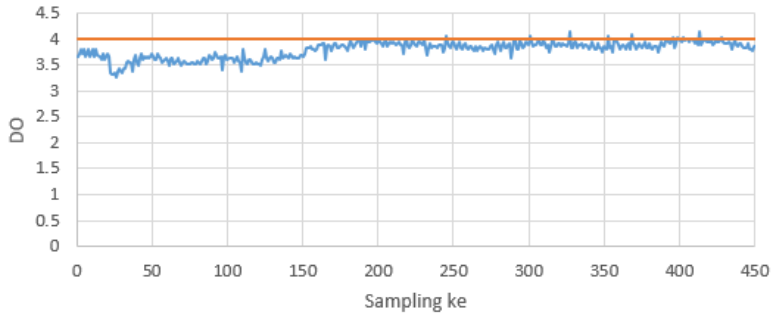
```

A. Pengambilan data ke 2

PEMBACAAN DI DAN SET POINT = 3.5 (2)

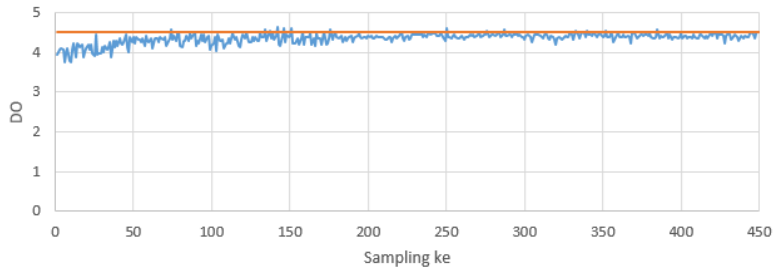


PEMBACAAN DO DAN SET POINT = 4 (2)



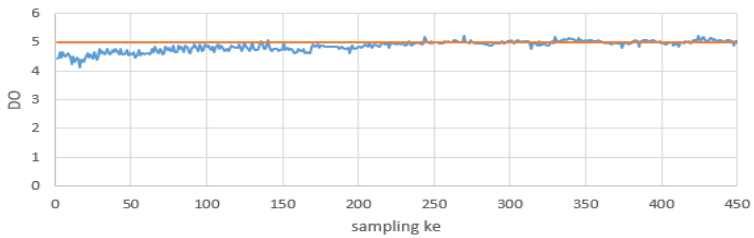
— pengukuran — setting point

PEMBACAAN DO DAN SET POINT = 4.5 (2)



— pengukuran — setting point

PEMBACAAN DO DAN SET POINT = 5 (2)



— pengukuran — setting point

BIODATA PENULIS



Luthfi Riadhi lahir di Dumai pada 14 Maret 1995, yang merupakan anak kedua dari dua bersaudara dari pasangan Karsidianto dan Fitriani. Penulis menyelesaikan pendidikan dasar di SD03 YKPP Dumai dan dilanjutkan dengan pendidikan menengah di SMP YKPP Dumai dan SMAN 1 Dumai. Pada tahun 2013, penulis memulai pendidikan di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Selama kuliah, penulis aktif membantu penyelenggaraan kegiatan dan aktif sebagai asisten laboratorium Elektronika Dasar.

Email:

luthfiriadhi95@gmail.com

Halaman ini sengaja dikosongkan