

**RANCANG BANGUN SISTEM STABILISASI
SUHU PENGGORENGAN DENGAN METODE
FUZZY BERBASIS IOT**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun oleh:

Karina Zaruska NIRM:1051810

Muhamad Azhari NIRM:1051812

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

TAHUN 2022

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM STABILISASI SUHU PENGGORENGAN
DENGAN METODE *FUZZY* BERBASIS IOT**

Oleh:

Karina Zaruska /1051810

Muhamad Azhari/1051812

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan

Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1

Pembimbing 2



Eko Sulistyono, M.T

Penguji 1



Aan Febriansyah, M.T

Penguji 2



Yudhi, M.T



Indra Dwisaputra, M.T

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1: Karina Zaruska	NIM:1051810
Nama Mahasiswa 2: Muhamad Azhari	NIM:1051812

Dengan Judul: RANCANG BANGUN SISTEM STABILISASI SUHU
PENGGORENGAN DENGAN METODE *FUZZY* BERBASIS
IOT

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata di kemudian hari melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 19 Januari 2022

Nama Mahasiswa

1. Karina Zaruska

2. Muhamad Azhari

Tanda Tangan

.....

.....

ABSTRAK

Di Bangka Belitung banyak UMKM penghasil oleh-oleh khas Bangka Belitung seperti kemplang, kretek, kerupuk dan sebagainya. Namun teknik penggorengan yang digunakan oleh UMKM dari hasil kunjungan kebanyakan masih menggunakan cara manual, seperti UMKM tidak dapat memonitoring suhu penggorengan pada saat produksi dan juga tidak dapat mengontrol. UMKM dalam pengontrolan suhu masih menggunakan cara manual, yakni jika suhu minyak sudah mulai panas dan produk sudah mulai kecoklatan maka knop kompor dkecilkan begitu juga sebaliknya jika suhu kurang panas maka knop kompor akan dibesarkan. Tetapi UMKM tidak dapat mendeteksi berapa suhu yang dipakai saat itu sehingga banyak menimbulkan cacat produksi. Dari hasil survei yang sudah dilakukan di UMKM itu maka diperlukan teknologi untuk menstabilisasi suhu penggorengan sehingga hasil produksi menghasilkan produk yang sesuai. Untuk pelaksanaan dari proses stabilisasi penggorengan ini menggunakan teknologi mikrocontroller, dimana aktuatornya menggunakan motor servo yang dikontrol ke kompor. Sedangkan proses kontrol menggunakan fuzzy logic, dimana input dan outputnya akan disetting sesuai rule base (aturan yang didefinisikan sesuai kondisi penggorengan pada UMKM). Rata-rata persentase error dari data hasil pengujian sensor suhu penggorengan dengan input rendah adalah 0.31%, input sedang adalah 0.29%, dan input tinggi adalah 0.43%.

Kata Kunci: IoT, Logika Fuzzy, Monitoring, Stabilisasi.

ABSTRACT

In Bangka Belitung, there's many UMKM produce souvenirs typical of Bangka Belitung such as kemplang, kretek, crackers, etc. However, the frying technique used by UMKM from the results of visits mostly still uses the manual method, such as UMKM cannot monitor frying temperatures during production and also cannot control them. UMKM in controlling the temperature still use the manual method, namely if the temperature of the oil has started to heat up and the product started to brown, the stove knob is reduced and otherwise if the temperature isn't hot enough, the stove knob will be raised. However, UMKM can't detect the temperature used at that time, causing many production defects. From the results that has been carried out in UMKM, technology is needed to stabilize the temperature of the frying pan so that the production results produce the appropriate product. For the implementation of this frying stabilization process using mikrokontroler technology, where the actuator use servo motor that's controlled to the stove. While the control process uses fuzzy logic, where the input and output will be set according to the rule base (rules defined according to the frying conditions in UMKM). The average percentage error from the test results of the frying temperature sensor with low input is 0.31%, medium input is 0.29%, and high input is 0.43%.

Keywords: Fuzzy Logic, IoT, Monitoring, Stabilisation.

KATA PENGANTAR

Bismillahirrahmanirrahim

Puji dan syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan makalah proyek akhir ini dengan baik dan sesuai dengan waktu yang telah ditentukan.

Makalah ini dibuat untuk memenuhi salah satu syarat wajib kelulusan Sarjana Terapan di Politeknik Manufaktur Negeri Bangka Belitung. Dengan adanya makalah ini diharapkan para pembaca dapat mengetahui gambaran proyek akhir yang dibuat oleh penulis. Makalah proyek akhir ini dibuat dengan melakukan pengembangan dari jurnal-jurnal penelitian yang sudah pernah dilakukan sebelumnya. Penulis mencoba menerapkan ilmu pengetahuan yang didapat selama 4 tahun menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dalam pembuatan proyek akhir ini. Selain itu, penulis juga mendapatkan informasi berupa data-data pendukung dari makalah-makalah proyek akhir mahasiswa Politeknik Manufaktur Negeri Bangka Belitung tahun-tahun sebelumnya.

Selama menyusun makalah proyek akhir ini penulis mendapatkan banyak bantuan dan bimbingan dari berbagai pihak sehingga proses penulisan makalah ini dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Orangtua dan keluarga tercinta yang selalu memberikan dukungan baik secara moral maupun materil sehingga penulis dapat memberikan hasil yang terbaik.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Eko Sulistyono, M.T. selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung sekaligus dosen pembimbing 1 dalam proyek akhir ini.
4. Bapak Aan Febriansyah, M.T. selaku dosen pembimbing 2 dalam

proyek akhir ini.

5. Bapak Muhammad Iqbal Nugraha, M.Eng. selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Bapak Indra Dwisaputra, M.T. selaku Ketua Program Studi D IV Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
7. Seluruh tenaga pendidik dan kependidikan Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah membantu dalam penyelesaian proyek akhir ini.
9. Seluruh pihak yang ikut terlibat baik secara langsung maupun tidak langsung dalam penyelesaian proyek akhir ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan laporan ini masih terdapat banyak kekurangan karena terbatasnya pengetahuan dan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak demi perbaikan dan pengembangan penulisan makalah ini di masa yang akan datang. Penulis berharap makalah ini dapat bermanfaat bagi pihak yang berkepentingan pada khususnya dan bagi perkembangan ilmu pengetahuan dan teknologi pada umumnya.

Sungailiat, 14 Februari 2022

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN BUKAN PLAGIAT	ii
<i>ABSTRAK</i>	iii
<i>ABSTRACT</i>	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Proyek Akhir	2
BAB II DASAR TEORI	3
2.1 Pengertian Sensor Suhu Resistive Temperature Detector (RTD)	3
2.2 <i>Converter</i> RTD ke <i>Digital</i> (MAX31865)	4
2.3 <i>Fuzzy Logic Control</i>	5
2.3.1 Pengertian	5
2.3.2 Logika <i>Fuzzy</i>	6
2.3.3 Metode <i>Mamdani</i>	6
2.3.4 Prosedur Metode <i>Fuzzy Mamdani</i>	6
2.4 ArduinoUno R3	8

2.5	Motor Servo MG996R.....	9
2.6	LCD (<i>Liquid Crystal Display</i>).....	10
2.7	I ² C/TWI LCD 2004	11
2.8	<i>Internet of Thing</i> (IoT)	12
2.9	NodeMCU ESP8266	12
2.10	ARDUINO IDE.....	13
2.11	MATLAB	14
BAB III METODE PELAKSANAAN		16
3.1	Pengumpulan dan Pengolahan Data	19
3.2	Perancangan <i>Hardware</i> dan Sistem Stabilisasi serta Kontrol Menggunakan <i>Fuzzy logic control</i> Berbasis IoT.....	19
3.2.1	Perancangan <i>Hardware</i>	19
3.2.2	Perancangan Sistem Stabilisasi dan Kontrol Suhu Penggorengan Dengan Metode <i>Fuzzy Logic Control</i> Berbasis IoT.	20
3.2.2.1	Perancangan Logika <i>Fuzzy</i>	20
3.2.2.2	Fungsi Keanggotaan <i>Input</i>	21
3.2.2.3	Fungsi Keanggotaan <i>Output</i>	22
3.2.2.4	Perancangan Aturan (<i>Rule Base</i>)	22
3.2.2.5	Perancangan Diagram Sistem	23
3.3	Perancangan <i>Software</i> Sistem Stabilisasi dan Kontrol Berbasis IoT	24
3.4	Pembuatan <i>Hardware</i> dan Sistem Stabiisasi serta Sistem Kontrol Suhu Penggorengan Berbasis IoT.....	24
3.4.1	Pembuatan <i>Hardware</i>	24
3.4.2	Pembuatan Sistem Stabilisasi dan Kontrol Suhu Penggorengan dengan Metode <i>Fuzzy Logic Control</i>	24
3.5	Pengujian.....	25

3.5.1	Pengujian <i>Hardware</i>	25
3.5.2	Pengujian <i>Software</i>	25
3.5.3	Pengujian Keseluruhan Sistem	25
3.6	Analisa Data	26
3.7	Pembuatan Makalah	26
BAB IV PEMBAHASAN.....		27
4.1	Deskripsi Alat.....	27
4.2	Perancangan <i>Hardware</i> dan <i>Software</i> Alat <i>Monitoring</i> dan Kontrol Suhu Penggorengan berbasis IoT	28
4.3	Pengujian Sensor RTD PT-100	30
4.3.1	Perancangan dan pembuatan Sensor RTD PT-100.....	30
4.3.2	Menampilkan Data di LCD 20x4 dan Modul I ² C.....	33
4.3.3	Komunikasi <i>Serial</i> ArduinoUnoR3 dengan NodeMCU	34
4.3.4	Pengujian dari NodeMCU ke <i>Smartphone</i>	34
4.3.5	Pengujian Logika <i>Fuzzy</i>	35
4.4	Pengujian Rancang Bangun Sistem Stabilisasi Suhu Penggorengan dengan Metode <i>Fuzzy</i> Berbasis IoT	35
4.4.1.	Pengujian Suhu Penggorengan dengan <i>Input</i> Rendah.....	36
4.4.2.	Pengujian Suhu Penggorengan dengan <i>Input</i> Sedang	37
4.4.3.	Pengujian Suhu Penggorengan dengan <i>Input</i> Tinggi.....	40
4.4.4.	Pengujian <i>Monitoring</i> dan Kontrol Menggunakan <i>Blynk</i>	42
BAB V PENUTUP.....		43
5.1	Kesimpulan.....	43
5.2	Saran.....	44
DAFTAR PUSTAKA		45

DAFTAR GAMBAR

Gambar 2. 1 Sensor Suhu PT100.....	3
Gambar 2. 2 Sambungan 3 Kabel PT100.....	4
Gambar 2. 3 Blok Diagram <i>Fuzzy logic control</i> [12]	6
Gambar 2. 4 ArduinoUno R3 [17]	8
Gambar 2. 5 <i>Servo</i> Digital MG996R[19].....	9
Gambar 2. 6 Sinyal PWM Motor <i>Servo</i> MG996R[19].....	10
Gambar 2. 7 Konfigurasi Pin LCD[20].....	10
Gambar 2. 8 Modul I ² C LCD 2004[17]	11
Gambar 2. 9 <i>Internet of Things</i> (IoT) [23]	12
Gambar 2. 10 NodeMCU [25]	13
Gambar 2. 11 Tampilan Arduino IDE	14
Gambar 2. 12 Tampilan <i>Software</i> MATLAB	15
Gambar 3. 1 Diagram Blok Proses Pembuatan Proyek Akhir	18
Gambar 3. 2 Rancangan Konstruksi Elektrik.....	20
Gambar 3. 3 Fungsi Keanggotaan (<i>Membership Function</i>) <i>Input Error</i>	21
Gambar 3. 4 Fungsi Keanggotaan(<i>Membership Function</i>) <i>input Delta-Error</i>	21
Gambar 3. 5 Fungsi Keanggotaan (<i>Member Function</i>) <i>Output</i>	22
Gambar 3. 6 Diagram Perancangan Sistem	23
Gambar 4. 1 Blok Diagram Prinsip Kerja Rancang Bangun Sistem Stabilisasi dan Kontrol Suhu Penggorengan	27
Gambar 4. 2 Desain blok diagram <i>Fuzzy Logic Control</i>	29
Gambar 4. 3 Rangkaian <i>Hardware</i> Dari Pengujian Suhu Menggunakan Sensor RTD PT-100.....	30
Gambar 4. 4 Komunikasi <i>Serial</i> ArduinoUnoR3 dengan NodeMCU.	34
Gambar 4. 5 Pemasangan Alat pada <i>Box</i>	35
Gambar 4. 6 Konstruksi Alat	36
Gambar 4. 7 Sistem <i>Monitoring</i> dan sistem kontrol menggunakan aplikasi <i>Blynk</i> pada <i>Smartphone</i>	42

DAFTAR TABEL

Tabel 2. 1 Contoh Suhu untuk PT100 dengan $400\Omega R_{REF}$ [8].....	4
Tabel 2. 2 Spesifikasi ArduinoUno R3	8
Tabel 2. 3 Fitur TowerPro MG-996R	9
Tabel 2. 4 <i>interface</i> Pin LCD[19]	10
Tabel 2. 5 Spesifikasi NodeMCU ESP8266	13
Tabel 3. 1 Rule Base	22
Tabel 4. 1 Hasil Pengujian Sensor RTD PT-100	32
Tabel 4. 2 Pengujian suhu penggorengan dengan <i>input</i> rendah.....	36
Tabel 4. 3 Pengujian suhu penggorengan dengan <i>input</i> sedang	38
Tabel 4. 4 Pengujian suhu penggorengan dengan <i>input</i> tinggi	40

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup.....	49
Lampiran 2 Program Arduino	52
Lampiran 3 Program NodeMCU.....	69



BAB I

PENDAHULUAN

1.1 Latar Belakang

Bangka Belitung merupakan salah satu penghasil produk pangan tradisional berbahan baku ikan seperti kerupuk kemplang, pempek dan otak-otak[1]. UMKM (Usaha Mikro, Kecil dan Menengah) di Bangka Belitung saat ini masih menggunakan proses pengontrolan suhu penggorengan secara manual yaitu apabila suhu penggorengan sudah dikatakan panas dan produk yang digoreng sudah berubah warna menjadi kecoklatan dan mekar maka knop kompor akan dikecilkan, begitu juga sebaliknya jika dirasa kurang panas maka knop kompor akan dibesarkan[2]. Pada proses penggorengan, salah parameter untuk melihat kualitas dari kerupuk kemplang yaitu membuat kerupuk kemplang menjadi mengembang dan berpori rendah atau pengembangan [3], Karena Tinggi dan rendahnya tingkat kesukaan konsumen sangat mempengaruhi keberlangsungan produktivitas produk yang akan dibuat[4]. Berdasarkan kunjungan yang telah dilakukan, suhu penggorengan yang stabil dianjurkan kisaran 180°C – 200°C. Adapun Dampak Dari menggoreng dengan suhu tidak stabil ialah apabila suhu tidak mencapai 180°C maka kerupuk kemplang akan mentah dan memiliki umur simpan yang cepat, dan apabila suhu telah melebihi 200°C maka kerupuk kemplang akan mengalami hangus. Sistem *monitoring* dan kontrol juga dapat membantu mempermudah UMKM karena dapat melihat suhu penggorengan saat ini menggunakan *smartphone*. IoT (*internet of things*) memungkinkan objek dikontrol dari jarak yang jauh melalui infrastruktur jaringan yang ada[5]. Sehingga pengguna tidak mengalami kesulitan atau kendala dalam proses penggorengan. Diharapkan dengan proses penggorengan yang sudah menggunakan teknologi dalam penelitian ini maka hasil penggorengannya lebih bagus juga produk tidak mengalami cacat produk dan hasil produksi dapat meningkat. Dari permasalahan ini menjadi proyek akhir dengan judul “Rancang

Bangun Sistem Stabilisasi Suhu Penggorengan dengan Metode *Fuzzy* Berbasis IoT”.

1.2 Perumusan Masalah

1. Bagaimana membuat sistem stabilisasi suhu penggorengan sesuai yang diinginkan melalui *input setpoint* ?
2. Berapakah suhu penggorengan kerupuk kemplang yang baik?
3. Bagaimana cara untuk mengontrol suhu berbasis IoT?

1.3 Batasan Masalah

1. Besarnya derajat motor tidak berbanding lurus dengan besarnya api.
2. *Delay* untuk menentukan *set point* sedikit lama.

1.4 Tujuan Proyek Akhir

Berdasarkan rumusan masalah diatas, adapun tujuan dari dilakukannya proyek akhir ini ialah :

1. Menstabilkan suhu penggorengan dengan sistem *Fuzzy logic control*.
2. Mengetahui suhu yang baik untuk menggoreng kerupuk kemplang dengan *smartphone*.
3. Mengontrol suhu yang diinginkan menggunakan *smartphone*.

BAB II DASAR TEORI

2.1 Pengertian Sensor Suhu Resistive Temperature Detector (RTD)

Suhu ialah suatu besaran yang dapat menyatakan derajat panas dingin suatu benda. Alat ukur yang biasanya digunakan untuk mengukur suhu adalah *thermometer*. Indera peraba merupakan alat yang cenderung digunakan untuk mengukur suhu dalam kehidupan sehari-hari. Namun dengan adanya suatu perkembangan teknologi, maka diciptakanlah *thermometer* untuk mengukur suhu dengan *valid*[6].

Sensor PT100 termasuk dalam kategori PRT(*platinum Resistant Thermometers*). Sensor- sensor ini tidak lain adalah resistor *variable* yang mengubah resistansinya berdasarkan suhu disekitarnya. Jangkauan suhu menggunakan PT100 yaitu 0°C sampai 400°C.

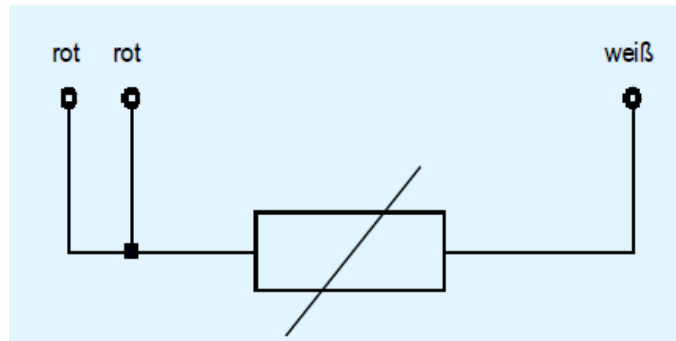


Gambar 2. 1 Sensor Suhu PT100

PT100 yang dipakai menggunakan 3 kabel koneksi, RTD PT100 terdapat 3 macam *wiring* yang keakuratannya berbanding lurus dengan banyaknya kabel yang di pakai, yaitu:

- 2 Wire (2 wiring kabel)
- 3 Wire (3 Wiring Kabel)
- 4 Wire (4 Wiring Kabel)

Kelebihan Sensor Suhu RTD PT-100 diantaranya adalah tingkat akurasi pengukuran yang tinggi, sinyal yang perubahannya linier, respon perubahan yang cepat.



Gambar 2. 2 Sambungan 3 Kabel PT100.

2.2 Converter RTD ke Digital (MAX31865)

Modul MAX31865 merupakan suatu perangkat dari RTD PT100 yang berfungsi untuk mendapat nilai suhu yang akurasi. Menggunakan *amplifier* yang dirancang untuk membaca resistansi rendah, dan *amplifier* yang dapat menyesuaikan dan mengkompensasi pada *output*. Memiliki tingkat presisi 15-bit ADC dengan resolusi suhu 0.03125°C [7]. PT100 menggunakan 400 resistor referensi, dan PT1000 menggunakan 4k resistor referensi.

Tabel 2. 1 Contoh Suhu untuk PT100 dengan $400\Omega R_{REF}$ [8]

No.	Temperature (C°)	RTD Resistance (Ω)	ADC CODE	ADC CODE/ 32-256(C°)
1.	20	107.79	8830	19.94
2.	30	111.67	9148	29.98
3.	40	115.54	9465	39.78
4.	50	119.40	9781	49.66
5.	60	123.24	10096	59.50
6.	70	127.08	10410	69.31
7.	80	130.90	10723	79.09
8.	90	134.71	11035	88.84
9.	100	138.51	11346	98.56

No.	Temperature (C°)	RTD Resistance (Ω)	ADC CODE	ADC CODE/ 32-256(C°)
10.	110	142.29	11657	108.28
11.	120	146.07	11966	117.94
12.	130	149.83	12274	127.56
13.	140	153.58	12583	137.19
14.	150	157.33	12888	146.75
15.	160	161.05	131.94	156.31
16.	170	164.77	13498	165.81
17.	180	168.48	13802	175.31
18.	190	172.17	14104	184.75
19.	200	175.86	14406	194.19
20.	225	185.01	15156	217.63
21.	250	194.10	15901	240.91

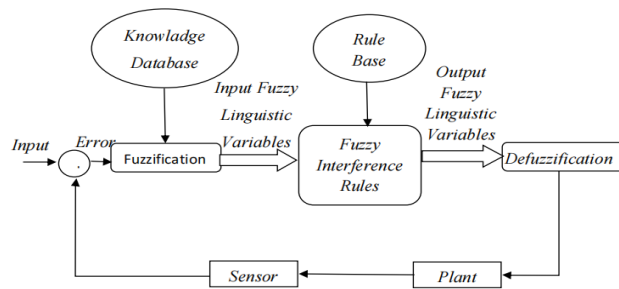
2.3 *Fuzzy Logic Control*

Fuzzy logic control (FLC) merupakan salah satu metode pengendalian sistem yang saat ini sering digunakan, khususnya di bidang sistem kendali atau kontrol[9]. *Fuzzy* secara bahasa dapat diartikan sebagai kabur atau samar yang memiliki arti suatu nilai dapat bernilai benar atau salah secara bersamaan[10].

2.3.1 **Pengertian**

Fuzzy pertama kali diperkenalkan oleh Jan Lukasiewicz di tahun 1920-an sebagai teori kemungkinan. Logika ini memberikan jangkauan luas untuk menyatakan suatu nilai kebenaran untuk semua bilangan riil pada interval antara 0 dan 1.

Dalam rancangan *Fuzzy logic control* tidak memerlukan model matematis dari sistem yang akan dikendalikan. Terdapat tiga tahapan dalam *Fuzzy logic control*, yaitu fuzzifikasi, mekanisme inferensi dan defuzzifikasi[11].



Gambar 2. 3 Blok Diagram *Fuzzy logic control*[12].

2.3.2 Logika Fuzzy

Logika *Fuzzy* pertama kali diperkenalkan oleh Lotfi A. Zadeh dalam makalahnya tentang teori himpunan *Fuzzy* atau dasar *Fuzzy set* yang meliputi *inclusion*, perpotongan, penyatuan, penjumlahan, relasi, dan *convexity* pada tahun 1965.

Logika *Fuzzy* adalah cabang dari sistem kecerdasan buatan (*Artificial Inteligent*) yang meniru kemampuan seseorang untuk berpikir dalam bentuk sebuah algoritma yang kemudian dijalankan oleh sebuah mesin. *Membership Function* (MF) menunjukkan derajat keanggotaan untuk setiap nilai pada variabel *input/output* [13].

2.3.3 Metode Mamdani

Metode *Fuzzy Mamdani* merupakan salah satu bagian dari *Fuzzy Inference System* (mekanisme inferensi) yang berfungsi untuk mengambil suatu kesimpulan atau keputusan dari permasalahan yang tidak pasti. Metode *Fuzzy Mamdani* digunakan untuk mempermudah pengerjaan karena penarikan kesimpulan pada metode ini sesuai dengan naluri manusia Sehingga dengan menggunakan Metode *Fuzzy Mamdani* dapat menghasilkan keputusan yang baik dalam menyelesaikan permasalahan.

2.3.4 Prosedur Metode Fuzzy Mamdani

Proses sampai pada suatu kesimpulan atau keputusan dengan menggunakan metode *Fuzzy Mamdani* dilakukan melalui beberapa tahapan, yaitu pembentukan himpunan *Fuzzy*, aplikasi fungsi implikasi (*Fuzzy Inference Rules*), defuzzifikasi [14].

2.3.4.1 Pembentukan Himpunan Fuzzy(Fuzzifikasi)

Tahap pertama yaitu membuat himpunan Fuzzy(Fuzzifikasi). Fuzzifikasi merupakan suatu proses yang dilakukan dengan cara mentransformasikan *input* himpunan tegas(*crisp*) ke himpunan Fuzzy. Seperti yang kita tahu bahwa nilai *input* yang digunakan harus berupa bilangan tegas(riil).

Pada metode Fuzzy Mamdani ini tipe fungsi keanggotaan(*Membership Function*) yang digunakan ialah tipe trapesium, tipe segitiga, dan tipe bahu kiri atau kanan. Penentuan tipe yang di pakai yaitu dari banyaknya keanggotaan, apabila menggunakan 2 titik dari himpunan Fuzzy maka menggunakan tipe trapesium, apabila menggunakan 1 titik dari himpunan Fuzzy maka menggunakan tipe segitiga, dan tipe bahu kiri dan kanan digunakan apabila diawali dan diakhiri variabel suatu daerah Fuzzy.

Rumus Membership Function(MF):

$$\text{Triangular MF: } \text{trimf}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \dots\dots\dots(2.1)$$

$$\text{Trapezoidal MF: } \text{trapmf}(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{d-x}{d-c}\right), 0\right) \dots(2.2)$$

$$\text{Gausisian MF: } \text{gaussmf}(x; a, b, c) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \dots\dots\dots(2.3)$$

$$\text{Generalized Bell MF: } \text{gbelmf}(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \dots\dots\dots(2.4)$$

2.3.4.2 Aplikasi Fungsi Implikasi(Fuzzy Inference Rules)

Tahap ini yaitu penerapan fungsi implikasi yang merupakan struktur logika yang terdiri atas kumpulan premis dan konklusi. Bentuk dari fungsi implikasi adalah dengan pernyataan *IF x is A THEN y is B*, dengan x dan y adalah *scalar*, serta A dan B adalah himpunan Fuzzy[15].

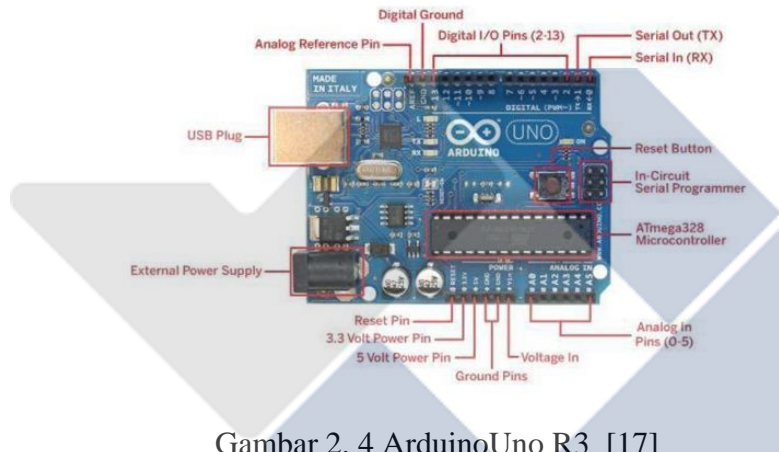
2.3.4.3 Defuzzifikasi

Tahap ini merupakan langkah terakhir dalam suatu sistem logika fuzzy yang dimana tujuannya untuk mengonversi dari hasil dari inference engine yang diekspresikan dalam bentuk fuzzy set ke suatu bilangan riil. Menurut Ross (2010), proses dalam menentukan titik pusat daerah Fuzzy dilakukan dengan menggunakan perumusan:

$$COG = \frac{\int_a^b \mu_A(x) \times dx}{\int_a^b \mu_A(x) dx} \dots\dots\dots(2.5)$$

2.4 ArduinoUno R3

ArduinoUno R3 merupakan salah satu mikrokontroler yang menggunakan chip Atmega328, dilengkapi dengan 14 pin *input/output* digital. 6 pin digunakan sebagai *output* PWM, 6 pin digunakan untuk *input* analog, osilator 16 MHz, satu port USB, *powerjack*, header ICSP, dan juga tombol reset. Arduino Uno R3 dioperasikan menggunakan port USB *computer*, USB *charger*, atau adaptor AC-DC dengan tegangan yang direkomendasikan 9 Volt[16].



Gambar 2. 4 ArduinoUno R3 [17]

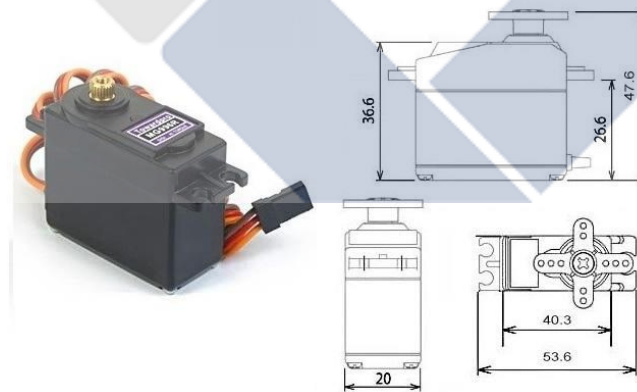
Tabel 2. 2 Spesifikasi ArduinoUno R3

No.	Pin	Keterangan
1.	<i>Microcontroller</i>	ATMega 328
2.	<i>OperatingVoltage</i>	5V
3.	<i>Digital I/O Pins</i>	14(of which 6 provide PWM output)
4.	<i>Analog InputPins</i>	6
5.	<i>InputVoltage</i>	6-20V
6.	<i>InputVoltage</i>	7-12V
7.	<i>DC Current per I/O Pin</i>	40mA
8.	<i>DC Currentfor 3.3v Pin</i>	50mA
9.	<i>FlashMemory</i>	32 KB (ATmega328)

No.	Pin	Keterangan
10.	SRAM	2KB (ATmega328)
11.	EEPROM	1KB (ATmega328)
12.	<i>ClockSpeed</i>	16MHz
13.	Panjang	68.6mm
14.	Lebar	53.4 mm
15.	Berat	25g

2.5 Motor Servo MG996R

Motor servo merupakan sebuah perangkat atau *actuator* (berputar) yang dirancang menggunakan *system close feedback* dimana posisi dari *motor servo* akan di informasikan kembali ke rangkaian kontrol yang ada dalam *motor servo*, sehingga *motor servo* dapat dikonfigurasi untuk menentukan dan memastikan posisi sudut poros *output* motor berdasarkan lebar pulsa yang dikirim melalui pin sinyal dari kabel motor[18].



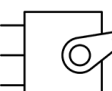
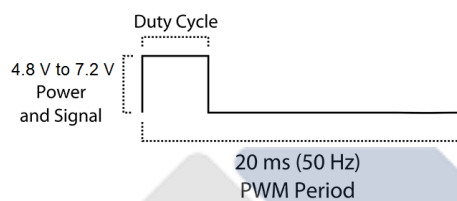
Gambar 2. 5 *Servo Digital MG996R*[19]

Tabel 2. 3 Fitur TowerPro MG-996R[19]

No.	Parameter	Keterangan
1.	Tegangan Operasi	4.8-7.2V
2.	Torsi	12.8kg/cm (4.8V)
3.	Torsi Maksimal	11kg/cm(6V)

No.	Parameter	Keterangan
4.	Tipe gear	Metal
5.	Rotasi	0°-180°
6.	Berat motor	55gm
7.	Kecepatan	0.1s/60°

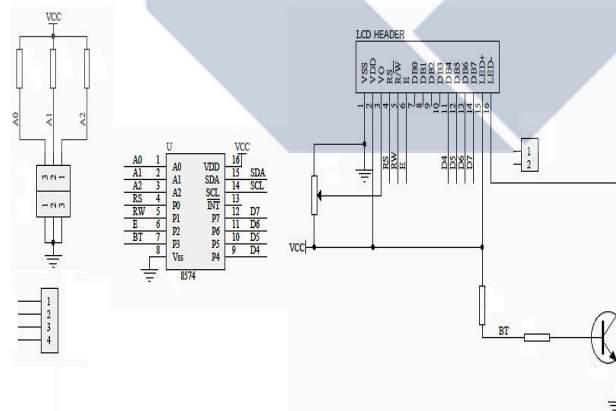
PWM=Orange (⏏)
Vcc = Red (+)
Ground=Brown (-)

Gambar 2. 6 Sinyal PWM Motor *Servo* MG996R[19]

2.6 LCD (*Liquid Crystal Display*)

Liquid Crystal Display(LCD) merupakan salah satu media yang digunakan untuk menampilkan hasil dari keluaran pada sebuah rangkaian elektronika[17].



Gambar 2. 7 Konfigurasi Pin LCD[20]

Tabel 2. 4 *interface* Pin LCD[21]

No. Pin	Simbol	Level	Keterangan
1	VSS	0V	<i>Ground</i>
2	VDD	5V	<i>Supply Voltage For Logic</i>

Pin No.	Symbol	Level	Description
3	VO	(VARIABEL)	Operating Voltage For LCD
4	R/W	H/L	H:Data,L:Instruction Code
5	RS	H/L	H:Read,L:Write
6	E	H,H->L	Chip Enable Signal
7	DB0	H/L	Data Bit 0
8	DB1	H/L	Data Bit 1
9	DB2	H/L	Data Bit 2
10	DB3	H/L	Data Bit 3
11	DB4	H/L	Data Bit 4
12	DB5	H/L	Data Bit 5
13	DB6	H/L	Data Bit 6
14	DB7	H/L	Data Bit 7
15	A	4.2-4.6	Led+
16	K	0v	Led -

2.7 I²C/TWI LCD 2004

I²C (*Inter Integrated Circuit*) LCD 2004 merupakan suatu modul yang dipakai untuk mempermudah penggunaan dari LCD 20x4 karena dapat mengurangi jumlah penggunaan kaki di LCD 20x4. Modul ini memiliki 4 Pin yang akan dihubungkan ke Arduino.

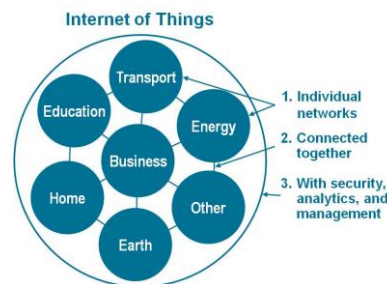
- GND : terhubung ke GND Arduino
- VCC : terhubung ke 5V Arduino
- SDA : Merupakan I²C data yang terhubung ke pin analog arduino
- SCL :Merupakan I²C *clock* dan terhubung ke pin analog pada arduino[17].



Gambar 2. 8 Modul I²C LCD 2004[17]

2.8 *Internet of Thing (IoT)*

Internet of Thing (IoT) merupakan suatu istilah yang memiliki artian penggunaan internet yang lebih besar, mengambil komputasi yang bersifat *mobile* dan konektivitas lalu menggabungkannya ke dalam kehidupan sehari-hari[22].

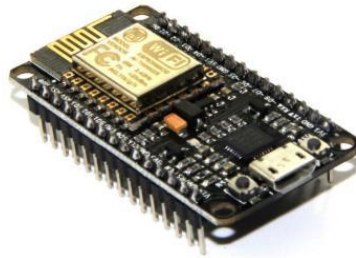


Gambar 2. 9 *Internet of Things (IoT)* [23]

Cara Kerja IoT (*Internet of Things*) yaitu melakukan interaksi sesama mesin yang terhubung secara otomatis tanpa campur tangan *user* dan dalam jarak tak tentu. Agar tercapainya cara kerja IoT (*Internet of Things*) diatas maka internet sebagai penghubung antara kedua interaksi mesin tersebut, sementara *user* atau pengguna hanya bertugas sebagai mengatur dan memantau pergerakan alat tersebut secara langsung. Manfaat yang dapat dirasakan dari konsep IoT (*Internet of Things*) ialah meringankan pekerjaan yang dilakukan dan pekerjaan dapat menjadi lebih cepat, mudah dan efisien[24].

2.9 **NodeMCU ESP8266**

NodeMCU ialah sebuah *platform IoT (internet of things)* bersifat *open source firmware* yang mengontrol aplikasi melalui jaringan *wi-fi* atau berbagai aplikasi lain. NodeMCU mirip dengan board Arduino pada ESP8266. NodeMCU menggabungkan ESP8266 menjadi sebuah *board* yang sederhana dengan berbagai fitur seperti mikrokontroler serta kemampuan akses Wi-fi serta *chip komunikasi USB to Serial*, sehingga hanya membutuhkan ekstensi kabel data mikro USB untuk memprogramnya. Secara umum, terdapat 3 produsen NodeMCU yang produknya saat ini beredar di pasaran yaitu: Amica, DOIT, dan Lolin/WeMos. Berikut ini ialah tampilan NodeMCU v1.0 yang dapat dilihat pada Gambar 2.10 dibawah ini.



Gambar 2. 10 NodeMCU [25]

Tabel 2. 5 Spesifikasi NodeMCU ESP8266

NO.	SPESIFIKASI	KETERANGAN
1.	<i>MICROCONTROLLER</i>	ESP8266
2.	TEGANGAN <i>INPUT</i> YANG DISARANKAN	3.3V - 12V
3.	GPIO	13 PIN
4.	KANAL PWM	10 KANAL
5.	10 BIT ADC PIN	1 PIN
6.	MEMORI <i>FLASH</i>	4 MB
7.	<i>WIFI</i>	IEEE 802.11 B/G/N
8.	FREKUENSI	2.4 GHZ – 22.5 GHZ
9.	<i>CLOCK SPEED</i>	40/26/24 MHZ
10.	<i>USB TO SERIAL CONVERTER</i>	CH340G
11.	PANJANG	57 MM
12.	LEBAR	30 MM
13.	<i>USB PORT</i>	MICRO USB

2.10 ARDUINO IDE

IDE (*Integrated Development Environment*) *Software* yang memungkinkan digunakan Arduino untuk mengeksekusi perintah terprogram dalam perangkat lunak yang digunakan untuk memprogram[26]. Bahasa pemrograman(*sketch*) Arduino adalah bahasa pemrograman C,dan dimodifikasi dari Bahasa aslinya untuk memudahkan pemula dalam memprogram. Program ini ditulis dalam *editor text* dan disimpan dalam format file.ino. Arduino IDE terdiri dari :

1. Uploader, sebuah modul yang memuat kode biner dari komputer ke *memory* di dalam papan Arduino.
2. Editor program, sebuah *windows* yang memungkinkan pengguna membuat dan mengedit program dalam bahasa *Processing*.
3. Compiler, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner.

Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Bahasa yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini[27].

Tampilan Arduino IDE dapat dilihat pada Gambar 2.11 berikut.



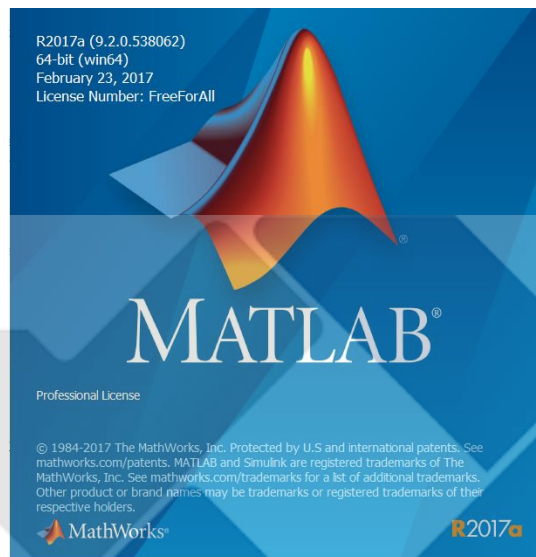
Gambar 2. 11 Tampilan Arduino IDE

IC mikrokontroler Arduino berisi program yang disebut Bootlader yang bertindak sebagai perantara *compiler* Arduino dengan mikrokontroler. Di bagian bawah program IDE terdapat kotak hitam dengan kemampuan untuk menampilkan status seperti *compile* dan *upload* program, kemudian dibagian pojok kanan atas terdapat *serial* monitor yang memiliki fungsi untuk menampilkan hasil eksekusi dari program yang dijalankan oleh Arduino IDE.

2.11 MATLAB

Matlab Menurut Widiarsono (2005:4) Nama Matlab merupakan akronim dari kata *Matrix Laboratory*. Versi pertama Matlab ditulis pada tahun 1970. Saat itu, Matlab digunakan mengajarkan teori matrik, aljabar linier dan analisis numerik.

Fungsi Matlab ini digunakan untuk memecahkan masalah bagian khusus, yang disebut *toolboxes* yang dapat digunakan untuk berbagai bidang pemroses sinyal, kontrol sistem, logika *fuzzy*, jaringan digital, optimasi, pemrosesan gambar, dan simulasi yang lain. Pada matlab terdapat tiga jendela perintah yang digunakan dalam operasinya yaitu *command windows* (layar perintah) dan *figure windows* (layar gambar), serta Note Pad (sebagai program *editor*)[28]. Pada Gambar 2.12 dibawah ini merupakan *software* MATLAB.

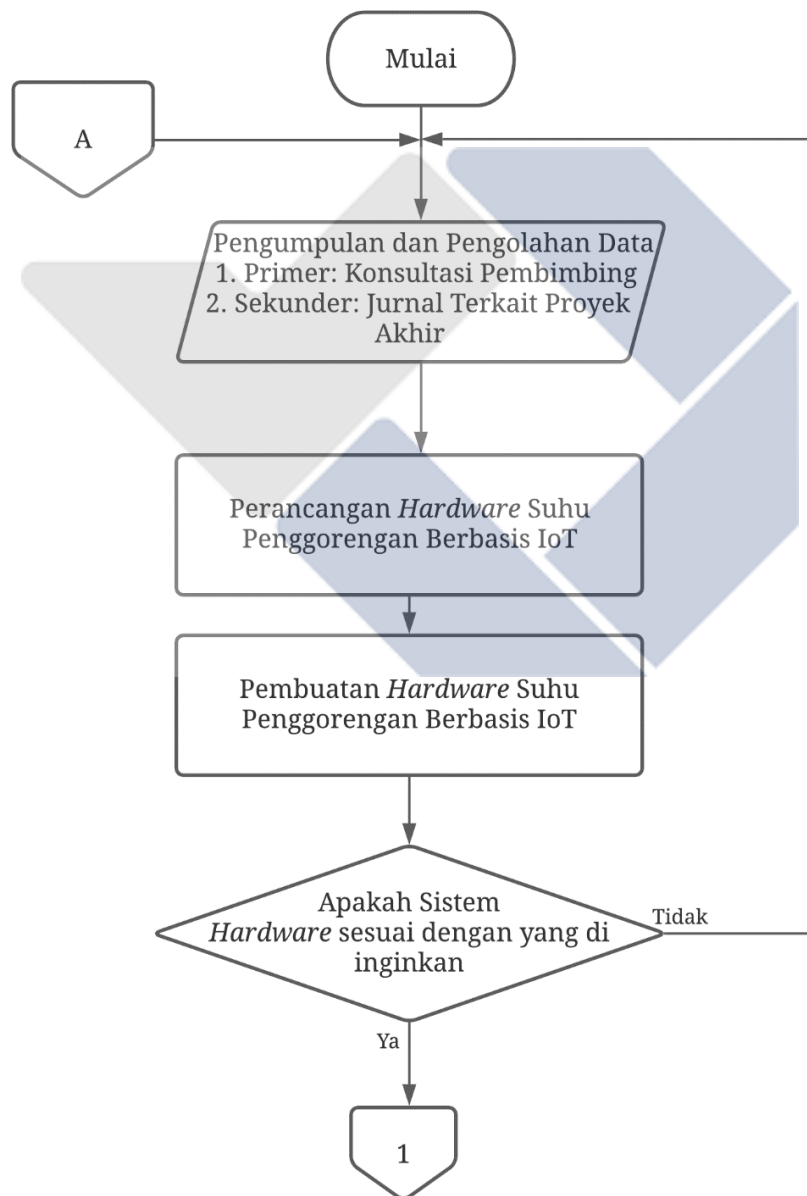


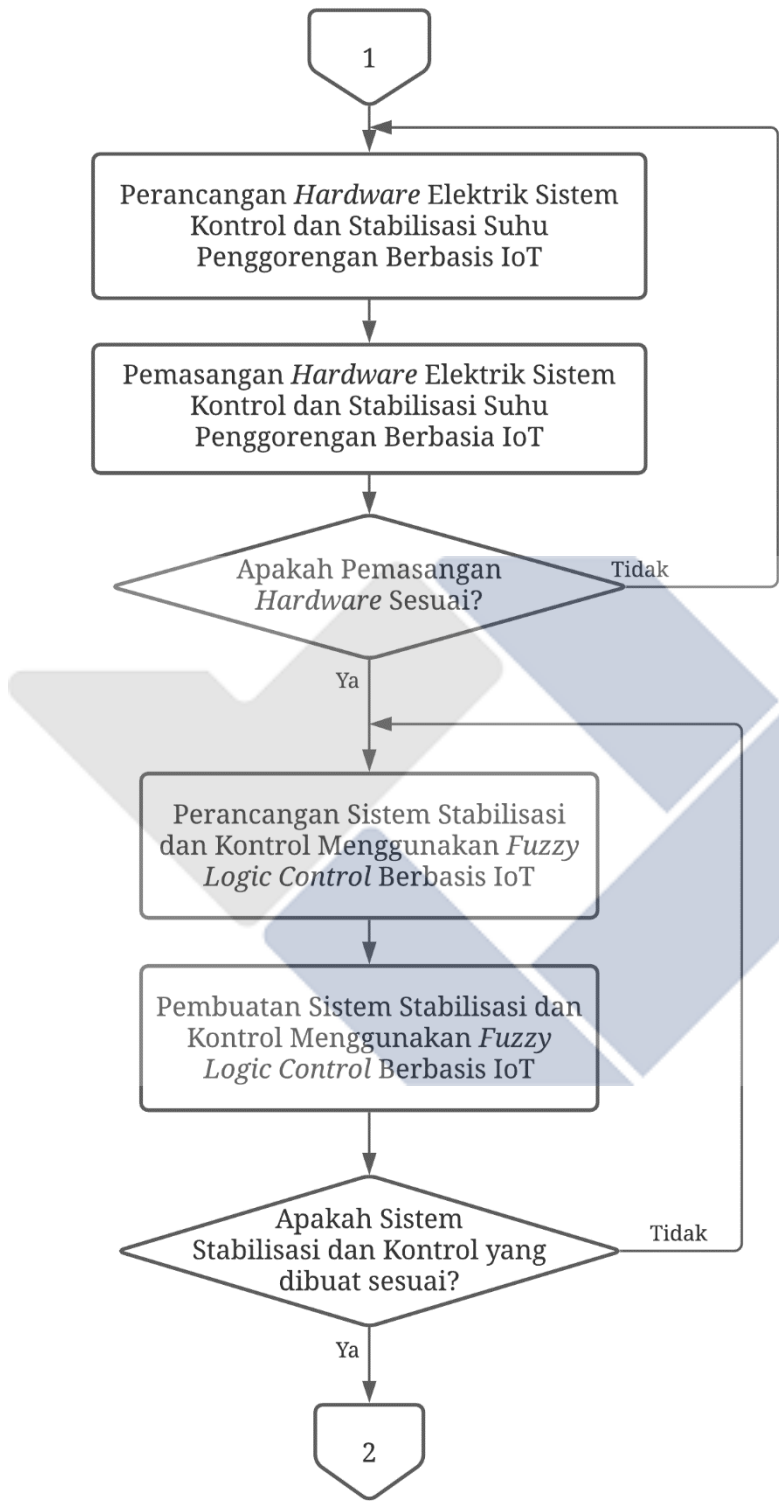
Gambar 2. 12 Tampilan *Software* MATLAB

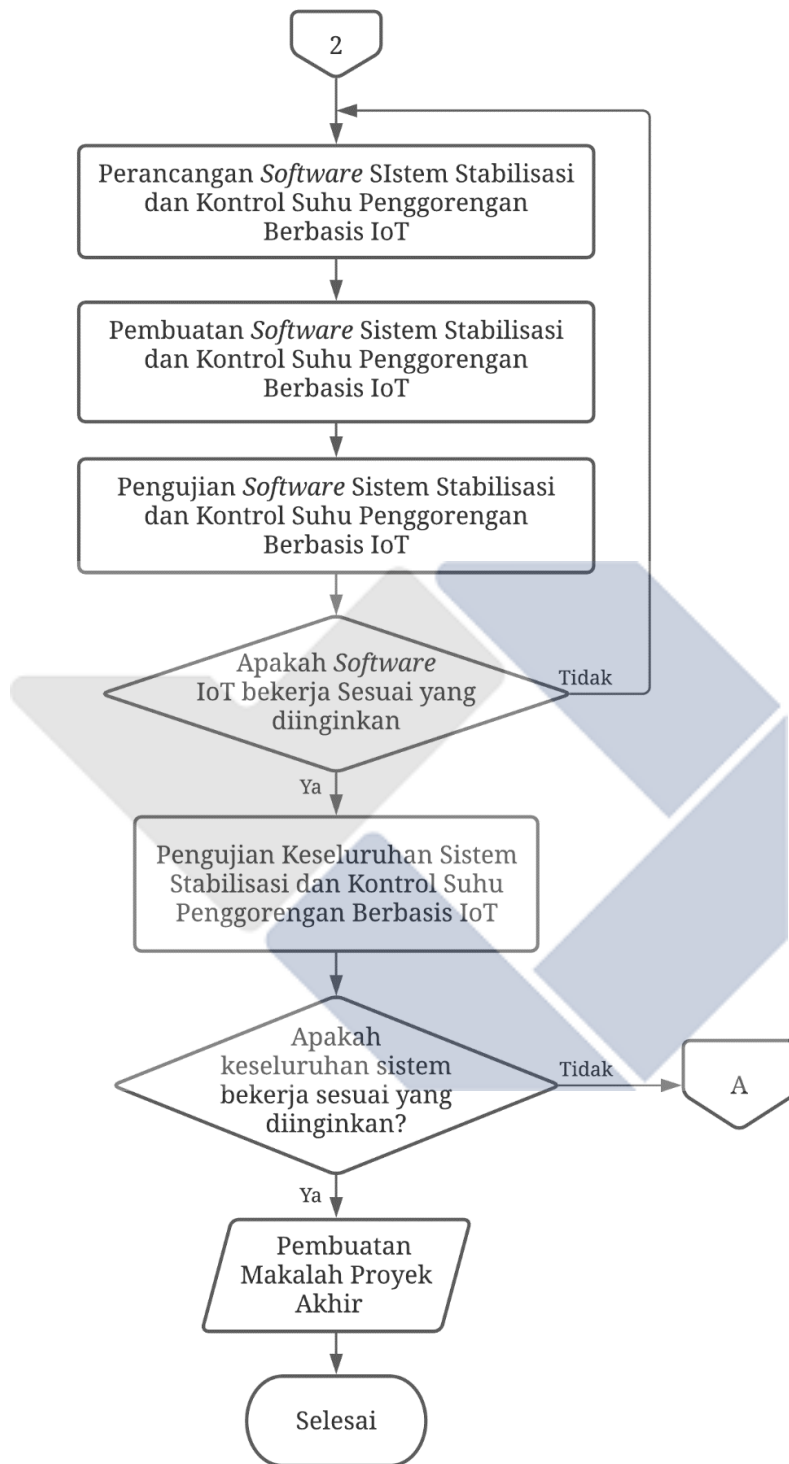
BAB III

METODE PELAKSANAAN

Dalam proses pembuatan proyek akhir yang berjudul “Rancang Bangun Sistem Stabilisasi Suhu Penggorengan dengan Metode *Fuzzy* Berbasis IoT” terdiri dari beberapa tahapan sebagai berikut:







Gambar 3. 1 Diagram Blok Proses Pembuatan Proyek Akhir

Penjelasan lebih lanjut dari tahapan di atas dapat di lihat pada langkah-langkah berikut ini :

3.1 Pengumpulan dan Pengolahan Data

Pengumpulan dan pengolahan data dilakukan dengan beberapa metode yang bertujuan untuk mendapatkan informasi berupa data-data yang dapat dijadikan sumber untuk mendukung pembuatan proyek akhir serta penulisan makalah proyek akhir ini.

Pada tahap ini menggunakan 2 metode untuk mengumpulkan data yaitu pengumpulan data primer dan pengumpulan data sekunder. Pengumpulan data primer yang dimaksud ialah melakukan konsultasi dengan dosen pembimbing berkaitan dengan pelaksanaan proyek akhir. Sedangkan pengumpulan data sekunder yang dimaksud ialah pengumpulan data dari jurnal-jurnal yang berhubungan dengan judul proyek akhir.

Selanjutnya dijadikan dalam sebuah ide baru dalam pelaksanaan proyek akhir ini.

3.2 Perancangan *Hardware* dan Sistem Stabilisasi serta Kontrol Menggunakan *Fuzzy logic control* Berbasis IoT

Pada tahap ini perancangan *hardware* berfungsi untuk menentukan dan memperkirakan bentuk konstruksi juga peletakan sensor. Selain itu, tahap ini juga bertujuan untuk merancang sistem kontrol untuk mengendalikan suhu penggorengan. Sedangkan perancangan *software* dilakukan untuk merancang sistem *monitoring* dan kontrol IoT, dan *interface* aplikasi *Blynk* pada *smartphone*.

3.2.1 Perancangan *Hardware*

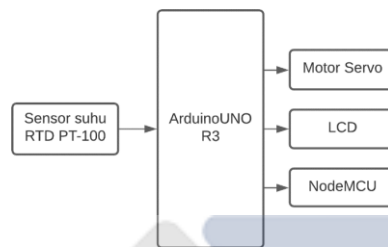
Rancangan *hardware* pada rancang bangun sistem stabilisasi dan kontrol ini terdiri dari 2 bagian yaitu perancangan *hardware* mekanik dan perancangan *hardware* elektrik.

Untuk perancangan *hardware* mekanik yang akan dibuat yaitu sebagai berikut.

- Pemasangan sensor suhu RTD PT-100 pada pinggir wajan penggoreng.

- Pemasangan Motor *Servo* yang di *couple* dengan knop kompor menggunakan *pulley timing* dan *timing belt*.
- Pemasangan mikrokontroler (ArduinoUno R3), MAX31865, NodeMCU ESP8266, LCD pada *panel box*.

Sedangkan untuk bagian perancangan *hardware* elektrik dapat dilihat pada Gambar 3.2 dibawah.



Gambar 3. 2 Rancangan Konstruksi Elektrik

3.2.2 Perancangan Sistem Stabilisasi dan Kontrol Suhu Penggorengan Dengan Metode *Fuzzy Logic Control* Berbasis IoT.

Tujuan dari mendesain sistem agar mempermudah dalam proses pembuatan sistem ini.

3.2.2.1 Perancangan Logika *Fuzzy*

Sistem kontrol dan sistem stabilisasi *fuzzy logic* memiliki 2 *input* dan 1 *output* yaitu nilai *Input error* dan *Delta-Error* lalu nilai *Output* 1 yaitu dalam bentuk pergerakan derajat motor yang digunakan untuk mengatur besarnya putaran knop kompor. Di bawah ini akan dijelaskan rumus mencari *error* dan *delta-error*.

- Mencari nilai *error*

$$error = set\ point - nilai\ aktual \dots\dots\dots(3.1)$$

Dimana,

set poin = besarnya suhu yang diinginkan

nilai aktual = besarnya suhu yang sebenarnya/suhu terbaca melalui sensor

- Mencari nilai *delta-error*

$$delta\ error = t_1 - t_0 \dots\dots\dots(3.2)$$

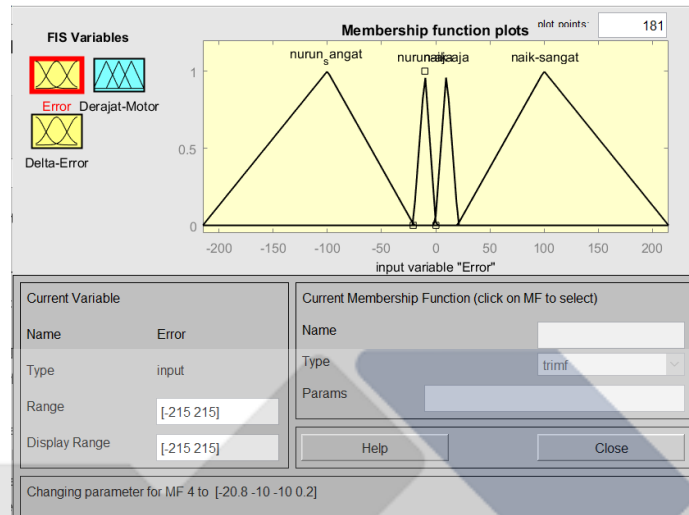
Dimana,

$t_0 =$ suhu awal dalam pengambilan data sebelum 3 detik

$t_1 =$ suhu akhir dalam pengambilan data setelah 3 detik

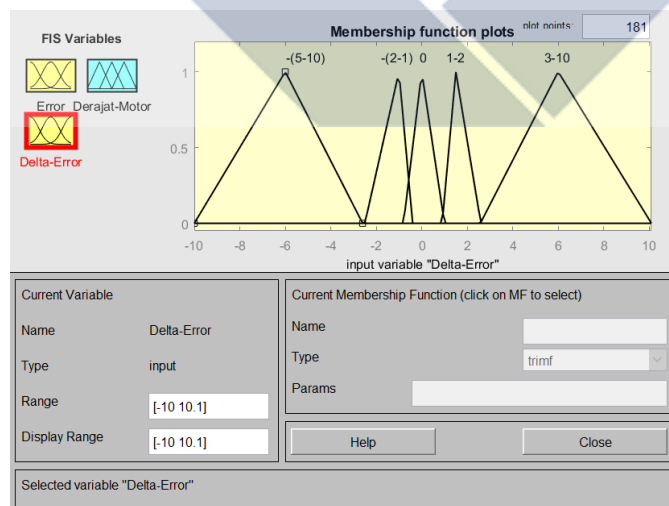
3.2.2.2 Fungsi Keanggotaan *Input*

Fungsi keanggotaan untuk *error* dapat dilihat pada Gambar 3.3 dibawah ini.



Gambar 3. 3 Fungsi Keanggotaan (*Membership Function*) *Input Error*

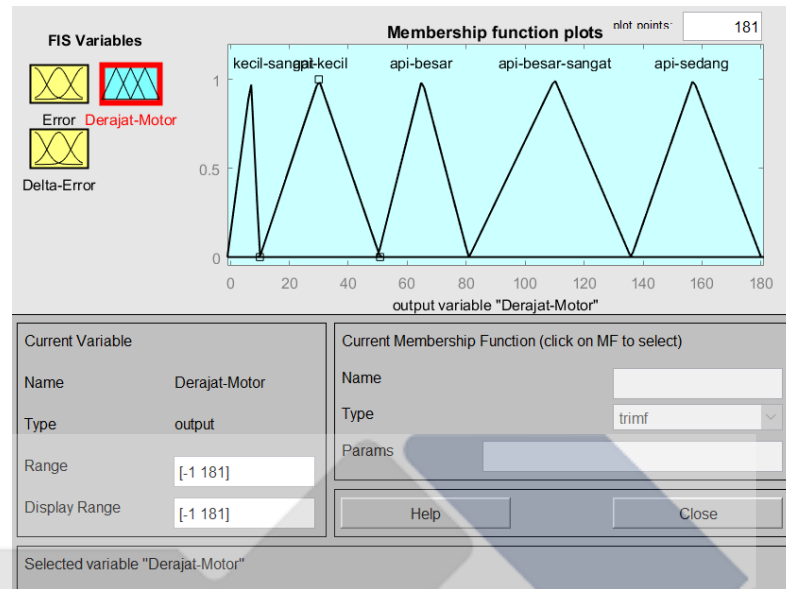
Fungsi keanggotaan untuk *Delta-Error* dapat dilihat pada Gambar 3.4 dibawah ini.



Gambar 3. 4 Fungsi Keanggotaan(*Membership Function*) *input Delta-Error*

3.2.2.3 Fungsi Keanggotaan Output

Gambar 3.5 menunjukkan fungsi keanggotaan *output* pergerakan derajat motor.



Gambar 3. 5 Fungsi Keanggotaan (*Member Function*) Output

3.2.2.4 Perancangan Aturan (*Rule Base*)

Pada tabel 3.1 menunjukkan perancangan *rule base fuzzy* yang dibuat:

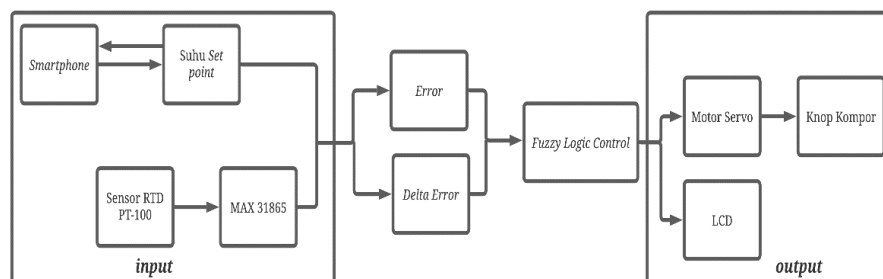
Tabel 3. 1 Rule Base

<i>D-Error</i> <i>Error</i>	0	1-2	3-10	-1- -2	-3 - -10
Naik	Api Sangat Kecil	Api Sangat Kecil	Api Sangat Kecil	Api Sangat Kecil	Api Sangat Sedang
Sangat Naik	Api Besar	Api Besar	Api Kecil	Api Besar	Api Sangat Besar
Turun	Api Kecil	Api Kecil	Api Kecil	Api Kecil	Api Kecil
Sangat Turun	Api Kecil	Api Kecil	Api Kecil	Api Kecil	Api Kecil

Dimana *error* naik merupakan pembacaan sensor suhu RTD saat kondisi *input error* 0°C - 20°C , *error* 0°C - $(-20)^{\circ}\text{C}$ merupakan pembacaan sensor suhu RTD saat kondisi *input error* turun, *error* 21°C - 215°C merupakan pembacaan *input error* sangat naik, dan *error* -21°C - $(-215)^{\circ}\text{C}$ merupakan pembacaan sensor suhu RTD saat kondisi *input error* sangat turun. Sedangkan *input Delta error* mempunyai rentang nilai dari -10 hingga 10. Untuk *Delta error* 0 merupakan nilai setelah pengambilan data selama 3 detik tidak memiliki perubahan nilai. Untuk *Delta error* 1-2 merupakan nilai setelah pengambilan data selama 3 detik memiliki selisih nilai 1°C atau 2°C . Untuk *Delta error* -1-(-2) merupakan nilai setelah pengambilan data selama 3 detik memiliki selisih nilai -1°C atau -2°C . Untuk *Delta error* 3-10 merupakan nilai selisih perbandingan suhu awal dan suhu akhir selama 3 detik yang memiliki nilai 3°C sampai 10°C . Untuk *Delta error* -3-(-10) merupakan nilai selisih perbandingan suhu awal dan suhu akhir selama 3 detik yang memiliki nilai -3°C sampai -10°C . Untuk *output* derajat motor mempunyai rentang nilai 0° hingga 180° . Untuk *output* api sangat kecil memiliki nilai 0° hingga 10° . Untuk *output* api kecil memiliki nilai 11° hingga 50° . Untuk *output* api besar memiliki rentang nilai dari 51° hingga 80° . Untuk *output* api besar sangat memiliki rentang nilai dari 81° hingga 135° . Dan untuk *output* api sedang memiliki rentang nilai 136° hingga 180° .

3.2.2.5 Perancangan Diagram Sistem

Pada Gambar 3.6 dibawah menunjukkan Diagram sistem keseluruhan pada sistem stabilisasi dan kontrol dengan metode *fuzzy* berbasis IoT



Gambar 3. 6 Diagram Perancangan Sistem

3.3 Perancangan *Software* Sistem Stabilisasi dan Kontrol Berbasis IoT

Perancangan *software* merupakan tahapan yang dilakukan secara bersamaan untuk menentukan dan merancang perangkat yang akan digunakan pada proyek akhir ini. Perangkat *hardware* yang digunakan pada proyek akhir ini meliputi perangkat *input*, prosesor, dan *output*. Perangkat *input* terdiri dari sensor suhu RTD dan *push button*. Perangkat pemroses terdiri dari ArduinoUnoR3 dan NodeMCU. Sedangkan perangkat *output* terdiri dari *servo* digital, LCD dan *smartphone*. Perancangan *software* pada proyek akhir ini merupakan pengkodean pada Arduino Uno R3 dan NodeMCU yang digunakan untuk mengontrol sistem secara keseluruhan dengan menerapkan logika *fuzzy* untuk menentukan tingkat kenaikan atau penurunan dari suhu penggorengan menggunakan *software* Arduino IDE dan dapat memonitoring data melalui *smartphone* dengan aplikasi *Blynk*.

3.4 Pembuatan *Hardware* dan Sistem Stabiisasi serta Sistem Kontrol Suhu Penggorengan Berbasis IoT

Tahap ini bertujuan untuk merealisasikan perancangan alat sebagai mana yang telah dibuat sebelumnya.

3.4.1 Pembuatan *Hardware*

Pembuatan *hardware* sistem stabiisasi serta sistem kontrol suhu penggorengan berbasis IoT ialah membuat kontruksi pada sistem stabiisasi serta sistem kontrol suhu penggorengan berbasis IoT, meliputi pembuatan pada bagian-bagian sistem mekanik. Pembuatan *hardware* yang akan dibuat meliputi *box* peletakan komponen dan pemasangan *servo* digital dengan knop kompor.

3.4.2 Pembuatan Sistem Stabilisasi dan Kontrol Suhu Penggorengan dengan Metode *Fuzzy Logic Control*

Pembuatan *software* pada proyek ini menggunakan pengkodean pada aplikasi Arduino IDE yang digunakan untuk mengontrol sistem secara keseluruhan dari “Sistem Stabiisasi serta Sistem Kontrol Suhu Penggorengan Berbasis IoT” yang akan dibuat. Beberapa langkah dalam pengkodean Arduino IDE yang dilakukan yaitu, pengkodean untuk masing-masing komponen, pembuatan tampilan *monitoring* pada *smartphone* dan pengkodean logika *fuzzy*.

3.5 Pengujian

Setelah proses pembuatan *hardware* dan *software* selesai, langkah selanjutnya yaitu proses uji coba sistem. Pengujian ini bertujuan untuk mengetahui kerja alat apakah bekerja secara optimal dan berfungsi sesuai dengan fungsi yang diinginkan.

3.5.1 Pengujian *Hardware*

- Pengujian MAX31865 terhadap Sensor RTD PT-100 untuk mengetahui kemampuan dalam mendeteksi suhu dari Sensor RTD PT-100.
- Pengujian MAX31865 untuk mengkonvert nilai analog dari Sensor RTD PT-100 ke nilai digital.
- Pengujian ArduinoUno terhadap MAX31865 untuk mengetahui kemampuan dalam mendeteksi suhu dari MAX31865.
- Pengujian ArduinoUno dengan Sensor RTD PT-100, MAX31865, motor *servo*, LCD ketika muncul suhu untuk mengetahui sistem bekerja dengan baik.

3.5.2 Pengujian *Software*

- Pengujian *interface* aplikasi yang berfungsi untuk memonitoring data sensor dengan menggunakan *smartphone*.
- pengujian mengirimkan *input* untuk sistem kontrol melalui *smartphone*

3.5.3 Pengujian Keseluruhan Sistem

Pada tahapan ini dilakukan untuk mengetahui sistem kontrol dan sistem *monitoring* suhu penggorengan dengan metode *fuzzy* berkerja dengan baik. Pengujian ini dilakukan dengan cara memberikan *input* suhu rendah melalui *smartphone* apakah sistem kontrol dan sistem *monitoring* akan bekerja. Kemudian diberi suatu kondisi berbeda yaitu memberikan *input* suhu sedang melalui *smartphone* apakah sistem kontrol dan sistem *monitoring* akan bekerja sesuai dengan yang diinginkan. Dan yang terakhir ketika diberi kondisi yaitu dengan memberikan *input* suhu tinggi melalui *smartphone* apakah sistem kontrol dan sistem *monitoring* akan bekerja sesuai dengan yang diinginkan.

3.6 Analisa Data

Tahap ini merupakan tahap untuk melakukan analisis data dari data hasil pengujian sistem pada *hardware* maupun *software* yang sudah dilakukan. Jika menemukan kurangnya optimalisasi pada penggunaan maka pada tahap ini juga perbaikan dilakukan guna mendapatkan hasil yang optimal.

3.7 Pembuatan Makalah

Pada tahap ini yaitu tahap yang terakhir pada proses pembuatan proyek akhir yaitu membuat makalah dari hasil yang mempunyai hubungan dengan pembuatan proyek akhir dari mulai pendahuluan, landasan teori, metode pelaksanaan, pembahasan, serta kesimpulan dan saran

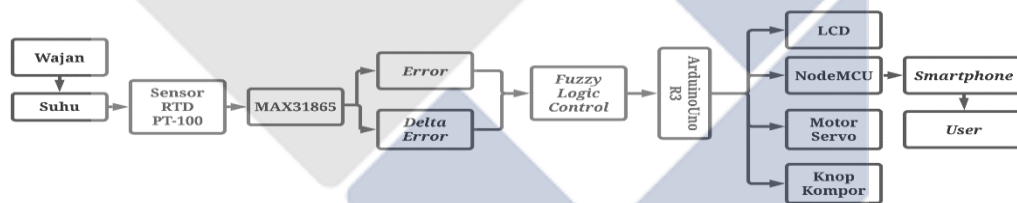


BAB IV PEMBAHASAN

Pada bab ini membahas tentang proses dalam pelaksanaan proyek akhir yang dimulai dari perancangan dan pembuatan sistem stabilisasi dan kontrol, proses perancangan dan pembuatan konstruksi, perancangan dan pembuatan sistem *monitoring* dan kontrol berbasis IoT, dan pengujian alat sebagai berikut

4.1 Deskripsi Alat

Rancang bangun sistem stabilisasi dan kontrol suhu penggorengan ini dirancang agar mempermudah proses *monitoring* suhu pada penggorengan secara *real time* menggunakan *smartphone*. Blok diagram prinsip kerja alat dapat dilihat pada Gambar 4.1 dibawah ini.



Gambar 4. 1 Blok Diagram Prinsip Kerja Rancang Bangun Sistem Stabilisasi dan Kontrol Suhu Penggorengan

Rancang bangun sistem stabilisasi suhu penggorengan dengan metode *fuzzy* berbasis *Internet of Things* (IoT) adalah sebuah alat yang digunakan untuk *monitoring* dan mengontrol suhu pada penggorengan. Alat ini menggunakan tegangan sumber dari adaptor 5V 2A untuk rangkaian kontrol dan adaptor 3.3V 1A untuk NodeMCU. Alat ini akan beroperasi jika saklar ditekan, maka akan menampilkan nilai ukur sensor pada LCD dan *smartphone* melalui aplikasi *Blynk*. Alat ini menggunakan sensor suhu RTD untuk parameter suhu penggorengan dengan satuan °C (celsius). Nilai ukur tersebut diproses oleh Arduino Uno R3 untuk menentukan tingkat kenaikan atau penurunan suhu dengan metode logika *fuzzy* Mamdani yang telah diprogram dan NodeMCU sebagai pengirim data ke *smartphone* melalui aplikasi *Blynk*. Alat ini juga ditambahkan sebuah *push button*

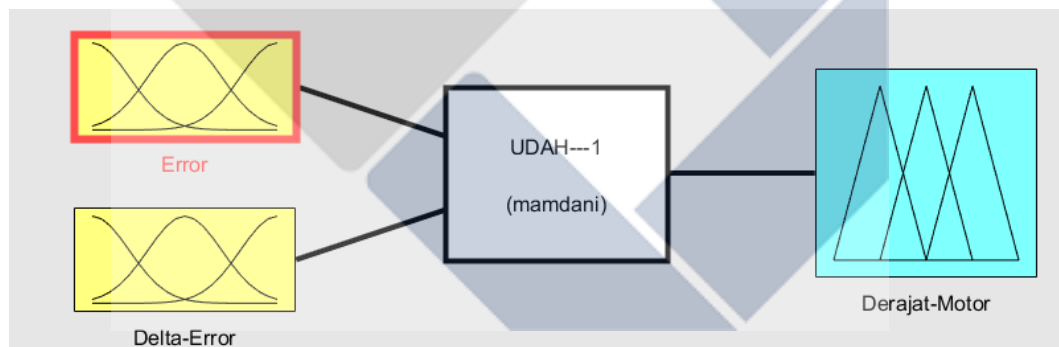
yang digunakan sebagai mulainya atau tidaknya alat bekerja, dan juga untuk tombol *emergency* sebagai *safety* apabila terjadi sesuatu yang tidak diinginkan.

4.2 Perancangan *Hardware* dan *Software* Alat *Monitoring* dan Kontrol Suhu Penggorengan berbasis IoT

Rangkaian kontrol sistem *Monitoring* dan Kontrol Suhu Penggorengan berbasis *Internet of Things* (IoT) ini dirangkai di dalam sebuah *box* berbentuk kotak yang ukurannya sudah disesuaikan. Ukuran dan bentuk *box* pada proyek akhir ini adalah 30x20x12cm. *Box* dengan bentuk kotak bertujuan untuk memudahkan peletakan komponen dan rangkaian kontrol, dan juga untuk mencegah agar tidak terkena cipratan minyak. Di dalam rangkaian ini terdapat sebuah Arduino Uno R3 yang digunakan untuk mengelola data *input* dan *output*. Alasan menggunakan Arduino Uno R3 dikarenakan ukurannya cukup standar untuk mempermudah dalam penyusunan dalam *box* dan penggunaan komponen *input* dan *output* pada proyek akhir ini sebanyak 12 pin sehingga cocok dalam proyek akhir ini karena tidak memerlukan banyak pin *input* dan *output*. NodeMCU digunakan sebagai perantara komunikasi perangkat keras dengan *smartphone* berbasis internet. NodeMCU juga digunakan untuk menampilkan data pada LCD yang diterima dari Arduino Uno R3 melalui komunikasi *serial*. Arduino Uno R3 dan NodeMCU memiliki fungsi masing-masing sehingga proses *monitoring* data dapat dilakukan secara *real time* tidak lambat. Sensor suhu RTD digunakan untuk mendeteksi suhu penggorengan, alasan menggunakan sensor RTD karena dari beberapa tipe sensor suhu, sensor RTD khusus pembacaannya lebih akurat lebih terulang. Diulang artinya pengguna yang membaca suhu yang sama menghasilkan hasil yang sama dalam beberapa percobaan. RTD menghasilkan pembacaan lebih berulang berarti bahwa pembacaan mereka lebih stabil. LCD 20x4 digunakan untuk menampilkan hasil *monitoring* dan di *couple* dengan modul I²C yang memudahkan dalam proses rangkaian karena dengan menggunakan modul ini pin pada LCD hanya menggunakan 4 pin. *Servo* Digital MG-996R sebagai kontrol motor DC. Pada proyek akhir ini menggunakan *Servo* Digital 180° untuk menggerakkan knop kompor. *Push button* sebagai tombol mulainya alat kerja dan juga sebagai tombol *emergency* yang akan melakukan

perpindahan ke 0° pada motor *servo*. Komponen tersebut disambungkan ke Arduino Uno R3 dan NodeMCU sebagai pengontrol alat ini. Sehingga setiap komponen memiliki pin tersendiri dan harus sama pada saat program di Arduino IDE.

Penerapan logika *fuzzy* pada proyek akhir ini diaplikasikan untuk menentukan tingkat kenaikan atau penurunan suhu berdasarkan *input* yaitu nilai *error* dari suhu set dan suhu RTD dengan rentang nilai -215 hingga 215°C . Sedangkan *input Delta error* mempunyai rentang nilai dari -10 hingga 10 . Dan *output* derajat motor memiliki rentang nilai 0° hingga 180° . Perancangan logika *fuzzy* ini berdasarkan penalaran dan pembacaan sensor yang digunakan, karena konsep logika *fuzzy* yang *fleksibel* dan sama dengan penalaran manusia. Desain FLC menggunakan 2 *input* dan 1 *output* sedangkan aturan *fuzzy* menggunakan metode Mamdani. Desain blok diagram *Fuzzy Logic Control* dapat dilihat dari Gambar 4.2.



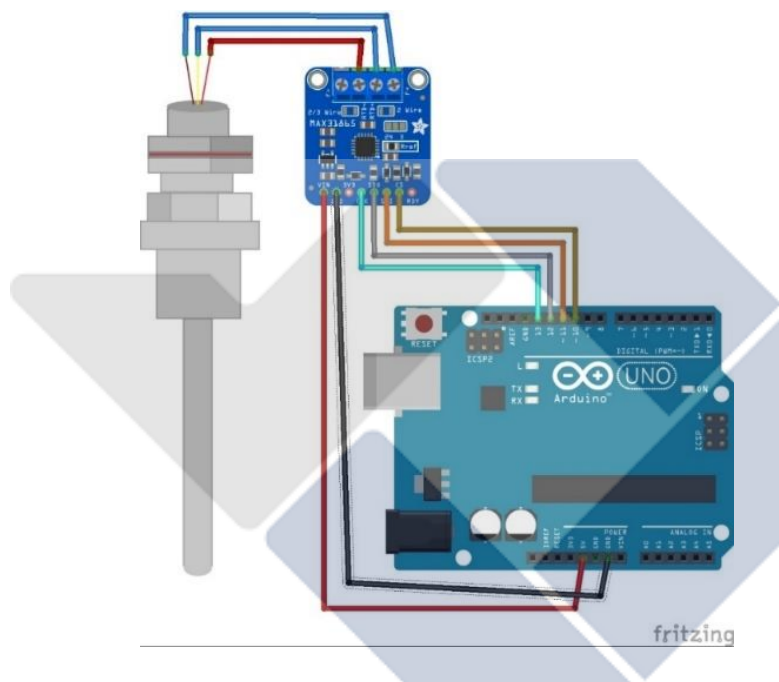
Gambar 4. 2 Desain blok diagram *Fuzzy Logic Control*

Untuk tampilan *software* IoT pada proyek akhir ini hanya memantau nilai kenaikan atau penurunan suhu, tingkat suhu dan kontrol suhu yang diinginkan. Nilai kenaikan atau penurunan suhu ditampilkan menggunakan *widget Value display* dengan rentang nilai dari 30°C - 240°C . Tampilan ini untuk memudahkan pengguna sehingga apabila suhu penggorengan tinggi maka pengguna bisa mengontrolnya ke suhu rendah dan nilai suhu akan ditampilkan pada tampilan aplikasi *Blynk*. Kontrol suhu yang disetel pengguna menggunakan *controllers button* ditampilkan dalam bentuk nilai menggunakan *widget value display* sehingga pengguna dapat mengetahui berapa nilai suhu penggorengan.

4.3 Pengujian Sensor RTD PT-100

Proses pengujian alat sistem stabilisasi suhu penggorengan dengan menguji masing-masing komponen elektrik. Tujuan pengujian ini yaitu untuk mengetahui apakah komponen dalam keadaan baik atau tidak dan apakah komponen bisa bekerja dengan fungsi yang diinginkan atau tidak. Pengujian ini juga bertujuan untuk mengambil data yang diperlukan pada proyek akhir ini.

4.3.1 Perancangan dan pembuatan Sensor RTD PT-100



Gambar 4. 3 Rangkaian *Hardware* Dari Pengujian Suhu Menggunakan Sensor RTD PT-100.

Pada Gambar 4.3 merupakan rangkaian *hardware* dari pengujian suhu menggunakan sensor RTD PT-100. Sensor RTD PT-100 dirangkai pada MAX31865 terlebih dahulu dan dirangkai pada Arduino Uno R3.

Pengujian sensor RTD PT-100 dilakukan menggunakan pemrograman Arduino Uno dengan list sebagai berikut:

```
#include <Adafruit_MAX31865.h>
int suhu, datalama, selisih;
void setup() {
```

```

    Serial.begin(115200);
    Serial.println("Adafruit MAX31865 PT100 Sensor
Test!");
    max.begin(MAX31865_3WIRE); // set to 2WIRE or 4WIRE
as necessary
void loop() {
    // Rumus RTD
    uint16_t rtd = max.readRTD();
    Serial.print("RTD value: ");
    Serial.println(rtd);
    float ratio = rtd;
    ratio /= 32768;
    suhu = max.temperature(RNOMINAL, RREF);
    datalama= suhu;
    Serial.print("Ratio = ");
    Serial.println(ratio, 8);
    Serial.print("Resistance = ");
    Serial.println(RREF * ratio, 8);
    Serial.print("Temperature = ");
    Serial.println(suhu);

selisih = suhu-datalama;
Serial.print("selisih = ");
    Serial.println(selisih);
    uint8_t fault = max.readFault();
    if (fault) {
        Serial.print("Fault 0x"); Serial.println(fault,
HEX);
        if (fault & MAX31865_FAULT_HIGHTHRESH) {
            Serial.println("RTD High Threshold");
        }
        if (fault & MAX31865_FAULT_LOWTHRESH) {

```

```

        Serial.println("RTD Low Threshold");
    }
    if (fault & MAX31865_FAULT_REFINLOW) {
        Serial.println("REFIN- > 0.85 x Bias");
    }
    if (fault & MAX31865_FAULT_REFINHIGH) {
        Serial.println("REFIN- < 0.85 x Bias - FORCE-
open");
    }
    if (fault & MAX31865_FAULT_RTDINLOW) {
        Serial.println("RTDIN- < 0.85 x Bias - FORCE-
open");
    }
    if (fault & MAX31865_FAULT_OVUV) {
        Serial.println("Under/Over voltage");
    }
    max.clearFault();
}
Serial.println();
delay(1000);
}

```

Dibawah ini merupakan tabel hasil pengujian Sensor RTD PT-100 yang dapat dilihat pada Tabel 4.1

Tabel 4. 1 Hasil Pengujian Sensor RTD PT-100

No.	Derajat Motor(°)	Thermometer (°C)	Sensor (°C)	Error (%)
1.	0	32.9	32.14	2.31
2.	10	60.6	60	0.9
3.	20	68.9	68	1.3
4.	30	76.1	75	1.44
5.	40	84.2	83	1.42

No.	Derajat Motor(°)	<i>Thermometer</i> (°C)	Sensor (°C)	<i>Error</i> (%)
6.	50	88.3	88	0.33
7.	60	93.9	93	0.95
8.	70	103	101	1.94
9.	80	113.1	111	1.85
10.	90	121	118	2.47
11.	100	135	130	3.70
12.	110	140	136	2.85
13.	120	145	143	1.37
14.	130	153	148	3.26
15.	140	162	154	4.93
16.	150	170	160	5.88
17.	160	178	173	2.80
18.	170	191	186	2.61
19.	180	213	203	4.69

Pada Tabel 4.1 di atas diketahui bahwa terdapat perbedaan suhu yang dihasilkan oleh *thermometer* dan sensor suhu *RTD*. Terhitung rata-rata *error* yang dihasilkan yaitu 2.47%.

Kemudian perhitungan persentase *error* (%) yang didapatkan pada pengujian sensor suhu, berikut rumus untuk mencari persentase *error*:

$$\%error = \left| \frac{Thermometer - sensor}{Thermometer} \right| \times 100\%$$

$$\%error = \left| \frac{60,6 - 60}{60,6} \right| \times 100\%$$

$$\%error = 0,9\% \dots \dots \dots (4.2)$$

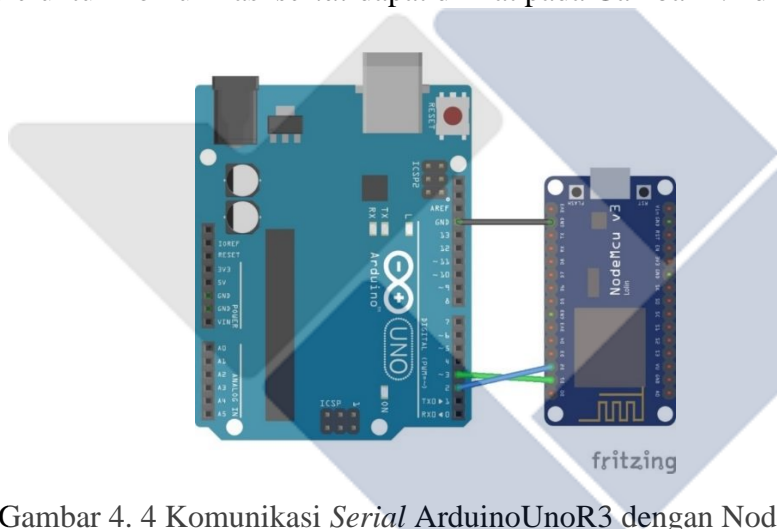
4.3.2 Menampilkan Data di LCD 20x4 dan Modul I²C

LCD 20x4 digunakan untuk menampilkan nilai yang terbaca oleh sensor RTD dan derajat motor. LCD juga akan memberikan informasi kondisi berdasarkan

data sensor yang diproses oleh ArduinoUnoR3. Modul I²C digunakan untuk mempermudah merangkai karena LCD hanya memerlukan 4 pin.

4.3.3 Komunikasi *Serial* ArduinoUnoR3 dengan NodeMCU

Komunikasi *serial* pada proyek akhir ini menggunakan ArduinoUnoR3 dengan NodeMCU. Komunikasi *serial* berfungsi untuk mengirimkan data hasil pembacaan sensor dari ArduinoUnoR3 yang dikirimkan ke NodeMCU untuk ditampilkan pada LCD 20x4 I2C dan monitoring pada *smartphone*. Pin yang digunakan yaitu D1 dan D2 dari NodeMCU. D1 dihubungkan ke pin 3 dan D2 dihubungkan ke pin 2 ArduinoUnoR3 sebagai komunikasi *serial*. Rangkaian *hardware* untuk komunikasi *serial* dapat dilihat pada Gambar 4.4 di bawah ini.



Gambar 4. 4 Komunikasi *Serial* ArduinoUnoR3 dengan NodeMCU.

4.3.4 Pengujian dari NodeMCU ke *Smartphone*

Pengujian dari NodeMCU ke *smartphone* menggunakan aplikasi *Blynk* dengan cara menghubungkan NodeMCU ke Wi-Fi. Cara pengujian sebagai berikut:

1. Untuk melakukan pengujian dari NodeMCU ke *Smartphone*, perlu dilakukan penambahan *library* pada program. Dan pembuatan proyek pada aplikasi *Blynk* untuk menampilkan nilai suhu (°C) dan nilai derajat motor (°) dengan cara menambahkan *widget box Gauge*. Dengan cara menambahkan *token* program *Blynk* yang telah dikirim ke *email* kita. Kemudian *upload* program.
2. Setelah itu memantau nilai yang ada di *serial monitor* dan di aplikasi *Blynk* yang telah dibuat bertujuan untuk melihat apakah data yang dikirim sama

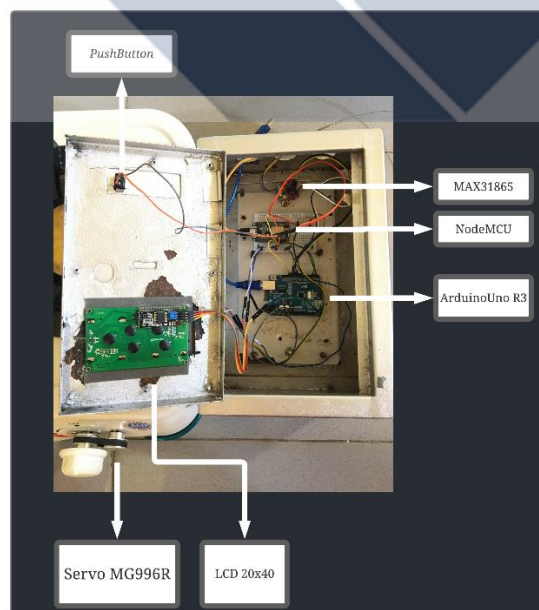
atau tidak antara data yang tampil pada *serial monitor* dan data yang tampil pada aplikasi *Blynk*.

4.3.5 Pengujian Logika *Fuzzy*

Pengujian sistem kerja FLC dilakukan dengan mengkombinasikan kemungkinan-kemungkinan yang ada, secara umum dilakukan dengan cara melakukan perubahan nilai *input* dengan kondisi 1 *input* tetap. Pengujian dilakukan dengan membandingkan hasil menggunakan simulasi Matlab dengan pemrograman arduino yang bertujuan untuk memastikan pemrograman logika *fuzzy* proyek akhir ini layak digunakan atau tidak.

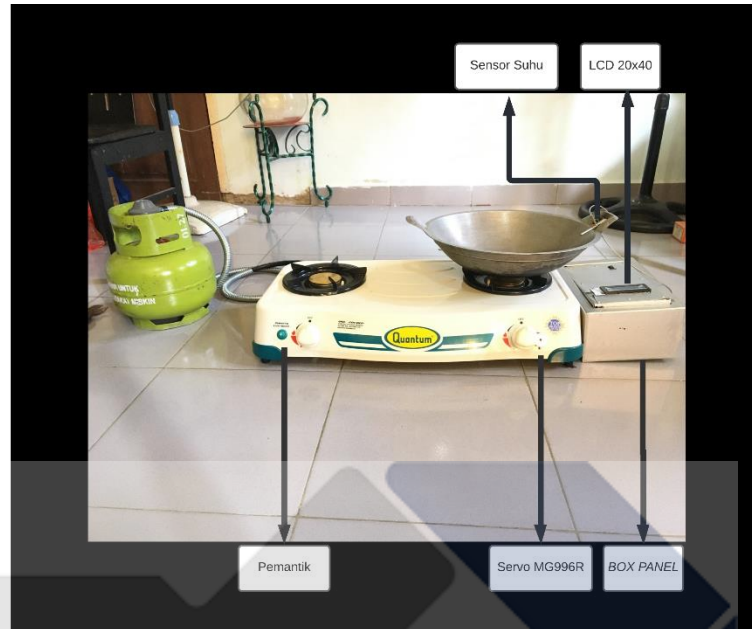
4.4 Pengujian Rancang Bangun Sistem Stabilisasi Suhu Pengorengan dengan Metode *Fuzzy* Berbasis IoT

Pengujian ini bertujuan untuk melihat apakah proyek akhir yang berjudul Rancang Bangun Sistem Stabilisasi Suhu Pengorengan dengan Metode *Fuzzy* Berbasis IoT bisa digunakan langsung atau tidak dan apakah suhu bisa terbaca oleh sensor atau tidak. Pengujian ini juga bertujuan untuk menciptakan suatu alat yang dapat mengantisipasi terjadinya kekosongan akibat ketidakstabilan suhu. Dapat dilihat pada Gambar 4.5 pemasangan rangkaian alat pada *box*.



Gambar 4. 5 Pemasangan Alat pada *Box*

Dibawah ini Gambar 4.6 menunjukkan gambar konstruksi keseluruhan pada proyek akhir ini.



Gambar 4. 6 Konstruksi Alat

4.4.1. Pengujian Suhu Penggorengan dengan *Input Rendah*

Pada pengujian suhu penggorengan dengan *input* rendah(70°C), dengan waktu pembacaan data setiap 5 detik.

Tabel 4. 2 Pengujian suhu penggorengan dengan *input* rendah

<i>Set point</i>	<i>Input(°C)</i>		Waktu (s)	Sensor (°C)	<i>Output (°)</i>		<i>%Error (%)</i>
	<i>Error</i>	<i>Delta error</i>			Knop Servo	Matlab	
70	27	2.71	0	42.71	30.78	30.8	0.064%
70	26	0.83	5	43.29	65.39	65.4	0.076%
70	24	1.28	10	45.62	65.38	65.4	0.091%
70	22	2.05	15	47.19	65.38	65.4	0.091%
70	19	1.36	20	50.72	4.61	4.63	0.431%
70	17	2.58	25	51.37	5.38	5.37	0.186%

<i>Set point</i>	<i>Input(°C)</i>		<i>Waktu (s)</i>	<i>Sensor (°C)</i>	<i>Output (°)</i>		<i>%Error (%)</i>
	<i>Error</i>	<i>Delta error</i>			<i>Knop Servo</i>	<i>Matlab</i>	
70	15	1.68	30	53.83	5.03	5.07	0.788%
70	13	0.51	35	55.20	4.96	4.99	0.601%
70	12	1.62	40	57.37	5.25	5.26	0.190%
70	10	1.93	45	60.46	5.16	5.18	0.386%
70	8	0.48	50	62.49	5.02	5.05	0.594%
70	6	1.38	55	64.91	5.17	5.19	0.385%
70	5	0.64	60	66.15	4.82	4.83	0.207%
70	4	0.74	65	67.31	4.65	4.66	0.214%
70	3	1.7	70	67.94	4.78	4.8	0.416%
70	2	0.61	75	68.47	4.62	4.63	0.215%
70	1	0.72	80	69.45	4.62	4.65	0.645%
70	0	0.21	85	70.63	19.99	20	0.05%
<i>ERROR</i>							0.31%

Dari hasil pengujian suhu penggorengan dengan *input* rendah dapat diketahui bahwa rata-rata nilai persentase *error* suhu yang terbaca sebesar 0.31%. Maka dari itu nilai yang terbaca oleh sensor dapat bekerja secara optimal.

4.4.2. Pengujian Suhu Penggorengan dengan *Input* Sedang

Pada pengujian suhu penggorengan dengan *input* sedang (170°C), dengan waktu pembacaan data setiap 5 detik.

Pada pengujian suhu penggorengan dengan *input* sedang dimulai dari suhu 60.90°C

Tabel 4. 3 Pengujian suhu penggorengan dengan *input* sedang

<i>Set point</i>	<i>Input</i> (°C)		Waktu (s)	Sensor (°C)	<i>Output</i> (°)		<i>%Error</i> (%)
	<i>Error</i>	<i>Delta error</i>			Knop Servo	Matlab	
170	109	0.10	0	60.90	65.27	65.3	0.04%
170	108	0.45	5	61.55	65.32	65.3	0.015%
170	107	0.42	10	62.58	65.32	65.3	0.015%
170	105	0.44	15	64.56	65.32	65.3	0.015%
170	104	1.53	20	65.42	65.27	65.3	0.076%
170	100	1.20	25	69.80	65.31	65.3	0.030%
170	98	1.04	30	71.96	65.36	65.4	0.122%
170	95	2.87	35	74.13	30.39	30.4	0.032%
170	92	2.51	40	77.49	65.32	65.4	0.183%
170	83	1.56	45	86.44	65.28	65.3	0.076%
170	80	1.87	50	89.13	65.29	65.3	0.061%
170	78	1.70	55	91.30	65.28	65.3	0.076%
170	76	1.80	60	93.20	65.29	65.3	0.061%
170	74	2.66	65	95.34	30.90	30.90	0%
170	66	0.08	70	103.92	65.30	65.4	0.152%
170	65	0.66	75	104.34	65.37	65.4	0.107%
170	64	2.42	80	105.58	65.37	65.4	0.107%
170	61	1.57	85	108.43	65.31	65.4	0.198%
170	59	1.63	90	110.37	65.31	65.4	0.198%
170	56	1.09	95	113.91	65.34	65.4	0,0917%

<i>Set point</i>	<i>Input(°C)</i>		<i>Waktu (s)</i>	<i>Sensor (°C)</i>	<i>Output (°)</i>		<i>%Error (%)</i>
	<i>Error</i>	<i>Delta error</i>			<i>Knop Servo</i>	<i>Matlab</i>	
170	54	1.07	100	115.93	65.35	65.4	0.076%
170	50	0.52	105	119.48	65.34	65.4	0.091%
170	49	0.76	110	120.24	65.39	65.4	0.015%
170	47	0.91	115	122.09	65.40	65.4	0%
170	44	1.85	120	126.30	65.40	65.4	0%
170	42	2.18	125	128.48	65.35	65.3	0.076%
170	45	3.3	130	131.72	30.39	30.4	0.164%
170	47	2.65	135	134.37	30.76	30.8	0.032%
170	43	3.81	140	138.18	30.38	30.4	0.065%
170	40	0.32	145	141.24	65.35	65.4	0.076%
170	38	1.72	150	143.76	65.47	65.4	0.107%
170	35	2.87	155	146.21	30.38	30.4	0.065%
170	32	0.21	160	149.45	65.31	65.4	0.1376%
170	28	3.49	165	151.97	30.42	30.4	0.066%
170	26	1.87	170	153.54	65.42	65.4	0.030%
170	23	2.39	175	155.29	65.39	65.4	0.076%
170	21	0.37	180	156.78	65.47	65.4	0.10%
170	18	1.31	185	158.84	4.56	4.66	2.14%
170	15	2.62	190	161.28	5.40	5.37	0.55%
170	13	0.56	195	162.21	4.89	4.92	0.60%
170	12	0.87	200	163.72	4.70	4.68	0.42%

<i>Set point</i>	<i>Input(°C)</i>		<i>Waktu (s)</i>	<i>Sensor (°C)</i>	<i>Output (°)</i>		<i>%Error (%)</i>
	<i>Error</i>	<i>Delta error</i>			<i>Knop Servo</i>	<i>Matlab</i>	
170	10	1.73	205	165.48	5.19	5.25	1.14%
170	9	0.26	210	166.37	5.16	5.23	1.33%
170	8	0.83	215	167.27	4.60	4.63	0.64%
170	7	0.39	220	167.94	5.15	5.18	0.579%
170	6	0.62	225	168.24	4.73	4.85	2.47%
170	5	1.16	230	168.98	4.90	4.91	0.203%
170	4	0.96	235	169.25	4.77	4.76	0.210%
170	2	1.47	240	169.61	4.59	4.6	0.217%
170	2	0.79	245	169.91	4.59	4.6	0.217%
170	1	0.83	250	170.83	4.57	4.63	1.295%
170	0	0.98	255	171.38	19.98	20	0.1%
<i>ERROR</i>							0.29%

Dari hasil pengujian suhu penggorengan dengan *input* sedang dapat diketahui bahwa rata-rata nilai persentase *error* suhu yang terbaca sebesar 0.29%. Maka dari itu nilai yang terbaca oleh sensor dapat bekerja secara optimal.

4.4.3. Pengujian Suhu Penggorengan dengan *Input Tinggi*

Pada pengujian suhu penggorengan dengan *input error* (210°C), dengan waktu pembacaan data setiap 5 detik.

Tabel 4. 4 Pengujian suhu penggorengan dengan *input tinggi*

<i>Set point (°C)</i>	<i>Input</i>		<i>Waktu (s)</i>	<i>Sensor (°C)</i>	<i>Output (°)</i>		<i>%Error(%)</i>
	<i>Error</i>	<i>Delta error</i>			<i>Knop Servo</i>	<i>Matlab</i>	
210	30	2.41	0	180.40	65.37	65.4	0.107%

<i>Set point</i>	<i>Input(°C)</i>		<i>Waktu (s)</i>	<i>Sensor (°C)</i>	<i>Output (°)</i>		<i>%Error(%)</i>
	<i>Error</i>	<i>Delta error</i>			<i>Knop Servo</i>	<i>Matlab</i>	
210	28	0.24	5	182.62	65.37	65.4	0.107%
210	25	1.49	10	185.26	65.39	65.4	0.076%
210	23	1.96	15	186.71	65.36	65.4	0.122%
210	20	2.61	20	188.27	5.39	5.37	0.372%
210	18	0.26	25	190.21	4.65	4.66	0.214%
210	16	1.40	30	192.41	4.90	4.92	0.406%
210	13	1.93	35	195.28	5.14	5.19	0.963%
210	10	2.59	40	197.75	5.33	5.37	0.744%
210	7	0.39	45	199.12	5.16	5.18	0.386%
210	6	0.76	50	201.27	4.59	4.62	0.649%
210	5	0.83	55	202.80	4.60	4.63	0.647%
210	4	0.72	60	204.37	4.67	4.71	0.849%
210	3	0.82	65	206.82	4.60	4.62	0.432%
210	2	0.93	70	207.61	4.61	4.65	0.860%
210	1	1.04	75	208.79	4.66	4.65	0.215%
210	0	0.74	80	209.23	19.97	20	0.15%
<i>ERROR</i>							0.43%

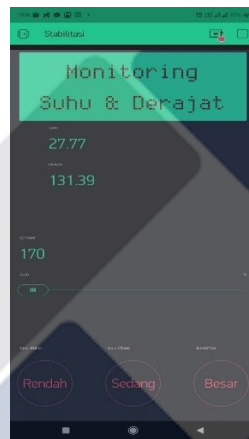
Dari hasil pengujian suhu penggorengan dengan *input* tinggi dapat diketahui bahwa rata-rata nilai persentase *error* suhu yang terbaca sebesar 0.43%. Maka dari itu nilai yang terbaca oleh sensor dapat bekerja secara optimal.

Dari hasil pengujian sensor suhu penggorengan dengan *input* rendah, sedang dan tinggi dengan rata-rata persentase *error* berturut-turut 0.31%, 0.29%, dan 0.43%. Hasil Pengujian sensor suhu pada tabel 4.2 dengan *input* suhu rendah 70°C, pada 0-20 detik suhu masih kurang dari 70°C, maka *output* berupa derajat

motor akan membesar. Akan tetapi jika telah mendekati 70°C , maka *output* motor akan langsung mengecil. Pengujian ini dengan cara meletakkan sensor suhu *RTD* pada pinggiran wajan. Setelah mendapat nilai pada sensor suhu, kemudian kontrol *fuzzy* akan menstabilkan derajat knop kompor.

4.4.4. Pengujian *Monitoring* dan Kontrol Menggunakan *Blynk*

Tampilan dari hasil pengujian sistem *monitoring* dan kontrol suhu penggorengan untuk parameter suhu, dan derajat motor menggunakan *Blynk* pada *smartphone*. Pada Gambar 4.7 dibawah ini merupakan tampilan aplikasi *Blynk* untuk sistem *monitoring* dan kontrol melalui *smartphone*.



Gambar 4. 7 Sistem *Monitoring* dan sistem kontrol menggunakan aplikasi *Blynk* pada *Smartphone*

Dari hasil pengujian sensor suhu dan derajat motor menggunakan *Blynk* pada *Smartphone* akan menampilkan nilai secara *real time* dan data akan tersimpan pada *Blynk*.

BAB V PENUTUP

5.1 Kesimpulan

Setelah melakukan tahap perancangan dan pembuatan rangkaian kontrol dan sistem *monitoring* yang kemudian dilanjutkan dengan tahap pengujian, berdasarkan hasil data yang diperoleh melalui pengujian dapat disimpulkan sebagai berikut :

1. Berdasarkan Pengujian suhu penggorengan dengan *input* rendah mencapai nilai *set point* 70°C berada pada waktu 85 detik dengan nilai *error* awal sebesar 27. Kemudian pengujian suhu penggorengan dengan *input* sedang memerlukan waktu 255 detik atau 4 menit 25 detik untuk mencapai nilai *set point* 170°C dengan nilai *error* awal sebesar 109. Sedangkan untuk *input* tinggi memerlukan waktu sebesar 80 detik untuk mencapai nilai *set point* 210 dengan nilai *error* awal sebesar 30.
2. Dari hasil pengukuran suhu penggorengan didapatkan bahwa rata-rata persentase *error* dari data hasil pengujian sensor suhu penggorengan dengan *input* rendah adalah 0.31%, *input* sedang adalah 0.29%, dan *input* tinggi adalah 0.43%. Jadi, keakuratan dari sistem stabilisasi suhu penggorengan dengan metode *fuzzy* berbasis IoT sebesar 99.57% - 99,71%. Jadi dari hasil pengujian ini dapat disimpulkan bahwa suhu untuk menggoreng kerupuk kemplang yang baik pada *input* suhu sedang yaitu di 170°C-200°C
3. Sistem kontrol berupa tindakan mengatur *input set point* untuk mengatur suhu yang dapat dilakukan secara manual oleh pengguna melalui *smartphone* dari jarak jauh selama NodeMCU ESP8266 masih terhubung dengan jaringan.

5.2 Saran

Proyek akhir ini masih terdapat beberapa kekurangan yang dapat menjadi saran untuk pengembangan proyek akhir ke depannya, yaitu:

1. Saat ini alat proyek akhir menggunakan cara manual untuk menghidupkan kompor, yaitu dengan cara menekan terlebih dahulu katup dan sedikit diputar. Untuk itu butuh inovasi lebih untuk mengembangkan proyek akhir ini agar dapat dijalankan secara otomatis secara keseluruhan.
2. Pengiriman data dari Arduino Uno ke *smartphone* dan sebaliknya mengalami *delay* karena program pada Arduino yang terlalu banyak. Untuk pengiriman data yang lebih baik maka perlu dilakukan perbaikan pada program untuk mengurangi *delay* tersebut.
3. Saat ini alat proyek akhir tidak dapat menurunkan suhu dikarenakan api yang terus menyala. Untuk itu butuh inovasi untuk dapat menurunkan suhu secara cepat.

DAFTAR PUSTAKA

- [1] H. Harris, D. Efreza, and I. Nafsiyah, “pengembangan-industri-tepung-ikan-dari-limbah-pengolahan-makanan-tradisional,” 2012.
- [2] T. Eka Prasasti, J. Susila, and J. Priambodo, “Sistem Kendali pada Mesin Produksi Kerupuk Mawar Menggunakan Networked Control System,” 2021.
- [3] N. Huda, A. Li Leng, and C. Xian Yee, “Asian Journal of Food and Agro-Industry Chemical composition, colour and linear expansion properties of Malaysian commercial fish cracker (keropok),” *As. J. Food Ag-Ind*, vol. 3, no. 05, pp. 473–482, 2010, [Online]. Available: www.ajofai.info
- [4] S. Ratna Sari, F. Pratama, T. Wardani Widowati, D. Prariska, I. Perikanan, and U. Sumatera Selatan, “Karakteristik Sensoris Microwaveable Kemplang Palembang dengan Perbedaan Ketebalan dan Level Daya pada Proses Pematangan Sensory Characteristics Microwaveable of fish creakers with Differences Thickness and Power Level in cook processing,” 2020.
- [5] F. Nahdi and H. Dhika, “Analisis Dampak Internet of Things (IoT) Pada Perkembangan Teknologi di Masa Yang Akan Datang 33,” 2021.
- [6] S. Indrawati, S. M. B. Respati, and Darmanto, “KEBUTUHAN DAYA PADA AIR CONDITIONERSAAT TERJADI PERBEDAANSUHU DAN KELEMBABAN,” vol. 15, pp. 91–95, 2019.
- [7] D. Sanzay Sirait, M. Hadi Santosa, and J. Teknik Elektro, “SISTEM OTOMASI PROSES LIKUIFIKASI DALAM PEMBUATAN SETENGAH JADI GULA CAIR DARI UBI SINGKONG,” *Scientific Journal Widya Teknik*, vol. 19, no. 2, p. 2020.

- [8] “General Description Benefits and Features • Integration Lowers System Cost, Simplifies Design Efforts, and Reduces Design Cycle Time • Simple Conversion of Platinum RTD Resistance to Digital Value • Handles 100 Ω to 1k Ω (at 0°C) Platinum RTDs (PT100 to PT1000) • Compatible with 2-, 3-, and 4-Wire Sensor Connections • SPI-Compatible Interface • 20-Pin TQFN and SSOP Packages.” [Online]. Available: www.maximintegrated.com
- [9] F. Wahab, A. Sumardiono, A. Rafi, A. Tahtawi, A. Faisal, and A. Mulayari, “Desain dan Purwarupa Fuzzy Logic Control untuk Pengendalian Suhu Ruangan,” *Direvisi: 23 Mei*, vol. 2, no. 1, p. 22, 2017.
- [10] S. Komariyah, R. M. Yunus, and S. F. Rodiyansyah, “LOGIKA FUZZY DALAM SISTEM PENGAMBILAN KEPUTUSAN PENERIMAAN BEASISWA.”
- [11] A. Syafii, A. Mursito, A. Muhlisin, A. Widiarto, A. Saifudin, and E. Nirmala, “Jurnal Teknologi Sistem Informasi dan Aplikasi Sistem Kendali Pengatur Suhu Ruangan pada Smart Building dengan Aplikasi Telegram menggunakan Fuzzy Logic Control,” vol. 4, no. 3, pp. 138–144, 2021, doi: 10.32493/jtsi.v4i3.8892.
- [12] I. Dwisaputra *et al.*, “PENGARUH JUMLAH INPUT DAN MEMBERSHIP FUNCTION FUZZY LOGIC CONTROL PADA ROBOT KESEIMBANGAN BERODA DUA.”
- [13] Yulmaini, “PENGUNAAN METODE FUZZY INFERENCE SYSTEM (FIS) MAMDANI DALAM PEMILIHAN PEMINATAN MAHASISWA UNTUK TUGAS AKHIR Yulmaini.”
- [14] M. Sumitre and R. Kurniawan, “RANCANG BANGUN SISTEM PENDUKUNG KEPUTUSAN SELEKSI PENERIMAAN TENAGA PENGAJAR DENGAN METODE FUZZY INFERENCE SYSTEM (FIS) MAMDANI,” 2014.

- [15] A. W. Dani, A. Mundhola, R. Rahmatullah, and A. Mundhofa, "Perangkat Uji Penciuman sebagai Protokol Kesehatan Menggunakan Fuzzy Mamdani Berbasis Internet Of Things," *Jurnal Teknologi Elektro*, vol. 12, no. 3, p. 160, 2021, doi: 10.22441/jte.2021.v12i3.010.
- [16] H. Susanto, R. Pramana, and M. Mujahidin, "PERANCANGAN SISTEM TELEMETRI WIRELESS UNTUK MENGUKUR SUHU DAN KELEMBABAN BERBASIS ARDUINO UNO R3 ATMEGA328P DAN XBEE PRO," 2013. [Online]. Available: <https://www.researchgate.net/publication/344520778>
- [17] D. Auliya Saputra, N. Utami, and R. Setiawan, "RANCANG BANGUN ALAT PEMBERI PAKAN IKAN MENGGUNAKAN MIKROKONTROLER," 2020.
- [18] A. Hilal and S. Manan, "PEMANFAATAN MOTOR SERVO SEBAGAI PENGGERAK CCTV UNTUK MELIHAT ALAT-ALAT MONITOR DAN KONDISI PASIEN DI RUANG ICU," 2012.
- [19] "MG996R High Torque Metal Gear Dual Ball Bearing Servo."
- [20] "Handson Technology User Guide I2C Serial Interface 1602 LCD Module." [Online]. Available: www.handsontec.com
- [21] R. Ayu Dewi Sartika Departemen Gizi Kesehatan Masyarakat and F. Kesehatan Masyarakat, "PENGARUH SUHU DAN LAMA PROSES MENGGORENG (DEEP FRYING) TERHADAP PEMBENTUKAN ASAM LEMAK TRANS," 2009.
- [22] M. Natsir, D. Bayu Rendra, and A. Derby Yudha Anggara, "IMPLEMENTASI IOT UNTUK SISTEM KENDALI AC OTOMATIS PADA RUANG KELAS DI UNIVERSITAS SERANG RAYA," vol. 6, no. 1, 2019, [Online]. Available: <https://www.arduino.cc/en/Products/Counterfeit>

- [23] “The Internet of Things How the Next Evolution of the Internet Is Changing Everything,” 2011.
- [24] N. H. L. Dewi, M. F. Rohmah, and S. Zahara, “PROTOTYPE SMART HOME DENGAN MODUL NODEMCU ESP8266 BERBASIS INTERNET OF THINGS (IOT)”.
- [25] A. Satriadi and dan Yuli Christiyono, “PERANCANGAN HOME AUTOMATION BERBASIS NodeMCU.” [Online]. Available: <https://ejournal3.undip.ac.id/index.php/transient>
- [26] Suherman, I. Andriyanto, and S. Dwiyatno, “RANCANG BANGUN ALAT UKUR TEMPERATUR SUHU PERANGKAT SERVERMENGUNAKAN SENSOR LM35BERBASIS SMS GATEWAY,” vol. 2, 2015.
- [27] D. Nurlette and T. K. Wijaya, “PERANCANGAN ALAT PENGUKUR TINGGI DAN BERAT BADAN IDEAL BERBASIS ARDUINO,” *Sigma Teknika*, vol. 1, no. 2, pp. 172–184, 2018.
- [28] E. S. Puspita and L. Yulianti, “PERANCANGAN SISTEM PERAMALAN CUACA BERBASIS LOGIKA FUZZY,” 2016.



DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Karina Zaruska
Tempat, Tanggal Lahir : Sungailiat, 15 Juli 2000
Alamat Rumah : Jl. Senang Hati IV No.1
No.HP : 081285181462
Email : karinazaruska15@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

- | | |
|---|---------------|
| 1. SD Negeri 3 Sungailiat | Lulus 2012 |
| 2. SMP Negeri 1 Sungailiat | Lulus 2015 |
| 3. SMA Negeri 1 Sungailiat | Lulus 2018 |
| 4. Politeknik Manufaktur Negeri Bangka Belitung | 2018-sekarang |



Sungailiat, 19 Januari 2022

Karina Zaruska

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Muhamad Azhari
Tempat, Tanggal Lahir : Puding Besar, 21 Oktober 1999
Alamat Rumah : Jl. Raya Pangkal Pinang No.16
Dusun V RT.11 Puding Besar,
Kepulauan Bangka Belitung.
No.HP : 085764441374
Email : azharia59@gmail.com
Jenis Kelamin : Laki-Laki
Agama : Islam



2. Riwayat Pendidikan

- | | |
|---|---------------|
| 5. SD Negeri 2 Puding Besar | Lulus 2012 |
| 6. SMP Negeri 1 Puding Besar | Lulus 2015 |
| 7. SMA Negeri 1 Puding Besar | Lulus 2018 |
| 8. Politeknik Manufaktur Negeri Bangka Belitung | 2018-sekarang |

Sungailiat, 19 Januari 2022

Muhamad Azhari



PROGRAM ARDUINO

```
#include <Fuzzy.h>
#include <Adafruit_MAX31865.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

Servo myservo;
int pos = 0;
int pin_pot1 = 10;

// Use software SPI: CS, DI, DO, CLK
Adafruit_MAX31865 max = Adafruit_MAX31865(10, 11, 12,
13);
// use hardware SPI, just pass in the CS pin
//Adafruit_MAX31865 max1 = Adafruit_MAX31865(10);

// The value of the Rref resistor. Use 430.0 for PT100
and 4300.0 for PT1000
#define RREF      430.0
// The 'nominal' 0-degrees-C resistance of the sensor
// 100.0 for PT100, 1000.0 for PT1000
#define RNOMINAL 100.0

#include<SoftwareSerial.h>

SoftwareSerial Arduino_SoftSerial(2,3);
float temp, output;

//below global variabel
char c;
```

```
String dataIn1;
int8_t indexOfC;

String data1;
int data2;
int pb = 8;
int datalama, selisih;

LiquidCrystal_I2C lcd(0x23, 20, 4);

// Instantiating a Fuzzy object
Fuzzy *fuzzy = new Fuzzy();

void setup()
{
  Serial.begin(115200);
  Serial.println("Adafruit MAX31865 PT100 Sensor
Test!");
  Arduino_SoftSerial.begin(9600);

  max.begin(MAX31865_3WIRE); // set to 2WIRE or 4WIRE
as necessary
  // initialize the LCD
  lcd.begin();

  myservo.attach(9);

  // Turn on the backlight and print a message.

  lcd.backlight();
```

```
lcd.setCursor(0,0);
lcd.print("Hallo");
delay(1000);
lcd.clear();

// Set a random seed
//randomSeed(analogRead(0));
pinMode(9, OUTPUT);

    // Instantiating a FuzzyInput object
FuzzyInput *distance = new FuzzyInput(1);
// Instantiating a FuzzySet object
FuzzySet *dis1 = new FuzzySet(-0.6, 10, 10, 20.8);
// Including the FuzzySet into FuzzyInput
distance->addFuzzySet(dis1);
    // Instantiating a FuzzySet object
FuzzySet *dis2 = new FuzzySet(20, 100, 100, 215);
// Including the FuzzySet into FuzzyInput
distance->addFuzzySet(dis2);
// Instantiating a FuzzySet object
FuzzySet *dis3 = new FuzzySet(-20.8, -10, -10, 0.2);
// Including the FuzzySet into FuzzyInput
distance->addFuzzySet(dis3);
// Instantiating a FuzzySet object
FuzzySet *dis4 = new FuzzySet(-215, -100, -100, -20);
// Including the FuzzySet into FuzzyInput
distance->addFuzzySet(dis4);
// Including the FuzzyInput into Fuzzy
fuzzy->addFuzzyInput(distance);

    // Instantiating a FuzzyInput object
```

```
FuzzyInput *difference = new FuzzyInput(2);
// Instantiating a FuzzySet object
FuzzySet *dif1 = new FuzzySet(-0.8, 0, 0, 1);
// Including the FuzzySet into FuzzyInput
difference->addFuzzySet(dif1);
FuzzySet *dif2 = new FuzzySet(0.9, 1.5, 1.5, 2.6);
// Including the FuzzySet into FuzzyInput
difference->addFuzzySet(dif2);
FuzzySet *dif3 = new FuzzySet(2.7, 6, 6, 10.1);
// Including the FuzzySet into FuzzyInput
difference->addFuzzySet(dif3);
// Instantiating a FuzzySet object
FuzzySet *dif4 = new FuzzySet(-2.5, -1, -1, -0.4);
// Including the FuzzySet into FuzzyInput
difference->addFuzzySet(dif4);
// Instantiating a FuzzySet object
FuzzySet *dif5 = new FuzzySet(-10, -6, -6, -2.6);
// Including the FuzzySet into FuzzyInput
difference->addFuzzySet(dif5);
// Including the FuzzyInput into Fuzzy
fuzzy->addFuzzyInput(difference);

// Instantiating a FuzzyOutput objects
FuzzyOutput *speed = new FuzzyOutput(1);
// Instantiating a FuzzySet object
FuzzySet *out1 = new FuzzySet(-1, 7, 7, 10);
// Including the FuzzySet into FuzzyOutput
speed->addFuzzySet(out1);
FuzzySet *out2 = new FuzzySet(10, 30, 30, 50.8);
// Including the FuzzySet into FuzzyOutput
speed->addFuzzySet(out2);
// Instantiating a FuzzySet object
```

```

FuzzySet *out3 = new FuzzySet(50, 65, 65, 80.8);
// Including the FuzzySet into FuzzyOutput
speed->addFuzzySet(out3);
FuzzySet *out4 = new FuzzySet(81, 110, 110, 135.8);
// Including the FuzzySet into FuzzyOutput
speed->addFuzzySet(out4);
// Instantiating a FuzzySet object
FuzzySet *out5 = new FuzzySet(136, 157, 157, 180);
// Including the FuzzySet into FuzzyOutput
speed->addFuzzySet(out5);
// Including the FuzzyOutput into Fuzzy
fuzzy->addFuzzyOutput(speed);

// Building
FuzzyRule//////////////////////////////////////
////////////////////////////////////// 1
FuzzyRuleAntecedent *dis1_dif1 = new
FuzzyRuleAntecedent();
dis1_dif1->joinWithAND(dis1, dif1);
FuzzyRuleConsequent *speed_out11 = new
FuzzyRuleConsequent();
speed_out11->addOutput(out1);
FuzzyRule *fuzzyRule1 = new FuzzyRule(1, dis1_dif1,
speed_out11);
fuzzy->addFuzzyRule(fuzzyRule1);

// Building
FuzzyRule//////////////////////////////////////
////////////////////////////////////// 1
FuzzyRuleAntecedent *dis1_dif2 = new
FuzzyRuleAntecedent();
dis1_dif2->joinWithAND(dis1, dif2);

```

```

    FuzzyRuleConsequent      *speed_out12      =      new
FuzzyRuleConsequent ();
    speed_out12->addOutput (out1);
    FuzzyRule *fuzzyRule2 = new FuzzyRule(2, dis1_dif2,
speed_out12);
    fuzzy->addFuzzyRule(fuzzyRule2);

    //                                                    Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis1_dif3      =      new
FuzzyRuleAntecedent ();
    dis1_dif3->joinWithAND(dis1, dif3);
    FuzzyRuleConsequent      *speed_out13     =      new
FuzzyRuleConsequent ();
    speed_out13->addOutput (out1);
    FuzzyRule *fuzzyRule3 = new FuzzyRule(3, dis1_dif3,
speed_out13);
    fuzzy->addFuzzyRule(fuzzyRule3);

    //                                                    Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis1_dif4      =      new
FuzzyRuleAntecedent ();
    dis1_dif4->joinWithAND(dis1, dif4);
    FuzzyRuleConsequent      *speed_out14     =      new
FuzzyRuleConsequent ();
    speed_out14->addOutput (out1);
    FuzzyRule *fuzzyRule4 = new FuzzyRule(4, dis1_dif4,
speed_out14);
    fuzzy->addFuzzyRule(fuzzyRule4);

```

```

//
// Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
FuzzyRuleAntecedent *dis1_dif5 = new
FuzzyRuleAntecedent ();
dis1_dif5->joinWithAND(dis1, dif5);
FuzzyRuleConsequent *speed_out35 = new
FuzzyRuleConsequent ();
speed_out35->addOutput(out3);
FuzzyRule *fuzzyRule5 = new FuzzyRule(5, dis1_dif5,
speed_out35);
fuzzy->addFuzzyRule(fuzzyRule5);

//
// Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
FuzzyRuleAntecedent *dis2_dif1 = new
FuzzyRuleAntecedent ();
dis2_dif1->joinWithAND(dis2, dif1);
FuzzyRuleConsequent *speed_out46 = new
FuzzyRuleConsequent ();
speed_out46->addOutput(out4);
FuzzyRule *fuzzyRule6 = new FuzzyRule(6, dis2_dif1,
speed_out46);
fuzzy->addFuzzyRule(fuzzyRule6);

//
// Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
FuzzyRuleAntecedent *dis2_dif2 = new
FuzzyRuleAntecedent ();

```



```

dis2_dif2->joinWithAND(dis2, dif2);
FuzzyRuleConsequent      *speed_out47      =      new
FuzzyRuleConsequent ();
speed_out47->addOutput (out4);
FuzzyRule *fuzzyRule7 = new FuzzyRule(7, dis2_dif2,
speed_out47);
fuzzy->addFuzzyRule(fuzzyRule7);

//                                                    Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
FuzzyRuleAntecedent      *dis2_dif3      =      new
FuzzyRuleAntecedent ();
dis2_dif3->joinWithAND(dis2, dif3);
FuzzyRuleConsequent      *speed_out28     =      new
FuzzyRuleConsequent ();
speed_out28->addOutput (out2);
FuzzyRule *fuzzyRule8 = new FuzzyRule(8, dis2_dif3,
speed_out28);
fuzzy->addFuzzyRule(fuzzyRule8);

//                                                    Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
FuzzyRuleAntecedent      *dis2_dif4      =      new
FuzzyRuleAntecedent ();
dis2_dif4->joinWithAND(dis2, dif4);
FuzzyRuleConsequent      *speed_out49     =      new
FuzzyRuleConsequent ();
speed_out49->addOutput (out4);
FuzzyRule *fuzzyRule9 = new FuzzyRule(9, dis2_dif4,
speed_out49);

```

```

fuzzy->addFuzzyRule(fuzzyRule9);

//
// Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
FuzzyRuleAntecedent *dis2_dif5 = new
FuzzyRuleAntecedent();
dis2_dif5->joinWithAND(dis2, dif5);
FuzzyRuleConsequent *speed_out510 = new
FuzzyRuleConsequent();
speed_out510->addOutput(out5);
FuzzyRule *fuzzyRule10 = new FuzzyRule(10, dis2_dif5,
speed_out510);
fuzzy->addFuzzyRule(fuzzyRule10);

//
// Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
FuzzyRuleAntecedent *dis3_dif1 = new
FuzzyRuleAntecedent();
dis3_dif1->joinWithAND(dis3, dif1);
FuzzyRuleConsequent *speed_out211 = new
FuzzyRuleConsequent();
speed_out211->addOutput(out2);
FuzzyRule *fuzzyRule11 = new FuzzyRule(1, dis3_dif1,
speed_out211);
fuzzy->addFuzzyRule(fuzzyRule11);

//
// Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
FuzzyRuleAntecedent *dis3_dif2 = new

```

```

FuzzyRuleAntecedent ();
    dis3_dif2->joinWithAND(dis3, dif2);
    FuzzyRuleConsequent      *speed_out212      =      new
FuzzyRuleConsequent ();
    speed_out212->addOutput(out2);
    FuzzyRule *fuzzyRule12 = new FuzzyRule(12, dis3_dif2,
speed_out212);
    fuzzy->addFuzzyRule(fuzzyRule12);

//                                                    Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis3_dif3      =      new
FuzzyRuleAntecedent ();
    dis3_dif3->joinWithAND(dis3, dif3);
    FuzzyRuleConsequent      *speed_out213      =      new
FuzzyRuleConsequent ();
    speed_out213->addOutput(out2);
    FuzzyRule *fuzzyRule13 = new FuzzyRule(13, dis3_dif3,
speed_out213);
    fuzzy->addFuzzyRule(fuzzyRule13);

//                                                    Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis3_dif4      =      new
FuzzyRuleAntecedent ();
    dis3_dif4->joinWithAND(dis3, dif4);
    FuzzyRuleConsequent      *speed_out214      =      new
FuzzyRuleConsequent ();
    speed_out214->addOutput(out2);
    FuzzyRule *fuzzyRule14 = new FuzzyRule(14, dis3_dif4,

```

```

speed_out214);
    fuzzy->addFuzzyRule(fuzzyRule14);

    //
    FuzzyRule//////////////////////////////////// Building
    //////////////////////////////////////
    ////////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis3_dif5      =      new
FuzzyRuleAntecedent ();
    dis3_dif5->joinWithAND(dis3, dif5);
    FuzzyRuleConsequent      *speed_out215    =      new
FuzzyRuleConsequent ();
    speed_out215->addOutput(out2);
    FuzzyRule *fuzzyRule15 = new FuzzyRule(15, dis3_dif5,
speed_out215);
    fuzzy->addFuzzyRule(fuzzyRule15);

    //
    FuzzyRule//////////////////////////////////// Building
    //////////////////////////////////////
    ////////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis4_dif1      =      new
FuzzyRuleAntecedent ();
    dis4_dif1->joinWithAND(dis4, dif1);
    FuzzyRuleConsequent      *speed_out216    =      new
FuzzyRuleConsequent ();
    speed_out216->addOutput(out2);
    FuzzyRule *fuzzyRule16 = new FuzzyRule(16, dis4_dif1,
speed_out216);
    fuzzy->addFuzzyRule(fuzzyRule16);

    //
    FuzzyRule//////////////////////////////////// Building
    //////////////////////////////////////
    ////////////////////////////////////// 1

```

```

    FuzzyRuleAntecedent      *dis4_dif2      =      new
FuzzyRuleAntecedent ();
    dis4_dif2->joinWithAND(dis4, dif2);
    FuzzyRuleConsequent      *speed_out217    =      new
FuzzyRuleConsequent ();
    speed_out217->addOutput(out2);
    FuzzyRule *fuzzyRule17 = new FuzzyRule(17, dis4_dif2,
speed_out217);
    fuzzy->addFuzzyRule(fuzzyRule17);

    //                                                    Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis4_dif3      =      new
FuzzyRuleAntecedent ();
    dis4_dif2->joinWithAND(dis4, dif3);
    FuzzyRuleConsequent      *speed_out218    =      new
FuzzyRuleConsequent ();
    speed_out218->addOutput(out2);
    FuzzyRule *fuzzyRule18 = new FuzzyRule(18, dis4_dif2,
speed_out218);
    fuzzy->addFuzzyRule(fuzzyRule18);

    //                                                    Building
FuzzyRule////////////////////////////////////
//////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis4_dif4      =      new
FuzzyRuleAntecedent ();
    dis4_dif4->joinWithAND(dis4, dif4);
    FuzzyRuleConsequent      *speed_out219    =      new
FuzzyRuleConsequent ();
    speed_out219->addOutput(out2);

```

```

    FuzzyRule *fuzzyRule19 = new FuzzyRule(19, dis4_dif4,
speed_out219);
    fuzzy->addFuzzyRule(fuzzyRule19);

    //
    FuzzyRule//////////////////////////////////// Building
////////////////////////////////////
//////////////////////////////////// 1
    FuzzyRuleAntecedent      *dis4_dif5      =      new
FuzzyRuleAntecedent ();
    dis4_dif5->joinWithAND(dis4, dif5);
    FuzzyRuleConsequent      *speed_out220    =      new
FuzzyRuleConsequent ();
    speed_out220->addOutput(out2);
    FuzzyRule *fuzzyRule20 = new FuzzyRule(20, dis4_dif5,
speed_out220);
    fuzzy->addFuzzyRule(fuzzyRule20);
}

void loop()
{
    if(digitalRead (pb)==1)
    {
        // Rumus RTD
        uint16_t rtd = max.readRTD();
        float ratio = rtd;
        ratio /= 32768;

        // Getting a random value
        //int input1 = random(0, 240);
        temp = max.temperature(RNOMINAL, RREF);
        int error;
        data2 = data1.toInt();
    }
}

```

```

// int data1;
error = data2-temp;
//int error = 0;
// Printing something
Serial.println("\n\n\nEntrance: ");
Serial.print("\t\t\tTemperature: ");
Serial.println(temp);
delay(2000);
Serial.println("");
datalama = max.temperature(RNOMINAL, RREF);
selisih = datalama-temp;
Serial.print("\t\t\tSelisih = ");
Serial.println(selisih);

// Set the random value as an input
//fuzzy->setInput(1, max1.temperature(RNOMINAL,
RREF));
fuzzy->setInput(1, error);
fuzzy->setInput(2, selisih);
// Running the Fuzzification
fuzzy->fuzzify();
// Running the Defuzzification
output = fuzzy->defuzzify(1);
// Printing something
Serial.println("Result: ");
Serial.print("\t\t\tSpeed: ");
Serial.println(output);
myservo.write(output);
// Serial.print("\t\t\tSet Point: ");
// Serial.println(data1);
// wait 12 seconds

```

```
    Arduino_SoftSerial.print(temp);
Arduino_SoftSerial.print("A");
    ///                Arduino_SoftSerial.print(output);
Arduino_SoftSerial.print("B");

Arduino_SoftSerial.print("\n");

while( Arduino_SoftSerial.available()>0)
{
    c =  Arduino_SoftSerial.read();

    if(c=='\n'){break;}
    else        {dataIn1+=c;}
}

if(c=='\n')
{
    Parse_the_Data();

    // Show All Data to Serial Monitor
    Serial.println("\t\t\tSet Point = " + data1);
//    Serial.println("\t\t\tDerajat = " + output);

Serial.println("=====");

    // Reset the variable
```



```
        c=0;
        dataIn1="";

    }

// HAsil di LCD
lcd.setCursor(0,0);
lcd.print("Monitoring");
lcd.setCursor(0,1);
lcd.print("Suhu= ");
lcd.setCursor(6,1);
lcd.print(temp);
lcd.setCursor(9,1);
lcd.print("Celcius");
lcd.setCursor(0,2);
lcd.print("Suhu Set=");
lcd.setCursor(10,2);
lcd.print(data1);
lcd.setCursor(13,2);
lcd.print("Celcius");
lcd.setCursor(0,3);
lcd.print("Derajat= ");
lcd.setCursor(9,3);
lcd.print(output);
lcd.setCursor(15,3);
lcd.print("'");
delay(1000);
lcd.clear();
}

if(digitalRead (pb)==0)
{
myservo.write(0);
```

```
lcd.setCursor(0,0);
lcd.print("STOPPP!!!");
delay(500);
lcd.clear();
}
}
////////////////////////////////////
////////////////////////////////////

void Parse_the_Data()
{
  indexOfC = dataIn1.indexOf("C");

  data1 = dataIn1.substring (0, indexOfC);
}
}
```



PROGRAM NodeMCU

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
//#include <Servo.h>
#include<SoftwareSerial.h>

SoftwareSerial NodeMcu_SoftSerial(D1,D2);

//Servo servo;
WidgetLCD lcd(V2);

int data1;

char auth[] = "Q8EsXGPJhvrTCZkpyCr16eFckcSlRqM7";
char ssid[] = "multiplayer1";
char pass[] = "ade1234567";

//below global variabel
char c;
String dataIn;
int8_t indexOfA, indexOfB;

String temp, output;
int pos = 0;

void setup() {
  // put your setup code here, to run once:

  // Open serial communication(Arduino-PC)
  Serial.begin(115200);
```

```
// Open serial communication(NodeMcu-PC)
NodeMcu_SoftSerial.begin(9600);

Blynk.begin(auth, ssid, pass);

// servo.attach(D8, 500, 2500); // NodeMCU D8 pin
}

void loop() {
  // put your main code here, to run repeatedly:
  Blynk.run();
  Blynk.virtualWrite(V0, temp);
  Blynk.virtualWrite(V1, output);
  Blynk.virtualWrite(V7, data1);

  NodeMcu_SoftSerial.print(data1);
  NodeMcu_SoftSerial.print("C");

  NodeMcu_SoftSerial.print("\n");

  while( NodeMcu_SoftSerial.available()>0)
  {
    c = NodeMcu_SoftSerial.read();

    if(c=='\n'){break;}
    else      {dataIn+=c;}
  }

  if(c=='\n')
```

```
{
    Parse_the_Data();

    // Show All Data to Serial Monitor
    Serial.println("Temperature = " + temp);
    Serial.println("Derajat = " + output);

Serial.println("=====");

    // Reset the variable
    c=0;
    dataIn="";
}

lcd.print(3, 0, "Monitoring");
lcd.print(1, 1, "Suhu & Derajat");
delay(1000);
}
////////////////////////////////////
////////////////////////////////////
BLYNK_WRITE (V3)

{

    data1 = param.asInt();
    Serial.println(data1);

}
//
BLYNK_WRITE (V4)
```

```
{
    data1 = param.asInt();
    Serial.println(data1);
}
//
BLYNK_WRITE (V5)

{
    data1 = param.asInt();
    Serial.println(data1);
}
//
BLYNK_WRITE (V6)

{
    data1 = param.asInt();
    Serial.println(data1);
}
//
BLYNK_WRITE (V8)

{
    data1 = param.asInt();
    Serial.println(data1);
}
//=====
=====

void Parse_the_Data()
{
    indexOfA = dataIn.indexOf("A");
    indexOfB = dataIn.indexOf("B");
```

```
temp = dataIn.substring (0, indexOfA);  
output = dataIn.substring (indexOfA+1, indexOfB);  
}
```

