

***SMART DEHYDRATOR BERBASIS INTERNET OF
THINGS MENGGUNAKAN METODE FUZZY LOGIC***

Tugas Akhir

diajukan untuk memenuhi salah satu syarat

memperoleh gelar sarjana

dari Program Studi Teknologi Informasi (Kampus Kota Surabaya)

Fakultas Informatika

Universitas Telkom

1202202055

Densa Tanzilda Widyawantara Dzahy



**Program Studi Sarjana Teknologi Informasi (Kampus Kota
Surabaya)**

Fakultas Informatika

Universitas Telkom

Surabaya

2024

LEMBAR PENGESAHAN

***SMART DEHYDRATOR BERBASIS INTERNET OF THINGS MENGGUNAKAN
METODE FUZZY LOGIC***

***SMART DEHYDRATOR BASED ON INTERNET OF THINGS USING FUZZY LOGIC
METHOD***

NIM : 1202202055

Densa Tanzilda Widyawantara Dzahy

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana Teknologi Informasi (Kampus Kota Surabaya)

Fakultas Informatika

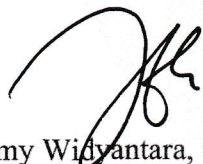
Universitas Telkom

Surabaya, 12 Agustus 2024

Menyetujui

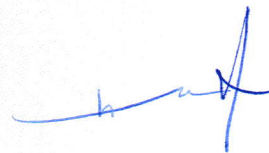
Pembimbing I,

Pembimbing II,



Dr. Helmy Widyantara, S.Kom., M.Eng.

NIP. 19790001



Muhammad Adib Kamali, S.T., M.Eng.

NIP. 22970007

Ketua Program Studi
Sarjana Teknologi Informasi,



Bernadus Anggo Seno Aji, S.Kom., M.Kom.

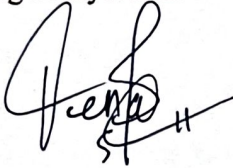
NIP. 23929009

LEMBAR PERNYATAAN

Dengan ini saya, Densa Tanzilda Widyawantara Dzahy, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul *Smart Dehydrator Berbasis Internet Of Things Menggunakan Metode Fuzzy Logic* beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya,

Surabaya, 12/08/2024

Yang Menyatakan



Densa Tanzilda Widyawantara Dzahy

Smart Dehydrator Berbasis Internet Of Things Menggunakan Metode Fuzzy Logic

Densa Tanzilda Widyawantara Dzahy¹, Helmy Widyantara², Muhammad Adib Kamali³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Surabaya

⁴Divisi Digital Service PT Telekomunikasi Indonesia

¹densatanzildawd@students.telkomuniversity.ac.id, ²helmywidyantara@telkomuniversity.ac.id,

³adibkamali@telkomuniversity.ac.id

Abstrak

Bunga telang memiliki berbagai macam khasiat bagi kesehatan dan sering sebagai olahan minuman sehat seperti teh bunga telang. Proses pengeringan bunga telang terdapat dua cara, yaitu menggunakan bantuan sinar matahari dan dehidrator. Proses pengeringan bunga telang yang dilakukan secara manual di bawah sinar matahari dapat memakan waktu kurang lebih selama 1-2 hari. Metode ini memiliki kekurangan yaitu bergantung pada sinar matahari, sehingga apabila sinar matahari terhalang karena cuaca buruk dapat mengakibatkan proses pengeringan menjadi terhambat. Selain itu, warna daun yang dihasilkan ungu tidak merata dan hitam yang dapat mempengaruhi rasa dan kualitas teh bunga telang. Berdasarkan permasalahan tersebut, maka penelitian ini bertujuan untuk merancang sistem dehidrator yang dapat memonitoring proses pengeringan bunga berbasis IoT menggunakan Fuzzy Logic. Melalui hasil pengujian laju kekeringan dan warna yang dilakukan menggunakan Fuzzy Logic, didapatkan bahwa suhu yang tinggi mendapatkan hasil laju pengeringan yang cepat serta suhu yang rendah mendapatkan hasil laju pengeringan yang lambat serta hasil akhir warna RGB paling rendah adalah pengeringan bunga dibawah matahari, sedangkan pengeringan menggunakan Smart Dehydrator paling rendah berada pada suhu 60°C 6 jam dan warna bunga yang paling tinggi terdapat pada suhu 40°C 4 jam. Sistem ini dapat mengurangi waktu pengeringan dari 1-2 hari menjadi 4-6 jam.

Kata kunci : Dehidrator, Bunga Telang, Pengeringan, IoT, Fuzzy Logic

Abstract

Butterfly pea flowers have various health benefits and are often processed into healthy drinks such as butterfly pea flower tea. There are two ways to dry butterfly pea flowers, namely using sunlight and a dehydrator. The manual drying process of butterfly pea flowers under the sun can take approximately 1-2 days. This method has the disadvantage of depending on sunlight, so if the sunlight is blocked due to bad weather, it can hamper the drying process. In addition, the color of the leaves produced is uneven purple and black which can affect the taste and quality of butterfly pea flower tea. Based on these problems, this study aims to design a dehydrator system that can monitor the IoT-based flower drying process using Fuzzy Logic. Through the results of the dryness rate and color tests carried out using Fuzzy Logic, it was found that high temperatures produce fast drying rates and low temperatures produce slow drying rates and the lowest RGB color results are flower drying under the sun, while drying using the Smart Dehydrator is lowest at a temperature of 60 °C 6 hours and the highest flower color is at a temperature of 40 °C 4 hours. This system can reduce drying time from 1-2 days to 4-6 hours.

Keywords: Dehydrator, Butterfly pea flowers, Drying, IoT, Fuzzy Logic

1. Pendahuluan Latar Belakang

Bunga telang merupakan tanaman yang sering dimanfaatkan dalam teh herbal karena memiliki beragam manfaat kesehatan, seperti membantu penurunan berat badan, mengatasi rambut rontok, dan mencegah kanker [1]. Proses pembuatan produk olahan teh bunga telang biasa dilakukan dengan melalui proses pengeringan, yang mana dengan cara mengeringkan bunga di bawah sinar matahari. Pengeringan dilakukan secara tradisional yaitu dijemur dibawah sinar matahari dengan kisaran waktu 1-2 hari [2]. Metode pengeringan dibawah sinar matahari memiliki kekurangan salah satunya yaitu ketika cuaca mendung maka proses pengeringan menjadi terhambat [3]. Beberapa faktor yang mempengaruhi kualitas teh bunga telang termasuk durasi dan suhu pengeringan. Hal ini terjadi karena suhu yang terlalu tinggi atau waktu pengeringan yang terlalu lama dapat menyebabkan perubahan pada sifat bahan yang dikeringkan, seperti kerusakan kandungan antioksidan dalam senyawa aktif yang terdapat pada bahan tersebut [4].

Pengeringan modern kini memanfaatkan peralatan seperti oven, rak pengering, atau *fresh dryer* dengan durasi sekitar 4-6 jam, dan suhunya dapat disesuaikan sesuai kebutuhan [5]. Penggunaan dehidrator dalam proses

pengeringan bunga telang memungkinkan untuk menghilangkan kadar air dari bunga tersebut tanpa mengakibatkannya menjadi gosong maupun mengurangi kandungan kualitas pada bunga telang [6]. Suhu yang terlalu tinggi dan waktu pengeringan yang lama dapat menyebabkan menurunnya kualitas bunga telang, serta menurunkan kandungan senyawa aktif dari bunga telang [7]. Pengeringan menggunakan dehidrator menjadi salah satu solusi untuk pengeringan bunga telang yang membutuhkan suhu yang tepat. Sistem pengeringan ini menggunakan teknologi dehidrasi yang dirancang khusus untuk memantau waktu dan suhu sehingga dapat memastikan bunga agar tidak terlalu matang. Dengan menggunakan dehidrator, memungkinkan untuk mengeringkan bunga dalam segala cuaca [8].

Berdasarkan latar belakang yang telah dibahas, perlu adanya sistem *monitoring* dan kendali otomatis dehidrator (*Smart Dehydrator*) untuk pengeringan bunga telang berbasis *Internet of Things* (IoT). *Smart Dehydrator* dilengkapi dengan IoT yang akan disambungkan melalui *website* agar dehidrator dapat melakukan kendali dan *monitoring* dari jarak jauh sehingga waktu produksi menjadi lebih efektif. Penggunaan dehidrator berbasis IoT pada pengeringan bunga akan membuat warna dan rasa bunga telang hampir sama dengan bunga asli serta kualitas akan tetap terjaga karena proses pengeringan bunga telang menggunakan dehidrator dapat mengatur suhu dan waktu pengeringan yang optimal [9]. Dehidrator berbasis IoT yang akan dibuat dapat memangkas waktu produksi dari pengeringan matahari yang dilakukan selama 1 - 2 hari menjadi hanya 4 - 6 jam.

Smart Dehydrator yang diusulkan ini menggunakan metode *Fuzzy Logic* serta digunakan untuk mengontrol kecepatan kipas dan *on/off heater* dalam dehidrator. Apabila suhu dan kelembaban sesuai *dengan rule fuzzy* maka kipas akan menyesuaikan mulai dari rendah, sedang hingga cepat. Sedangkan jika suhu melebihi *set point* maka *heater* mati, dan apabila suhu kurang dari *set point* maka *heater* akan menyala sesuai *rule*. Data tersebut akan dikirimkan ke server *website* untuk ditampilkan data berupa suhu, kelembaban, kecepatan kipas, dan *heater*.

Topik dan Batasannya

Permasalahan yang dirumuskan berdasarkan latar belakang yang telah dijelaskan sebelumnya yaitu kecepatan pengeringan yang kurang efisien, belum adanya sistem untuk pemantauan pengeringan secara berkala, serta pengaruh terhadap kualitas bunga yang dikeringkan. Batasan masalah pada penelitian ini adalah pengaturan panas pengeringan bunga telang yang berdasarkan pada suhu dan kelembaban, sistem *monitoring* suhu dan kelembaban berbasis *website*, pengeringan bunga telang berdasarkan berat dan warna bunga tidak termasuk menghitung kadar air.

Tujuan

Berdasarkan rumusan masalah yang telah disusun, penelitian ini bertujuan untuk mengefisiensi waktu karena mempercepat proses produksi, lebih mudah dalam mengontrol dan memantau dehidrator dalam proses pengeringan bunga telang dalam jarak jauh, dan menghasilkan produk bunga telang yang terjaga kualitas fisik dan warnanya.

2. Studi Terkait

2.1 Penelitian Terdahulu

Penelitian pada tugas akhir ini merupakan pengembangan metode penelitian sebelumnya dan menjadi referensi penulis dalam mengembangkan alat untuk menyelesaikan permasalahan yang telah dibahas. Penelitian pertama membuat sistem notifikasi penetas telur ayam kampung dengan teknologi IoT yang terhubung pada aplikasi Telegram dan menggunakan *Fuzzy Logic*. Komponen yang digunakan adalah Arduino Nano, DHT22, Sensor PIR, RTC, ESP32-CAM, *Motor Servo*, *Blower Fan*, lampu. Sistem ini menerima notifikasi ketika suhu terlalu dingin maupun terlalu panas yang dikirimkan melalui aplikasi telegram [10].

Penelitian kedua mengenai estimasi waktu terhadap pengeringan chip singkong. Alat yang digunakan pada pengeringan chips singkong adalah *cabinet dryer*. Pengeringan otomatis chips singkong menggunakan IR dan *Hot Air Dryer* lebih efektif (kurang dari 1 hari) dibandingkan pengeringan dengan sinar matahari (kurang dari 3 hari). Pengeringan otomatis menggunakan IR pada suhu 40°C terbukti lebih efisien, dengan waktu pengeringan yang 15% lebih cepat dibandingkan menggunakan *Hot Air Dryer* [11].

Penelitian ketiga mengenai sistem monitoring suhu sapi potong berbasis IoT menggunakan logika *Fuzzy*. Alat-alat yang digunakan meliputi sensor DHT22, MQ-135, dan modul ESP266. Penelitian ini mengimplementasikan IoT berbasis web dengan menerapkan metode logika *fuzzy* Mamdani untuk menentukan durasi penyalaan mini water pump. Pengujian sistem dilakukan dengan sensor MLX90614 untuk mendeteksi suhu tubuh sapi, sensor DHT11 untuk mengukur suhu dan kelembaban lingkungan, serta sensor HC-SR04 untuk mengukur ketinggian air [12].

2.2 Dasar Teori

2.2.1 Bunga Telang

Bunga telang (*Clitoria ternatea*) adalah salah satu tanaman yang tergolong dalam keluarga *Fabaceae* digunakan sebagai bahan pangan, obat tradisional, pakan, dll [13]. Pembuatan teh herbal bunga telang, diperlukan

sebuah proses pengeringan yang tepat. Suhu yang terlalu tinggi dan durasi pengeringan yang lama dapat mengurangi kualitas bunga telang serta menurunkan kandungan senyawa aktif di dalamnya [3].

2.2.2 Sensor DHT22

Penelitian ini menggunakan Sensor DHT22 karena digunakan sebagai pengukur suhu dan kelembapan pada dehidrator. DHT22 adalah sensor yang mengukur suhu dan kelembapan relatif dengan sinyal keluaran digital. Sensor ini memiliki 4 pin, yaitu *power supply*, *data signal*, *null*, dan *ground*. DHT22 menawarkan akurasi yang lebih tinggi dibandingkan DHT11, dengan kesalahan relatif pengukuran suhu sebesar 4% dan kelembapan sebesar 18% [14].

2.2.3 Logika Fuzzy

Logika *fuzzy* digunakan untuk menginterpretasikan nilai-nilai yang dinyatakan dalam bentuk bahasa (*linguistic*), seperti kecepatan kendaraan yang dapat diungkapkan dengan istilah seperti pelan, agak cepat, cepat, dan sangat cepat. Logika *Fuzzy* yang akan digunakan pada Tugas Akhir ini adalah *Fuzzy Mamdani*. Metode Mamdani memanfaatkan fungsi *MIN* dan fungsi *MAX* untuk mengombinasikan aturan-aturan dan menghasilkan himpunan *fuzzy* yang baru [15]. Cara kerja metode *Fuzzy Logic* dengan 3 tahap, yaitu:

1. Fuzzifikasi: suatu proses untuk mengubah masukan-masukan berbentuk tegas (*crisp*) menjadi bentuk fuzzy (variabel linguistik).
2. *Inference*: proses pengambilan keputusan ini melibatkan *input fuzzy* dan aturan *fuzzy* untuk menghasilkan *output fuzzy*. Dalam proses inferensi, fungsi implikasi yang digunakan untuk membentuk aturan adalah *Max-Min*. Secara umum rumus *max* adalah:

$$\mu_{sf}(xi) = \max (\mu_{sf}(xi), \mu_{kf}(xi)) \quad (2.1)$$

Dengan :

$\mu_{sf}(xi)$ = nilai keanggotaan solusi *fuzzy* pada aturan ke-i.

$\mu_{kf}(xi)$ = nilai keanggotaan konsekuensi *fuzzy* pada aturan ke-i

3. Defuzzifikasi: suatu proses mengubah *fuzzy output* dari hasil *inference* menjadi nilai tegas (*crisp*). Dalam proses defuzzifikasi, digunakan metode *Center Of Area* untuk menentukan nilai tegas output berdasarkan titik berat (z^*) dari kurva yang dihasilkan dari proses pengambilan keputusan. Secara umum dirumuskan :

$$z^* = \frac{\int_a^b \mu(x) \cdot x \cdot dx}{\int_a^b \mu(x) \cdot dx}, \text{ untuk variabel kontinyu} \quad (2.2)$$

$$z^* = \frac{\sum_{j=1}^n x_j \cdot \mu(x_j) \cdot x \cdot dx}{\sum_{j=1}^n \mu(x_j)}, \text{ untuk variabel diskrit} \quad (2.3)$$

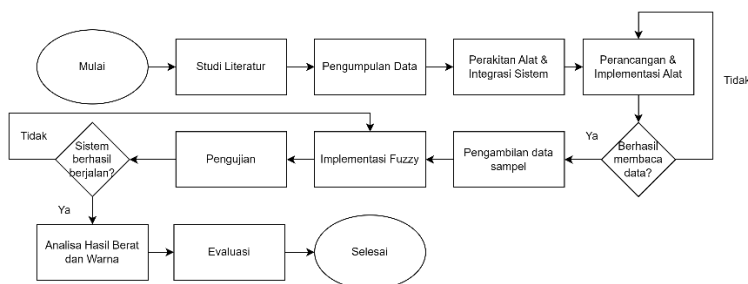
2.2.4 Firebase

Firestore adalah *platform* untuk aplikasi *real-time*. Ketika data mengalami perubahan, aplikasi yang terhubung dengan *Firestore* akan memperbarui informasi secara langsung di semua perangkat, baik itu *website* atau *mobile*. *Firestore Realtime Database* adalah *database online* yang berfungsi sebagai media penyimpanan data aplikasi. Data disimpan dalam format *JSON* dan dapat disinkronkan secara *real-time* dengan klien yang terhubung. Layanan ini memiliki tiga fitur utama: *Realtime*, *Offline*, dan Akses dari perangkat klien [16].

3. Sistem yang Dibangun

3.2.1 Prosedur Penelitian

Dalam melaksanakan penelitian ini, tahapan penelitian yang akan dilakukan ditunjukkan pada Gambar 1.1.

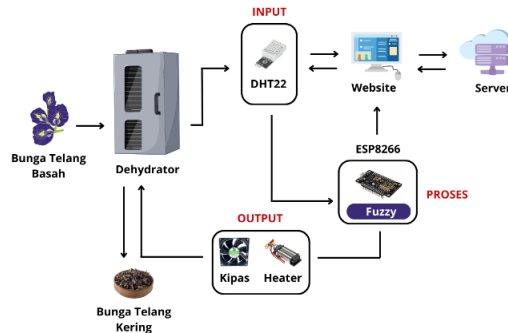


Gambar 1.1 Prosedur Penelitian *Smart Dehydrator*

Langkah pertama yang dilakukan pada penelitian ini adalah survei lapangan untuk mengamati produksi bunga telang ditempat kelompok tani yang membudidayakan bunga telang. Berdasarkan hasil pengamatan, diperoleh data sebagai acuan masalah dan solusinya. Setelah mengidentifikasi masalah, dilakukan studi literatur dengan membaca jurnal, *paper*, dan buku terkait solusi permasalahan untuk menentukan metode yang cocok untuk melakukan penelitian.

Setelah melakukan *review* terhadap literatur dan menentukan metode yang digunakan, pengumpulan data didapatkan dari hasil survei dan literatur. Perakitan alat dan integrasi sistem dilakukan serinci mungkin dan alur kerjanya agar jelas dan memudahkan dalam penyusunan alat. Setelah melakukan perakitan alat dan integrasi sistem, maka sistem diimplementasikan pada dehidrator lalu dilakukan pengujian dan implementasi sistem. Hasil pengujian dianalisa, kemudian dievaluasi dan dilakukan perbaikan apabila diperlukan. Langkah terakhir dari penelitian ini yaitu pembuatan jurnal untuk mendokumentasikan hasil analisis.

3.2.2 Diagram Alur

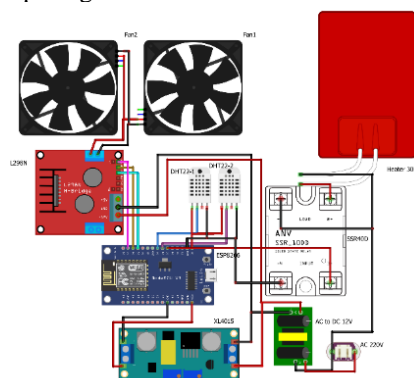


Gambar 3.2 Diagram Alur

Metode yang digunakan pada tugas akhir ini adalah metode *fuzzy logic mamdani*. *Fuzzy logic* digunakan untuk menentukan kecepatan kipas sehingga suhu didalam dehidrator dapat tercapai dan terjaga sesuai yang diinginkan. Berdasarkan alur diagram sistem pada Gambar 3.2, bunga telang basah dimasukkan ke dehidrator lalu sensor DHT22 akan mengambil data suhu dan kelembaban. Data suhu dan kelembaban dikirim ke mikrokontroler untuk diproses menggunakan *fuzzy*. Hasil *output fuzzy* adalah kecepatan kipas dan *heater* sesuai *rule* menggunakan PWM modul L298N sehingga suhu pada dehidrator stabil. Apabila waktu telah habis, maka otomatis sistem akan berhenti dan bunga telang dapat dikeluarkan.

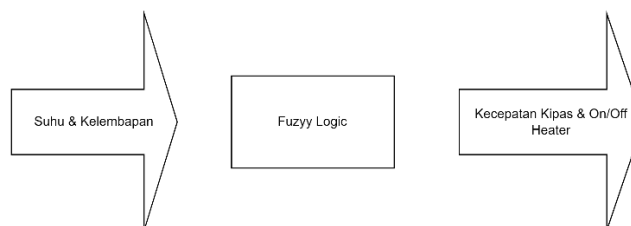
3.2.3 Perancangan Hardware

Merupakan tahapan awal dalam merakit *prototype* alat sebelum difinalisasi. Tujuan adanya *prototype* yaitu menghindari kegagalan penyatuan komponen dan meminimalisir kerusakan komponen elektronika. Rangkaian elektronika yang akan dibuat merujuk pada gambar 3.3.



Gambar 3.3 Rangkaian Elektronika

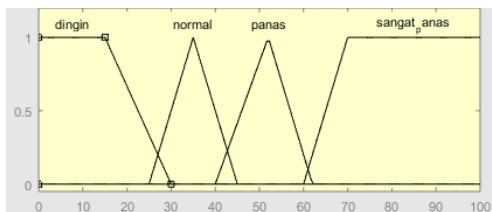
3.2.4 Implementasi Fuzzy Logic



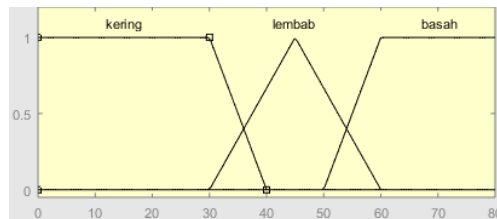
Gambar 3.3 Parameter *Input* dan *Output Fuzzy*

Berdasarkan blok diagram parameter *input* dan *output Fuzzy Mamdani* pada Gambar 3.3, sistem ini memiliki inputan suhu dan kelembapan yang didapatkan dari membaca Sensor DHT22. Sedangkan *output* dari sistem adalah kecepatan kipas dan on/off *heater* sesuai dengan perhitungan yang dilakukan.

3.2.4.1 Fungsi Keanggotaan Input Suhu dan Kelembaban



Gambar 3.4 Grafik *Input Suhu*



Gambar 3.5 Grafik *Input Kelembaban*

Pada gambar 3.4 dan 3.5 merupakan grafik fungsi keanggotaan *input* suhu serta kelembaban yang memiliki beberapa himpunan *fuzzy*, sedangkan *domain range input* suhu tercantum pada tabel 3.1 dan *domain range input* kelembaban tercantum pada tabel 3.2.

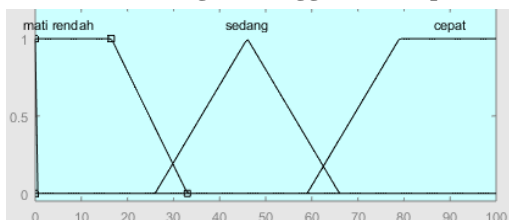
Tabel 3.1 *Domain Range Input Suhu*

No	Himpunan Fuzzy	Domain
1	Dingin	[0 – 30]
2	Normal	[25 – 45]
3	Panas	[40 – 62]
4	Sangat Panas	[60 – 100]

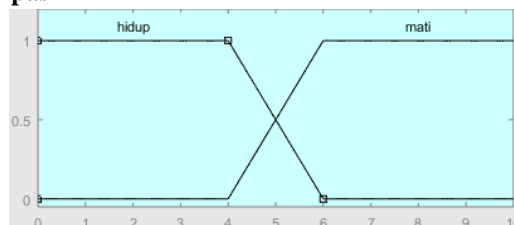
Tabel 3.2 *Domain Range Input Kelembaban*

No	Himpunan Fuzzy	Domain
1	Kering	[0 – 40]
2	Lembab	[30 – 60]
3	Basah	[50 – 100]

3.2.4.2 Fungsi Keanggotaan Output Kecepatan Kipas



Gambar 3.6 Grafik *Output Kecepatan Kipas*



Gambar 3.7 Grafik *Output Heater*

Pada gambar 3.6 dan 3.7 merupakan grafik fungsi keanggotaan *output* kecepatan kipas serta *heater* yang memiliki beberapa himpunan *fuzzy*, sedangkan *domain range output* kecepatan kipas tercantum pada tabel 3.3 dan *domain range output heater* tercantum pada tabel 3.4.

Tabel 3.3 *Domain Range Output Kecepatan Kipas*

No	Himpunan Fuzzy	Domain
1	Rendah	[0 – 33]
2	Sedang	[26 – 66]
3	Cepat	[59 – 100]

Tabel 3.4 *Domain Range Output Heater*

No	Himpunan Fuzzy	Domain
1	Hidup	[0 – 60]
2	Mati	[40 – 100]

3.2.4.3 Rulebase Fuzzy

Pembentukan *rule Fuzzy* digunakan untuk menyatakan relasi antara variabel *input* dan *output*. Terdapat dua variabel inputan yaitu suhu dan kelembaban, sedangkan *output* variabel terdapat dua yaitu kecepatan kipas dan *heater*. Total *rule fuzzy* yang terbentuk adalah 12 sebagaimana tercantum pada tabel 3.5.

Tabel 3.5 *Rulebase Fuzzy*

No	Input		Output	
	Suhu	Kelembaban	Kecepatan Kipas	Heater
1	Dingin	Kering	Rendah	Hidup
2	Dingin	Lembab	Sedang	Hidup
3	Dingin	Basah	Sedang	Hidup
4	Normal	Kering	Cepat	Hidup
5	Normal	Lembab	Sedang	Hidup
6	Normal	Basah	Sedang	Hidup
7	Panas	Kering	Cepat	Hidup
8	Panas	Lembab	Sedang	Hidup
9	Panas	Basah	Cepat	Hidup
10	Sangat Panas	Kering	Cepat	Mati
11	Sangat Panas	Lembab	Cepat	Mati
12	Sangat Panas	Basah	Cepat	Mati

4. Evaluasi

4.1 Hasil Pengujian

4.1.1 Hasil Perancangan Hardware

Hasil perancangan *hardware* merupakan hasil penggabungan komponen elektronika dari rencana yang telah dibuat. Di dalam ruang dehidrator terletak *Heater* dan sensor DHT22 yang terhubung melalui kabel *g* sebagaimana pada gambar 4.8 dan 4.9.



Gambar 4.8 Tampak Depan Hardware



Gambar 4.9 Tampak Dalam Hardware

Komponen lainnya berada di bawah ruang dehidrator, yaitu sensor ESP8266, *Power Supply*, *Driver Motor* L298N, *Step Down*, SSR, serta Kipas yang terletak dibawah *Heater*. Kipas diletakkan dibawah *Heater* agar penyebaran udara panas merata ke seluruh ruang dehidrator. Sensor DHT22 diletakkan menjadi dua bagian, yakni bagian atas *tray* pertama serta diantara *tray* pertama dan kedua. Peletakan dua bagian memiliki fungsi agar pengambilan data suhu dan kelembaban dapat lebih merata pada bagian atas dan bawah *tray*.

4.1.2 Pengujian Sensor DHT22

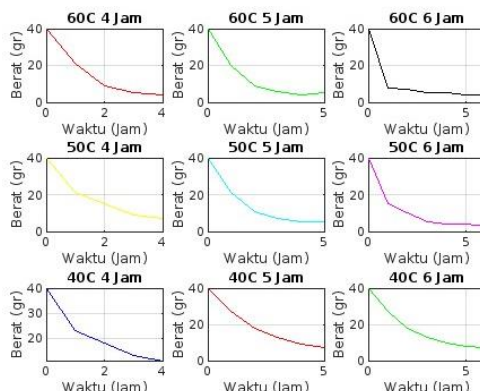
Pengujian sensor DHT22 bertujuan untuk mengevaluasi kemampuannya dalam mendeteksi perubahan suhu di ruang dehidrator. Pengujian dilakukan dengan membandingkan suhu yang diukur oleh termometer dengan data suhu yang diperoleh dari sensor DHT22.

Tabel 4.6 Pengujian Sensor DHT22

No	Smart Dehydrator (°C)	Termometer (°C)	Selisih	Error (%)
1	43.6	42.7	0.9	0,021
2	32.8	34.0	1.2	0,035
3	37.8	38.1	0.3	0,007
4	39.4	40.8	1.4	0,034
5	41.6	41.4	0.2	0,004
Rata – Rata Error				0,020

4.1.3 Pengujian Laju Kekeringan

Pengujian laju kekeringan bertujuan untuk mengetahui kemampuan pengeringan dehidrator sesuai dengan skema yang ada yakni pada suhu 60°C, 50°C, 40°C dengan tiga macam waktu pengujian. Perhitungan yang dilakukan adalah perubahan pada berat bunga yang ditimbang setiap satu jam. Perubahan laju pengeringan dapat dilihat melalui gambar 4.10.



Gambar 4.10 Grafik Laju Pengeringan

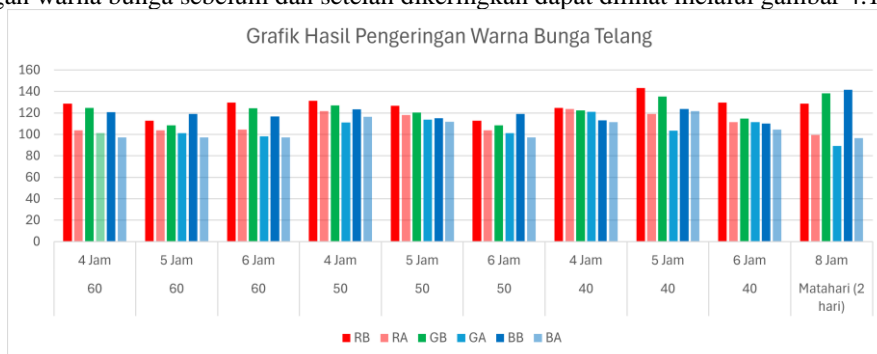
4.1.4 Pengujian Warna

Pada hasil pengujian warna dengan menggunakan pengolahan citra digital matlab Red (R), Green (G), Blue (B) dapat diketahui bahwa, warna bunga sebelum dikeringkan memiliki nilai RGB lebih tinggi dibandingkan dengan warna bunga yang sudah dikeringkan. Perhitungan warna sebelum dan sesudah dikeringkan menggunakan dehidrator dan matahari juga meliputi waktu yang sudah di tentukan seperti pada tabel 4.7.

Tabel 4.7 Pengujian Warna

No	Skema	Waktu	Hasil					
			RB	RA	GB	GA	BB	BA
1	60°C	4 Jam	128.5857	103.6791	124.6415	101.1925	120.6086	97.173
2		5 Jam	112.5799	103.6791	108.4158	101.1925	119.1511	97.173
3		6 Jam	129.7636	104.4918	124.3057	98.1438	116.8354	97.287
4	50°C	4 Jam	131.1483	121.6019	127.0993	111.0845	123.1539	116.2498
5		5 Jam	126.7674	118.1556	120.4232	113.7362	115.0916	111.5966
6		6 Jam	112.5799	103.6791	108.4158	101.1925	119.1511	97.173
7	40°C	4 Jam	124.6353	123.7693	122.3925	120.8844	113.1605	111.2948
8		5 Jam	143.0713	119.1255	135.1473	103.5549	123.7671	121.7399
9		6 Jam	129.5007	111.3785	114.5849	111.3702	110.0692	104.5021
10	Matahari (2 hari)	8 Jam	128.7175	99.5048	138.1199	89.229	141.5074	96.5519

Perbandingan warna bunga sebelum dan setelah dikeringkan dapat dilihat melalui gambar 4.11.



Gambar 4.11 Grafik Warna Before & After Warna Bunga

4.1.5 Pengujian Fuzzy Sistem dan Matlab

Pengujian kesesuaian *output* bertujuan untuk memeriksa sejauh mana *output* persentase kecepatan kipas dan heater sesuai dengan aturan *fuzzy* yang telah ditetapkan. Masing-masing *rule* diuji coba dengan tiga skema yakni skema 1 adalah 60°C, skema 2 adalah 50°C, skema 3 adalah 40°C. Tabel 4.8 menampilkan *output* presentase kecepatan kipas dan *heater* hasil perhitungan *fuzzy* pada *Arduino*.

Tabel 4.8 Pengujian Fuzzy Sistem Per-Skema

No	Skema	Waktu	Suhu	Kelembaban	Kecepatan Kipas	Matlab	Heater	Matlab	Rule Sesuai
1	60	4	60.10	14.85	Cepat	80.7	On	3.06	Sesuai
2		5	61.80	11.80	Cepat	80.7	Off	6.75	Sesuai
3	50	5	50.05	19.30	Cepat	75.7	On	2.51	Sesuai
4		6	51.75	19.00	Cepat	75.6	Off	6.30	Sesuai
5	40	4	40.25	28.20	Cepat	40.1	On	2.83	Sesuai
6		6	41.50	27.65	Cepat	65.7	Off	5.68	Sesuai

4.1.6 Pengujian Website

Setelah keseluruhan sistem berjalan, data akan dikirimkan ke *firebase* untuk disimpan dan ditampilkan melalui *website hosting*. Data yang disimpan pada *firebase* meliputi data *on/off smart dehydrator*, *heater*, sensor *dht22*, kipas, waktu. Tampilan *database firebase* dapat dilihat pada gambar 4.12.



Gambar 4.12 Tampilan Penyimpanan *Firestore*



Gambar 4.13 Tampilan *Website*

Website akan menampilkan data seperti pada gambar 4.13. Data yang ditampilkan antara lain suhu, kelembaban, kecepatan kipas, *on/off heater*. *Website monitoring* dapat diakses secara *online* melalui *domain*

<https://tugas-akhir-254b5.web.app>. Hasil pengujian *delay* pengiriman data melalui *firebase* ke *website* dilakukan menggunakan aplikasi *Postman* dengan rata – rata *delay* yang didapatkan dari 10 kali pengujian adalah 1368,1 ms. Dokumentasi serta tabel pengujian *delay* terdapat pada Lampiran 6.

4.2 Analisis Hasil Pengujian

Analisa dari hasil pengujian yang dilakukan diantaranya adalah analisa pengujian laju kekeringan dan pengujian warna. Pengujian laju kekeringan memiliki sembilan skema pengeringan diantaranya suhu 60°C (4 jam, 5 jam, 6 jam), suhu 50°C (4 jam, 5 jam, 6 jam), suhu 40°C (4 jam, 5 jam, 6 jam). Setiap skema pengeringan dilakukan dengan mengumpulkan data per-jam dan tercatat keseluruhan adalah 45 data. Tampilan grafik yang terdapat pada gambar 4.10 dapat di analisa bahwa yang pertama, pengeringan yang dilakukan dengan suhu 60°C dengan waktu 4 jam mengalami laju pengeringan yang cukup cepat, dikarenakan penurunan awal dimulai dari 21gr hingga 9gr serta hasil akhir yang didapatkan adalah 4gr. Kedua, pengeringan yang dilakukan dengan suhu 60°C dengan waktu 5 jam mengalami laju pengeringan yang cukup cepat, dikarenakan penurunan awal dimulai dari 20gr hingga 9gr serta hasil akhir yang didapatkan adalah 5gr. Ketiga, pengeringan yang dilakukan dengan suhu 60°C dengan waktu 6 jam mengalami laju pengeringan yang cukup cepat, dikarenakan penurunan awal dimulai dari 21gr hingga 7gr serta hasil akhir yang didapatkan adalah 4gr.

Selanjutnya pada pengujian keempat, pengeringan yang dilakukan dengan suhu 50°C dengan waktu 4 jam mengalami laju pengeringan yang cukup stabil, dikarenakan penurunan awal dimulai dari 21gr hingga 15gr serta hasil akhir yang didapatkan adalah 7gr. Kelima, pengeringan yang dilakukan dengan suhu 50°C dengan waktu 5 jam mengalami laju pengeringan yang cukup stabil, dikarenakan penurunan awal dimulai dari 21gr hingga 11gr serta hasil akhir yang didapatkan adalah 5gr. Keenam, pengeringan yang dilakukan dengan suhu 50°C dengan waktu 6 jam mengalami laju pengeringan yang cukup stabil, dikarenakan penurunan awal dimulai dari 20gr hingga 10gr serta hasil akhir yang didapatkan adalah 4gr.

Pada pengujian ketujuh pengeringan yang dilakukan dengan suhu 40°C dengan waktu 4 jam mengalami laju pengeringan yang cukup lambat, dikarenakan penurunan awal dimulai dari 23gr hingga 18gr serta hasil akhir yang didapatkan adalah 11gr. Kedelapam, pengeringan yang dilakukan dengan suhu 40°C dengan waktu 5 jam mengalami laju pengeringan yang cukup lambat, dikarenakan penurunan awal dimulai dari 27gr hingga 18gr serta hasil akhir yang didapatkan adalah 7gr. Kesembilan, pengeringan yang dilakukan dengan suhu 40°C dengan waktu 6 jam mengalami laju pengeringan yang cukup lambat, dikarenakan penurunan awal dimulai dari 27gr hingga 18gr serta hasil akhir yang didapatkan adalah 7gr.

Dari data pengujian diatas dapat dilihat bahwa suhu yang tinggi mendapatkan hasil laju pengeringan yang cepat dan kecil serta suhu yang rendah mendapatkan hasil laju pengeringan yang lambat dan besar. Selain pengaruh suhu, waktu juga berpengaruh terhadap hasil pengeringan, yakni waktu yang lebih lama akan mendapatkan hasil akhir pengeringan yang lebih kecil dibandingkan dengan waktu yang lebih cepat. Dengan begitu suhu dan waktu akan berkorelasi antara satu dan lain untuk menghasilkan hasil pengeringan yang diinginkan.

Adapun analisa pengujian warna didapatkan melalui membandingkan warna awal dan warna akhir bunga dari setiap skema pengujian. Dapat di analisa melalui perhitungan Red (R), Green (R), Blue (B) bahwa warna RGB awal bunga sebelum dikeringkan memiliki warna lebih tinggi (cerah) dibandingkan dengan warna akhir bunga setelah dikeringkan. Semakin rendah warna, maka hasil yang didapatkan adalah warna bunga yang tidak cerah dan gosong. Dari hasil pengujian tercatat bahwa hasil akhir warna RGB paling rendah ada pada pengeringan bunga dibawah matahari, yang menandakan hasil warna bunga lebih gelap, sedangkan pengeringan menggunakan *Smart Dehydrator* paling rendah berada pada suhu 60°C 6 Jam dan warna bunga yang paling tinggi (cerah) terdapat pada suhu 40°C 4 Jam. Hal ini sesuai dengan pendapat Meutia et al. (2019) bahwa peningkatan suhu dan durasi pemanasan dapat meningkatkan intensitas perubahan warna. Hal ini juga membuktikan bahwa hasil pengeringan menggunakan *Smart Dehydrator* terbukti dapat menjaga kualitas warna lebih baik dibandingkan dengan pengeringan dibawah sinar matahari.

5 Kesimpulan

Sistem *monitoring* dan kendali otomatis *Dehydrator (Smart Dehydrator)* untuk pengeringan bunga telang berbasis *Internet of Things (IoT)* menggunakan metode *fuzzy logic* berhasil dibuat dan diimplementasikan. Berdasarkan hasil pengujian dan pembahasan sistem yang dibuat berhasil melakukan proses pengeringan bunga telang lebih cepat 4 – 6 jam dibandingkan dengan pengeringan sinar matahari yakni 1 – 2 hari.

Pengguna dapat memantau data suhu, kelembaban, kecepatan kipas, *heater* secara *real-time* melalui *website*. Seluruh data hasil pengukuran sensor DHT22 maupun pengolahan *fuzzy* berhasil dikirim melalui *Wi-Fi* ke *firebase* dan ditampilkan pada *website*. *Website* yang digunakan juga sudah dapat di akses dengan internet.

Pengeringan bunga telang yang berdasarkan pengaturan *fuzzy logic* berhasil berjalan sesuai dengan *rule* sehingga dapat menghasilkan kualitas fisik dan warna bunga telang yang sesuai yang diinginkan.

Saran dari penelitian yang telah dilakukan untuk kedepannya adalah menghitung kandungan kadar air di dalam bunga telang baik sebelum maupun sesudah mengeringkan bunga serta menambah objek pengeringan dengan bunga maupun bahan lainnya.

Daftar Pustaka

- [1] R. Angelina and F. A. Syuhada, 2023, “Manfaat Bunga Telang Dan Pembudidayaan Di Cv. Faruq Farm (Benefits Of Telang Flower And Cultivation At Cv. Faruq Farm),” *J. Agriness*, vol. 1, no. 1, pp. 1–7.
- [2] D. A. Kunti Mulangsri, 2019, “93 Penyuluhan Pembuatan Bunga Telang Kering Sebagai Seduhan Teh Kepada Anak Panti Asuhan Yatim Putra Baiti Jannati,” *Abdimas Unwahas*, vol. 4, no. 2, pp. 2017–2020, doi: 10.31942/abd.v4i2.3010.
- [3] B. I. A. Muttaqin, 2014, “Pengembangan Alat Pengering Simplisia Jahe Menggunakan Sumber Panas Sinar Matahari dengan Backup Panas Kompor”. Available: <https://digilib.uns.ac.id/dokumen/detail/48831>
- [4] M. Syafrida, S. Darmanti, and M. Izzati, 2018, “Pengaruh Suhu Pengeringan Terhadap Kadar Air, Kadar Flavonoid dan Aktivitas Antioksidan Daun dan Umbi Rumput Teki (*Cyperus rotundus* L.),” *Bioma Berk. Ilm. Biol.*, vol. 20, no. 1, p. 44, doi: 10.14710/bioma.20.1.44-50.
- [5] R. A. Fauzi, A. Widyasanti, S. Dwiratna, N. Perwitasari, and S. Nurhasanah, 2022, “Optimasi Proses Pengeringan Terhadap Aktivitas Antioksidan Bunga Telang (*Clitoria Ternatea*) Menggunakan Metode Respon Permukaan Optimization of Drying Process on Antioxidant Activity of Butterfly Pea (*Clitoria ternatea*) by Using Response Surface Methodolo,” *J. Teknol. Pertan.*, vol. 23, no. 1, pp. 9–22.
- [6] M. P. Aulia, R. Rusmanto, W. Agustria, 2023, “Pengaruh Suhu Dan Lama Pengeringan Terhadap Karakteristik Serta Uji Kesukaan Teh Bunga Telang,” *Agrotech ...*, vol. 4, no. 1, pp. 1–5, doi: 10.36596/arj.v4i1.926.
- [7] N. K. Ayu Martini, N. G. Ayu Ekawati, and P. Timur Ina, 2020, “Pengaruh Suhu Dan Lama Pengeringan Terhadap Karakteristik Teh Bunga Telang (*Clitoria ternatea* L.),” *J. Ilmu dan Teknol. Pangan*, vol. 9, no. 3, p. 327, doi: 10.24843/itepa.2020.v09.i03.p09.
- [8] F. B. A. Tritama, Haslina, and D. Larasati, 2017, “Pengaruh Lama Waktu Pengeringan Dengan Food Dehidrator Terhadap Sifat Fisik Dan Kimia Bubuk Tongkol Jagung,” *J. Mhs. USM*, pp. 1–6.
- [9] P. D. Mentari, 2022, “Implementasi Fuzzy Logic Mamdani Pada Sistem Notifikasi Penetas Telur Ayam Kampung Dengan Teknologi Iot”.
- [10] P. D. Mentari, “Implementasi Fuzzy Logic Mamdani Pada Sistem Notifikasi Penetas Telur Ayam Kampung Dengan Teknologi Iot,” 2022.
- [11] A. Ma’ruf et al., “Efektifitas Pengeringan Chip Singkong Menggunakan Infrared dan Hot Air Dryer dalam Pembuatan Modified Cassava Flour (MOCAF),” *JRST (Jurnal Ris. Sains dan Teknol.*, vol. 7, no. 1, p. 77, 2023, doi: 10.30595/jrst.v7i1.16133.
- [12] A. Tresna Utama, A. Panji Sasmito, and A. Faisol, “Implementasi Logika Fuzzy Pada Sistem Monitoring Online Suhu Sapi Potong Berbasis Iot,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 5, no. 1, pp. 16–24, 2021, doi: 10.36040/jati.v5i1.3226.
- [15] N. Asiah, U. Bakrie, M. Djaeni, and U. Diponegoro, [Date] [Document Title], no. September. 2021.
- [16] E. A. W. Sanad, “Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire,” *J. Penelit. Enj.*, vol. 22, no. 1, pp. 20–26, 2019, doi: 10.25042/jpe.052018.04.

Lampiran

Lampiran 1. Tampilan Bunga Telang Dikeringkan dengan *Smart Dehydrator* & Sinar Matahari



Gambar 14. Sebelum bunga dikeringkan matahari



Gambar 15. Setelah bunga dikeringkan matahari



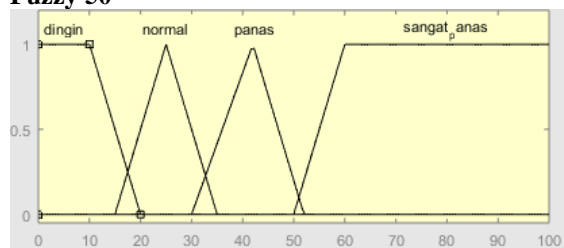
Gambar 16. Sebelum bunga dikeringkan SMATOR



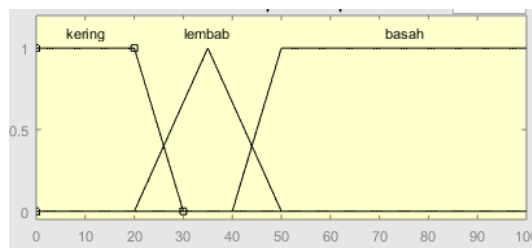
Gambar 17. Setelah bunga dikeringkan SMATOR

Lampiran 2. Fungsi Keanggotaan *Input* dan *Output Fuzzy*

Fuzzy 50



Gambar 18. Fungsi Input Suhu 50



Gambar 19. Fungsi Input Kelembaban 50

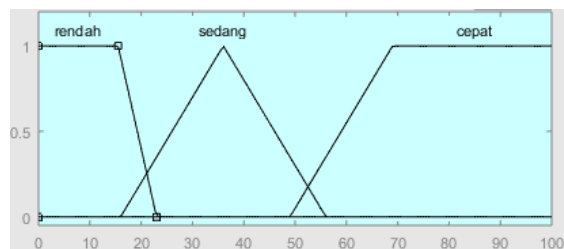
Pada gambar 15 dan 16 merupakan grafik fungsi keanggotaan *input* suhu serta kelembaban pada skema 50°C yang memiliki beberapa himpunan *fuzzy*, sedangkan *domain range input* suhu tercantum pada tabel 9 dan *domain range input* kelembaban tercantum pada tabel 10.

Tabel 9. *Domain Range Input* Suhu 50

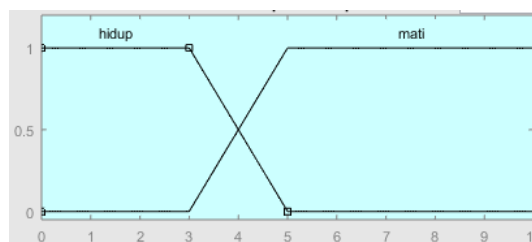
No	Himpunan Fuzzy	Domain
1	Dingin	[0 – 20]
2	Normal	[15 – 35]
3	Panas	[30 – 52]
4	Sangat Panas	[50 – 100]

Tabel 10. *Domain Range Input* Kelembaban 50

No	Himpunan Fuzzy	Domain
1	Kering	[0 – 30]
2	Lembab	[20 – 50]
3	Basah	[40 – 100]



Gambar 20. Fungsi Output Kecepatan Kipas 50



Gambar 21. Fungsi Output Heater 50

Pada gambar 17 dan 18 merupakan grafik fungsi keanggotaan *output* kecepatan kipas serta *heater* pada skema 50°C yang memiliki beberapa himpunan *fuzzy*, sedangkan *domain range output* kecepatan kipas tercantum pada tabel 11 dan *domain range output heater* tercantum pada tabel 12.

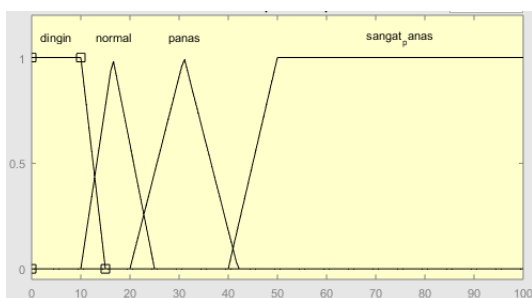
Tabel 11. Domain Range Output Kipas 50

No	Himpunan Fuzzy	Domain
1	Rendah	[0 – 23]
2	Sedang	[16 – 56]
3	Cepat	[49 – 100]

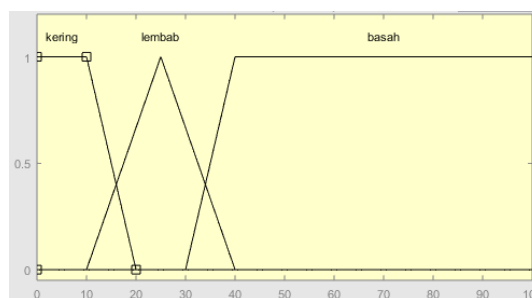
Tabel 12. Domain Range Output Heater 50

No	Himpunan Fuzzy	Domain
1	Hidup	[0 – 5]
2	Mati	[3 – 10]

Fuzzy 40



Gambar 22. Fungsi Input Suhu 40



Gambar 23. Fungsi Input Kelembaban 40

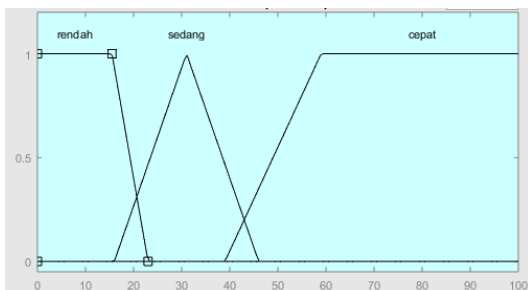
Pada gambar 17 dan 18 merupakan grafik fungsi keanggotaan *input* suhu serta kelembaban pada skema 40°C yang memiliki beberapa himpunan *fuzzy*, sedangkan *domain range input* suhu tercantum pada tabel 13 dan *domain range input* kelembaban tercantum pada tabel 14.

Tabel 13. Domain Range Input Suhu

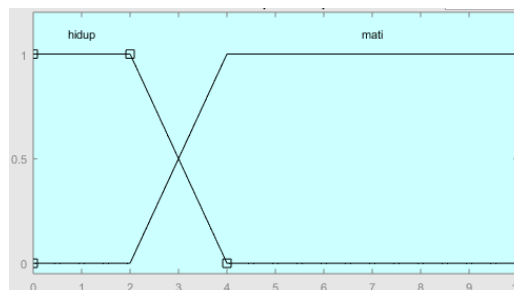
No	Himpunan Fuzzy	Domain
1	Dingin	[0 – 15]
2	Normal	[10 – 25]
3	Panas	[20 – 42]
4	Sangat Panas	[40 – 100]

Tabel 14. Domain Range Input Kelembaban

No	Himpunan Fuzzy	Domain
1	Kering	[0 – 20]
2	Lembab	[10 – 40]
3	Basah	[30 – 100]



Gambar 24. Fungsi Output Kipas 40



Gambar 25. Fungsi Output Heater 40

Pada gambar 15 dan 16 merupakan grafik fungsi keanggotaan *input* suhu serta kelembaban yang memiliki beberapa himpunan *fuzzy*, sedangkan *domain range output* kecepatan kipas tercantum pada tabel 15 dan *domain range output heater* tercantum pada tabel 16.

Tabel 15. Domain Range Output Kipas

No	Himpunan Fuzzy	Domain
1	Rendah	[0 – 23]
2	Sedang	[16 – 46]
3	Cepat	[39 – 100]

Tabel 16. Domain Range Output Heater

No	Himpunan Fuzzy	Domain
1	Hidup	[0 – 4]
2	Mati	[2 – 10]

Lampiran 3. Alat dan Bahan Penelitian

Alat yang digunakan pada penelitian ini dibagi menjadi hardware dan software. Hardware penelitian tercantum pada Tabel 9 dan software penelitian tercantum pada Tabel 17.

Tabel 17. Hardware yang Digunakan

No	Uraian	Keterangan

1	NodeMCU ESP8266	Mikrokontroler yang berfungsi sebagai pengendalian Sensor DHT22
2	Sensor DHT22	Pengukur suhu dan kelembapan pada Dehydrator
3	Heater 300W	Pemanas yang menggunakan listrik sebagai input untuk menghasilkan listrik
4	Kipas 12 Volt 5x5	Aktuator yang berguna untuk memasukkan udara di dalam dehydrator sehingga kelembapan tetap terjaga
5	Power Supply 12 Volt	Penyedia daya untuk satu atau lebih beban listrik
6	Driver Motor L289n	Mengontrol kecepatan serta arah perputaran motor DC
7	Relay AC SSR D40	Untuk mengatur On/Off Heater
8	XL4015 Step Down	Mengubah tegangan tinggi dari power supply menjadi tegangan rendah ke ESP8266

Tabel 18. Software yang Digunakan

No	Uraian	Keterangan
1	Arduino IDE	Interface pemrograman arduino yang diprogram langsung ke ESP8266
2	Database Firebase	Penyimpanan data yang dikirimkan oleh server disimpan menggunakan database firebase
3	Visual Studio Code	Editor untuk membuat program dan perancangan website
4	Matlab	Berfungsi untuk menguji coba alat dan analisis data

Lampiran 4. Tabel Pengujian Laju Kekeringan

Tabel 19. Laju Pengeringan

No	Suhu (°C)	Waktu	Berat (gr)
1	60°C	1	21
2		2	9
3		3	5
4		4	4
5		1	20
6		2	9
7		3	6
8		4	4
9		5	5
10		1	21
11		2	7
12		3	5
13		4	5
14		5	4

15		6	4
16	50°C	1	21
17		2	15
18		3	9
19		4	7
20		1	21
21		2	11
22		3	7
23		4	5
24		5	5
25		1	15
26		2	10
27		3	5
28		4	4
29		5	4
30	6	4	
31	40°C	1	23
32		2	18
33		3	13
34		4	11
35		1	27
36		2	18
37		3	13
38		4	9
39		5	7
40		1	27
41		2	18
42		3	13
43		4	10
44		5	8
45		6	7

Lampiran 5. Pengujian Fuzzy Sistem dan Matlab

Tabel 20. Pengujian Output 60

No	Skema	Jam Ke -	Suhu	Kelembaban	Kecepatan Kipas	Matlab	Heater	Matlab	Rule Sesuai
1	4	1	61.30	14.60	Cepat	80.4	On	5.70	Sesuai
2		2	60.10	14.85	Cepat	80.7	On	3.06	Sesuai
3		3	61.80	11.80	Cepat	80.7	Off	6.75	Sesuai
4		4	62.00	15.25	Cepat	80.7	Off	7.12	Sesuai
5	5	1	58.00	18.90	Cepat	81.7	On	2.78	Sesuai
6		2	59.05	13.80	Cepat	81.2	On	2.83	Sesuai
7		3	61.80	11.55	Cepat	80.7	Off	6.75	Sesuai
8		4	61.50	11.55	Cepat	80.5	Off	6.14	Sesuai
9		5	60.55	12.00	Cepat	80.5	On	3.86	Sesuai
10	6	1	60.10	15.15	Cepat	80.7	On	3.06	Sesuai
11		2	61.70	12.35	Cepat	80.6	Off	6.55	Sesuai
12		3	59.70	12.50	Cepat	80.9	On	2.86	Sesuai
13		4	60.60	11.85	Cepat	80.5	On	4.07	Sesuai
14		5	61.80	10.60	Cepat	80.7	Off	6.75	Sesuai
15		6	61.70	10.10	Cepat	80.6	Off	6.55	Sesuai

Tabel 21. Pengujian Output 50

No	Skema	Jam Ke -	Suhu	Kelembaban	Kecepatan Kipas	Matlab	Heater	Matlab	Rule Sesuai
1	4	1	51.35	22.55	Cepat	67.8	Off	5.94	Sesuai
2		2	51.95	17.90	Cepat	75.7	Off	6.56	Sesuai

3		3	52.25	16.35	Cepat	75.9	Off	6.64	Sesuai
4		4	52.80	15.45	Cepat	76.1	Off	6.66	Sesuai
5	5	1	51.70	21.55	Cepat	71.1	Off	6.23	Sesuai
6		2	51.85	18.35	Cepat	75.7	Off	6.44	Sesuai
7		3	50.05	19.30	Cepat	75.7	On	2.51	Sesuai
8		4	48.60	18.70	Cepat	76.4	On	2.31	Sesuai
9		5	50.35	17.65	Cepat	75.6	On	3.32	Sesuai
10		6	1	51.75	19.00	Cepat	75.6	Off	6.30
11	2		49.15	19.15	Cepat	76.2	On	2.33	Sesuai
12	3		52.00	16.45	Cepat	75.7	Off	6.62	Sesuai
13	4		50.30	16.35	Cepat	75.6	On	3.19	Sesuai
14	5		50.45	19.90	Cepat	75.5	On	3.58	Sesuai
15	6		53.40	19.15	Cepat	76.4	Off	6.69	Sesuai

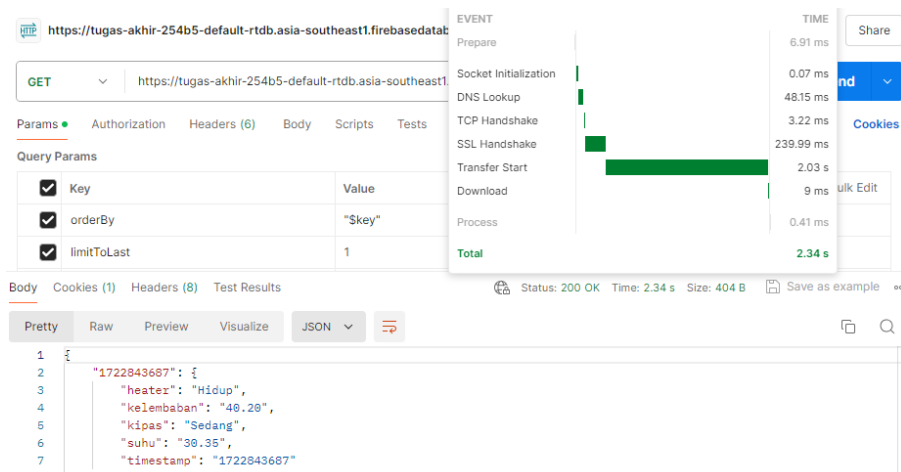
Tabel 22. Pengujian Output 40

No	Skema	Jam Ke -	Suhu	Kelembaban	Kecepatan Kipas	Matlab	Heater	Matlab	Rule Sesuai
1	4	1	43.40	33.50	Cepat	71.4	Off	6.19	Sesuai
2		2	42.30	30.15	Cepat	70.9	Off	6.14	Sesuai
3		3	40.25	28.20	Cepat	70.1	On	2.83	Sesuai
4		4	41.95	26.10	Cepat	70.3	Off	6.09	Sesuai
5	5	1	43.40	33.45	Cepat	71.4	Off	6.19	Sesuai
6		2	42.20	32.30	Cepat	70.8	Off	6.14	Sesuai
7		3	43.75	23.70	Cepat	71.6	Off	6.21	Sesuai
8		4	40.40	27.90	Cepat	74.8	On	3.33	Sesuai
9		5	39.95	29.25	Cepat	71.0	On	1.88	Sesuai
10	6	1	43.80	31.15	Cepat	71.6	Off	6.21	Sesuai
11		2	43.95	25.75	Cepat	71.7	Off	6.22	Sesuai
12		3	41.50	27.65	Cepat	66.1	Off	5.72	Sesuai
13		4	39.75	30.20	Cepat	68.0	On	1.87	Sesuai
14		5	42.35	26.30	Cepat	70.9	Off	6.14	Sesuai
15		6	41.35	27.45	Cepat	64.3	Off	5.56	Sesuai

Lampiran 6. Pengujian Delay Pengiriman Data *Firestore* ke *Website*

Tabel 23. Pengujian Delay Pengiriman Data

Pengujian Ke -	Delay
1	1633 ms
2	912 ms
3	1845 ms
4	917 ms
5	1658 ms
6	978 ms
7	1581 ms
8	1635 ms
9	872 ms
10	1650 ms
Rata - Rata	1368,1 ms



Gambar 26. Dokumentasi Pengujian Delay Pengiriman Data Menggunakan Postman

Lampiran 7. Code Program Arduino

Fuzzy 60°C

```
#include <Fuzzy.h>
#include "DHT.h"
#define DHT_PIN D5 // The ESP8266 pin D5 connected to DHT22 sensor
#define DHT_PIN1 D6 // The ESP8266 pin D6 connected to DHT22 sensor
#define DHT_TYPE DHT22
#define pinSSR 13

// pin L298n
int pinA = D2;
int pinB = D3;
int enableAB = D1;

// package dht22
DHT dht22(DHT_PIN, DHT_TYPE);
DHT dht221(DHT_PIN1, DHT_TYPE);

Fuzzy *fuzzy = new Fuzzy(); //deklarasi fuzzy sebuah variabel, bisa diubah sesuai keinginan

//Fuzzy input suhu
FuzzySet *dingin = new FuzzySet(0, 0, 15, 30);
FuzzySet *normal = new FuzzySet(25, 35, 35, 45);
FuzzySet *panas = new FuzzySet(40, 52, 52, 62);
FuzzySet *sangat_panas = new FuzzySet(60, 70, 100, 100);

//Fuzzy input kelembaban
FuzzySet *kering = new FuzzySet(0, 0, 30, 40);
FuzzySet *lembab = new FuzzySet(30, 45, 45, 60);
FuzzySet *basah = new FuzzySet(50, 60, 100, 100);

//Fuzzy output kipas
// FuzzySet *mati = new FuzzySet(0, 0, 0, 0);
FuzzySet *rendah = new FuzzySet(0, 0, 16.5, 33);
FuzzySet *sedang = new FuzzySet(26, 46, 46, 66);
FuzzySet *cepat = new FuzzySet(59, 79, 100, 100);

//Fuzzy output heater
FuzzySet *hidup = new FuzzySet(0, 0, 40, 60);
FuzzySet *mati = new FuzzySet(40, 60, 100, 100);

// fungsi untuk membuat fuzzifikasi
```

```
void declareFuzzy()
{
    // set FuzzyInput suhu
    FuzzyInput *suhu = new FuzzyInput(1);
    suhu->addFuzzySet(dingin);
    suhu->addFuzzySet(normal);
    suhu->addFuzzySet(panas);
    suhu->addFuzzySet(sangat_panas);
    fuzzy->addFuzzyInput(suhu);

    // set FuzzyInput kelembaban
    FuzzyInput *kelembaban = new FuzzyInput(2);
    kelembaban->addFuzzySet(kering);
    kelembaban->addFuzzySet(lembab);
    kelembaban->addFuzzySet(basah);
    fuzzy->addFuzzyInput(kelembaban);

    // set fuzzyOutput kipas
    FuzzyOutput *kipas = new FuzzyOutput(1);
    kipas->addFuzzySet(rendah);
    kipas->addFuzzySet(sedang);
    kipas->addFuzzySet(cepat);
    fuzzy->addFuzzyOutput(kipas);

    // set fuzzyOutput heater
    FuzzyOutput *heater = new FuzzyOutput(2);
    heater->addFuzzySet(hidup);
    heater->addFuzzySet(mati);
    fuzzy->addFuzzyOutput(heater);

    //Building fuzzy rule
    //SuhuDingin_KelembabanKering_KipasRendah
    FuzzyRuleAntecedent *ifDinginAndKering = new FuzzyRuleAntecedent(); //fuzzy rule input
    ifDinginAndKering->joinWithAND(dingin, kering);

    FuzzyRuleConsequent *thenKipasRendahHeaterHidup = new FuzzyRuleConsequent(); //fuzzy rule output
    thenKipasRendahHeaterHidup->addOutput(rendah);
    thenKipasRendahHeaterHidup->addOutput(hidup);

    FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifDinginAndKering, thenKipasRendahHeaterHidup);
    fuzzy->addFuzzyRule(fuzzyRule01);

    //SuhuDingin_KelembabanLembab_KipasCepat
    FuzzyRuleAntecedent *ifDinginAndLembab = new FuzzyRuleAntecedent();
    ifDinginAndLembab->joinWithAND(dingin, lembab);

    FuzzyRuleConsequent *thenKipasSedangHeaterHidup = new FuzzyRuleConsequent();
    thenKipasSedangHeaterHidup->addOutput(sedang);
    thenKipasSedangHeaterHidup->addOutput(hidup);

    FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifDinginAndLembab, thenKipasSedangHeaterHidup);
    fuzzy->addFuzzyRule(fuzzyRule02);

    //SuhuDingin_KelembabanBasah_KipasSedang
    FuzzyRuleAntecedent *ifDinginAndBasah = new FuzzyRuleAntecedent();
    ifDinginAndBasah->joinWithAND(dingin, basah);

    FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifDinginAndBasah, thenKipasSedangHeaterHidup);
    fuzzy->addFuzzyRule(fuzzyRule03);
}
```

```
//SuhuNormal_KelembabanKering_KipasRendah
FuzzyRuleAntecedent *ifNormalAndKering = new FuzzyRuleAntecedent();
ifNormalAndKering->joinWithAND(normal, kering);

FuzzyRuleConsequent *thenKipasCepatHeaterHidup = new FuzzyRuleConsequent();
thenKipasCepatHeaterHidup->addOutput(cepat);
thenKipasCepatHeaterHidup->addOutput(hidup);

FuzzyRule *fuzzyRule04 = new FuzzyRule(4, ifNormalAndKering, thenKipasCepatHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule04);

//SuhuNormal_KelembabanLembab_KipasSedang
FuzzyRuleAntecedent *ifNormalAndLembab = new FuzzyRuleAntecedent();
ifNormalAndLembab->joinWithAND(normal, lembab);

FuzzyRule *fuzzyRule05 = new FuzzyRule(5, ifNormalAndLembab, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule05);

//SuhuNormal_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifNormalAndBasah = new FuzzyRuleAntecedent();
ifNormalAndBasah->joinWithAND(normal, basah);

FuzzyRule *fuzzyRule06 = new FuzzyRule(6, ifNormalAndBasah, thenKipasRendahHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule06);

//SuhuPanas_KelembabanKering_KipasMati
FuzzyRuleAntecedent *ifPanasAndKering = new FuzzyRuleAntecedent();
ifPanasAndKering->joinWithAND(panas, kering);

FuzzyRule *fuzzyRule07 = new FuzzyRule(7, ifPanasAndKering, thenKipasCepatHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule07);

//SuhuPanas_KelembabanLembab_KipasSedang
FuzzyRuleAntecedent *ifPanasAndLembab = new FuzzyRuleAntecedent();
ifPanasAndLembab->joinWithAND(panas, lembab);

FuzzyRule *fuzzyRule08 = new FuzzyRule(8, ifPanasAndLembab, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule08);

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifPanasAndBasah = new FuzzyRuleAntecedent();
ifPanasAndBasah->joinWithAND(panas, basah);

FuzzyRule *fuzzyRule09 = new FuzzyRule(9, ifPanasAndBasah, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule09);

//SuhuSangatPanas_KelembabanKering_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndKering = new FuzzyRuleAntecedent();
ifSangatPanasAndKering->joinWithAND(sangat_panas, kering);

FuzzyRuleConsequent *thenKipasCepatHeaterMati = new FuzzyRuleConsequent();
thenKipasCepatHeaterMati->addOutput(cepat);
thenKipasCepatHeaterMati->addOutput(mati);

FuzzyRule *fuzzyRule10 = new FuzzyRule(10, ifSangatPanasAndKering, thenKipasCepatHeaterMati);
fuzzy->addFuzzyRule(fuzzyRule10);

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndLembab = new FuzzyRuleAntecedent();
```

```

ifSangatPanasAndLembab->joinWithAND(sangat_panas, lembab);

FuzzyRule *fuzzyRule11 = new FuzzyRule(11, ifSangatPanasAndLembab, thenKipasCepatHeaterMati);
fuzzy->addFuzzyRule(fuzzyRule11);

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndBasah = new FuzzyRuleAntecedent();
ifSangatPanasAndBasah->joinWithAND(sangat_panas, basah);

FuzzyRule *fuzzyRule12 = new FuzzyRule(12, ifSangatPanasAndBasah, thenKipasCepatHeaterMati);
fuzzy->addFuzzyRule(fuzzyRule12);
}

void setup ()
{
  Serial.begin(9600); //setting serial board ke monitor

  //setting pin menjadi OUTPUT
  pinMode(pinA, OUTPUT);
  pinMode(pinB, OUTPUT);
  pinMode(enableAB, OUTPUT);
  pinMode(pinSSR, OUTPUT);

  //setting pin SSR menjadi nyala/HIGH (mengalirkan listrik)
  digitalWrite(pinSSR, HIGH);

  //dht22
  dht22.begin(); // initialize the DHT sensor
  dht221.begin(); // initialize the DHT sensor

  // pemanggilan fungsi declareFuzzy agar dapat digunakan
  declareFuzzy();
}

void loop()
{
  // dht22
  // baca kelembaban
  float humi = dht22.readHumidity(); //pake ini
  // float humi = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor
  // baca suhu
  float temperature_C = dht22.readTemperature(); //pake ini
  // float temperature_C = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor

  // baca kelembaban
  float humi1 = dht221.readHumidity(); //pake ini
  // float humi1 = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor
  // baca suhu
  float temperature_C1 = dht221.readTemperature(); //pake ini
  // float temperature_C1 = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor

  // mencari nilai rata2 dari kedua sensor
  float uhu = (temperature_C + temperature_C1) / 2;
  float bab = (humi + humi1) / 2;

  Serial.println("\n===== INPUT =====");
  // cek apakah sensor berfungsi dengan benar
  if ( !isnan(temperature_C && temperature_C1) || !isnan(humi && humi1)) {
    Serial.println("Gagal membaca sensor!");
  }
}

```

```

}
// cetak nilai sensor
else {
  Serial.print("Kelembaban Atas: ");
  Serial.print(humi);
  Serial.print("%");
  Serial.print(" | ");

  Serial.print("Suhu Atas: ");
  Serial.print(temperature_C);
  Serial.println("°C");

  Serial.print("Kelembaban Bawah: ");
  Serial.print(humi1);
  Serial.print("%");

  Serial.print(" | ");

  Serial.print("Suhu Bawah: ");
  Serial.print(temperature_C1);
  Serial.println("°C");
}

Serial.print("Rata-Rata Suhu : ");
Serial.println(uhu);
Serial.print("Rata-Rata Kelembaban : ");
Serial.println(bab);

//perhitungan fuzzy
fuzzy->setInput(1, uhu);
fuzzy->setInput(2, bab);
fuzzy->fuzzify();

float defuzkipas = fuzzy->defuzzify(1); //Output range 0-100%
float defuzheater = fuzzy->defuzzify(2); //Output range 0-100%

float pwmkipas = map(defuzkipas, 0, 100, 0, 255); //Merubah range 0-100 menjadi 0-255 (nilai pwm)
// float onoffheater = map(defuzheater, 0, 100, 0, 10); //Merubah range 0-100 menjadi 0-255 (nilai pwm)

Serial.println("\n===== OUTPUT =====");
// cetak nilai defuzifikasi, pwm, suhu, dan kelembaban
Serial.print("Nilai Defuzifikasi Kipas : ");
Serial.println(defuzkipas);
Serial.print("PWM Kipas : ");
Serial.println(pwmkipas);
Serial.print("Nilai Heater Hidup : ");
Serial.println(hidup->getPertinence());
Serial.print("Nilai Heater Mati : ");
Serial.println(mati->getPertinence());
Serial.print("Defuzifikasi Heater : ");
Serial.println(defuzheater);

if (defuzheater >= 60){
  // menonaktifkan pin SSR dengan LOW (tidak mengalirkan arus listrik)
  digitalWrite(pinSSR, LOW);
  Serial.println("Heater : Mati");
} else {
  digitalWrite(pinSSR, HIGH);
  Serial.println("Heater : Hidup");
}
}

```

```

// cetak kekuatan kipas (pwm 0-255), sesuaikan dengan rule fuzzy
if(pwmkipas == 0){
  Serial.println("Kecepatan Kipas Mati");
} else if (pwmkipas > 0 && pwmkipas < 102){
  Serial.println("Kecepatan Kipas Rendah");
} else if (pwmkipas >= 102 && pwmkipas <= 178.5){
  Serial.println("Kecepatan Kipas Sedang");
} else if (pwmkipas > 178.5 && pwmkipas <= 255){
  Serial.println("Kecepatan Kipas Cepat");
}

// assign pin enableAB sesuai nilai pwmkipas
digitalWrite(enableAB, pwmkipas);
digitalWrite(pinA, HIGH); //mengaktifkan pina
digitalWrite(pinB, LOW); //mematikan pinb

delay(30000); //30 detik
}

```

Fuzzy 50°C

```

#include <Fuzzy.h>
#include "DHT.h"
#define DHT_PIN D5 // The ESP8266 pin D5 connected to DHT22 sensor
#define DHT_PIN1 D6 // The ESP8266 pin D6 connected to DHT22 sensor
#define DHT_TYPE DHT22
#define pinSSR 13

// pin L298n
int pinA = D2;
int pinB = D3;
int enableAB = D1;

// package dht22
DHT dht22(DHT_PIN, DHT_TYPE);
DHT dht221(DHT_PIN1, DHT_TYPE);

Fuzzy *fuzzy = new Fuzzy(); //deklarasi fuzzy sebuah variabel, bisa diubah sesuai keinginan

//Fuzzy input suhu
FuzzySet *dingin = new FuzzySet(0, 0, 10, 20);
FuzzySet *normal = new FuzzySet(15, 25, 25, 35);
FuzzySet *panas = new FuzzySet(30, 42, 42, 52);
FuzzySet *sangat_panas = new FuzzySet(50, 60, 100, 100);

//Fuzzy input kelembaban
FuzzySet *kering = new FuzzySet(0, 0, 20, 30);
FuzzySet *lembab = new FuzzySet(20, 35, 35, 50);
FuzzySet *basah = new FuzzySet(40, 50, 100, 100);

//Fuzzy output kipas
// FuzzySet *mati = new FuzzySet(0, 0, 0, 0);
FuzzySet *rendah = new FuzzySet(0, 0, 15.5, 23);
FuzzySet *sedang = new FuzzySet(16, 36, 36, 56);
FuzzySet *cepat = new FuzzySet(49, 69, 100, 100);

//Fuzzy output heater
FuzzySet *hidup = new FuzzySet(0, 0, 30, 50);
FuzzySet *mati = new FuzzySet(30, 50, 100, 100);

```

```
// fungsi untuk membuat fuzzifikasi
void declareFuzzy()
{

    // set FuzzyInput suhu
    FuzzyInput *suhu = new FuzzyInput(1);
    suhu->addFuzzySet(dingin);
    suhu->addFuzzySet(normal);
    suhu->addFuzzySet(panas);
    suhu->addFuzzySet(sangat_panas);
    fuzzy->addFuzzyInput(suhu);

    // set FuzzyInput kelembaban
    FuzzyInput *kelembaban = new FuzzyInput(2);
    kelembaban->addFuzzySet(kering);
    kelembaban->addFuzzySet(lembab);
    kelembaban->addFuzzySet(basah);
    fuzzy->addFuzzyInput(kelembaban);

    // set fuzzyOutput kipas
    FuzzyOutput *kipas = new FuzzyOutput(1);
    kipas->addFuzzySet(rendah);
    kipas->addFuzzySet(sedang);
    kipas->addFuzzySet(cepat);
    fuzzy->addFuzzyOutput(kipas);

    // set fuzzyOutput heater
    FuzzyOutput *heater = new FuzzyOutput(2);
    heater->addFuzzySet(hidup);
    heater->addFuzzySet(mati);
    fuzzy->addFuzzyOutput(heater);

    //Building fuzzy rule
    //SuhuDingin_KelembabanKering_KipasRendah
    FuzzyRuleAntecedent *ifDinginAndKering = new FuzzyRuleAntecedent(); //fuzzy rule input
    ifDinginAndKering->joinWithAND(dingin, kering);

    FuzzyRuleConsequent *thenKipasRendahHeaterHidup = new FuzzyRuleConsequent(); //fuzzy rule output
    thenKipasRendahHeaterHidup->addOutput(rendah);
    thenKipasRendahHeaterHidup->addOutput(hidup);

    FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifDinginAndKering, thenKipasRendahHeaterHidup);
    fuzzy->addFuzzyRule(fuzzyRule01);

    //SuhuDingin_KelembabanLembab_KipasCepat
    FuzzyRuleAntecedent *ifDinginAndLembab = new FuzzyRuleAntecedent();
    ifDinginAndLembab->joinWithAND(dingin, lembab);

    FuzzyRuleConsequent *thenKipasSedangHeaterHidup = new FuzzyRuleConsequent();
    thenKipasSedangHeaterHidup->addOutput(sedang);
    thenKipasSedangHeaterHidup->addOutput(hidup);

    FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifDinginAndLembab, thenKipasSedangHeaterHidup);
    fuzzy->addFuzzyRule(fuzzyRule02);

    //SuhuDingin_KelembabanBasah_KipasSedang
    FuzzyRuleAntecedent *ifDinginAndBasah = new FuzzyRuleAntecedent();
    ifDinginAndBasah->joinWithAND(dingin, basah);
```



```
FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifDinginAndBasah, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule03);

//SuhuNormal_KelembabanKering_KipasRendah
FuzzyRuleAntecedent *ifNormalAndKering = new FuzzyRuleAntecedent();
ifNormalAndKering->joinWithAND(normal, kering);

FuzzyRuleConsequent *thenKipasCepatHeaterHidup = new FuzzyRuleConsequent();
thenKipasCepatHeaterHidup->addOutput(cepat);
thenKipasCepatHeaterHidup->addOutput(hidup);

FuzzyRule *fuzzyRule04 = new FuzzyRule(4, ifNormalAndKering, thenKipasCepatHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule04);

//SuhuNormal_KelembabanLembab_KipasSedang
FuzzyRuleAntecedent *ifNormalAndLembab = new FuzzyRuleAntecedent();
ifNormalAndLembab->joinWithAND(normal, lembab);

FuzzyRule *fuzzyRule05 = new FuzzyRule(5, ifNormalAndLembab, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule05);

//SuhuNormal_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifNormalAndBasah = new FuzzyRuleAntecedent();
ifNormalAndBasah->joinWithAND(normal, basah);

FuzzyRule *fuzzyRule06 = new FuzzyRule(6, ifNormalAndBasah, thenKipasRendahHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule06);

//SuhuPanas_KelembabanKering_KipasMati
FuzzyRuleAntecedent *ifPanasAndKering = new FuzzyRuleAntecedent();
ifPanasAndKering->joinWithAND(panas, kering);

FuzzyRule *fuzzyRule07 = new FuzzyRule(7, ifPanasAndKering, thenKipasCepatHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule07);

//SuhuPanas_KelembabanLembab_KipasSedang
FuzzyRuleAntecedent *ifPanasAndLembab = new FuzzyRuleAntecedent();
ifPanasAndLembab->joinWithAND(panas, lembab);

FuzzyRule *fuzzyRule08 = new FuzzyRule(8, ifPanasAndLembab, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule08);

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifPanasAndBasah = new FuzzyRuleAntecedent();
ifPanasAndBasah->joinWithAND(panas, basah);

FuzzyRule *fuzzyRule09 = new FuzzyRule(9, ifPanasAndBasah, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule09);

//SuhuSangatPanas_KelembabanKering_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndKering = new FuzzyRuleAntecedent();
ifSangatPanasAndKering->joinWithAND(sangat_panas, kering);

FuzzyRuleConsequent *thenKipasCepatHeaterMati = new FuzzyRuleConsequent();
thenKipasCepatHeaterMati->addOutput(cepat);
thenKipasCepatHeaterMati->addOutput(mati);

FuzzyRule *fuzzyRule10 = new FuzzyRule(10, ifSangatPanasAndKering, thenKipasCepatHeaterMati);
fuzzy->addFuzzyRule(fuzzyRule10);
```

```

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndLembab = new FuzzyRuleAntecedent();
ifSangatPanasAndLembab->joinWithAND(sangat_panas, lembab);

FuzzyRule *fuzzyRule11 = new FuzzyRule(11, ifSangatPanasAndLembab, thenKipasCepatHeaterMati);
fuzzy->addFuzzyRule(fuzzyRule11);

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndBasah = new FuzzyRuleAntecedent();
ifSangatPanasAndBasah->joinWithAND(sangat_panas, basah);

FuzzyRule *fuzzyRule12 = new FuzzyRule(12, ifSangatPanasAndBasah, thenKipasCepatHeaterMati);
fuzzy->addFuzzyRule(fuzzyRule12);
}

void setup ()
{
  Serial.begin(9600); //setting serial board ke monitor

  //setting pin menjadi OUTPUT
  pinMode(pinA, OUTPUT);
  pinMode(pinB, OUTPUT);
  pinMode(enableAB, OUTPUT);
  pinMode(pinSSR, OUTPUT);

  //setting pin SSR menjadi nyala/HIGH (mengalirkan listrik)
  digitalWrite(pinSSR, HIGH);

  //dht22
  dht22.begin(); // initialize the DHT sensor
  dht221.begin(); // initialize the DHT sensor

  // pemanggilan fungsi declareFuzzy agar dapat digunakan
  declareFuzzy();
}

void loop()
{
  // dht22
  // baca kelembaban
  float humi = dht22.readHumidity(); //pake ini
  // float humi = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor
  // baca suhu
  float temperature_C = dht22.readTemperature(); //pake ini
  // float temperature_C = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor

  // baca kelembaban
  float humi1 = dht221.readHumidity(); //pake ini
  // float humi1 = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor
  // baca suhu
  float temperature_C1 = dht221.readTemperature(); //pake ini
  // float temperature_C1 = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor

  // mencari nilai rata2 dari kedua sensor
  float uhu = (temperature_C + temperature_C1) / 2;
  float bab = (humi + humi1) / 2;

  Serial.println("\n===== INPUT =====");
  // cek apakah sensor berfungsi dengan benar

```

```

if ( isNaN(temperature_C && temperature_C1) || isNaN(humi && humi1)) {
  Serial.println("Gagal membaca sensor!");
}
// cetak nilai sensor
else {
  Serial.print("Kelembaban Atas: ");
  Serial.print(humi);
  Serial.print("%");
  Serial.print(" | ");

  Serial.print("Suhu Atas: ");
  Serial.print(temperature_C);
  Serial.println("°C");

  Serial.print("Kelembaban Bawah: ");
  Serial.print(humi1);
  Serial.print("%");

  Serial.print(" | ");

  Serial.print("Suhu Bawah: ");
  Serial.print(temperature_C1);
  Serial.println("°C");
}

Serial.print("Rata-Rata Suhu : ");
Serial.println(uhu);
Serial.print("Rata-Rata Kelembaban : ");
Serial.println(bab);

//perhitungan fuzzy
fuzzy->setInput(1, uhu);
fuzzy->setInput(2, bab);
fuzzy->fuzzify();

float defuzkipas = fuzzy->defuzzify(1); //Output range 0-100%
float defuzheater = fuzzy->defuzzify(2); //Output range 0-100%

float pwmkipas = map(defuzkipas, 0, 100, 0, 255); //Merubah range 0-100 menjadi 0-255 (nilai pwm)
// float onoffheater = map(defuzheater, 0, 100, 0, 10); //Merubah range 0-100 menjadi 0-255 (nilai pwm)

Serial.println("\n===== OUTPUT =====");
// cetak nilai defuzifikasi, pwm, suhu, dan kelembaban
Serial.print("Nilai Defuzifikasi Kipas : ");
Serial.println(defuzkipas);
Serial.print("PWM Kipas : ");
Serial.println(pwmkipas);
Serial.print("Nilai Heater Hidup : ");
Serial.println(hidup->getPertinence());
Serial.print("Nilai Heater Mati : ");
Serial.println(mati->getPertinence());
Serial.print("Defuzifikasi Heater : ");
Serial.println(defuzheater);

if (defuzheater >= 50){
  // menonaktifkan pin SSR dengan LOW (tidak mengalirkan arus listrik)
  digitalWrite(pinSSR, LOW);
  Serial.println("Heater : Mati");
} else {
  digitalWrite(pinSSR, HIGH);
}

```

```

Serial.println("Heater : Hidup");
}

// cetak kekuatan kipas (pwm 0-255), sesuaikan dengan rule fuzzy
if(pwmkipas == 0){
  Serial.println("Kecepatan Kipas Mati");
} else if (pwmkipas > 0 && pwmkipas < 102){
  Serial.println("Kecepatan Kipas Rendah");
} else if (pwmkipas >= 102 && pwmkipas <= 178.5){
  Serial.println("Kecepatan Kipas Sedang");
} else if (pwmkipas > 178.5 && pwmkipas <= 255){
  Serial.println("Kecepatan Kipas Cepat");
}

// assign pin enableAB sesuai nilai pwmkipas
digitalWrite(enableAB, pwmkipas);
digitalWrite(pinA, HIGH); //mengaktifkan pina
digitalWrite(pinB, LOW); //mematikan pinb

delay(30000); //30 detik
}

```

Fuzzy 40°C

```

#include <Fuzzy.h>
#include "DHT.h"
#define DHT_PIN D5 // The ESP8266 pin D5 connected to DHT22 sensor
#define DHT_PIN1 D6 // The ESP8266 pin D6 connected to DHT22 sensor
#define DHT_TYPE DHT22
#define pinSSR 13

// pin L298n
int pinA = D2;
int pinB = D3;
int enableAB = D1;

// package dht22
DHT dht22(DHT_PIN, DHT_TYPE);
DHT dht221(DHT_PIN1, DHT_TYPE);

Fuzzy *fuzzy = new Fuzzy(); //deklarasi fuzzy sebuah variabel, bisa diubah sesuai keinginan

//Fuzzy input suhu
FuzzySet *dingin = new FuzzySet(0, 0, 10, 15);
FuzzySet *normal = new FuzzySet(10, 15, 15, 25);
FuzzySet *panas = new FuzzySet(20, 32, 32, 42);
FuzzySet *sangat_panas = new FuzzySet(40, 50, 100, 100);

//Fuzzy input kelembaban
FuzzySet *kering = new FuzzySet(0, 0, 10, 20);
FuzzySet *lembab = new FuzzySet(10, 25, 25, 40);
FuzzySet *basah = new FuzzySet(30, 40, 100, 100);

//Fuzzy output kipas
// FuzzySet *mati = new FuzzySet(0, 0, 0, 0);
FuzzySet *rendah = new FuzzySet(0, 0, 15.5, 23);
FuzzySet *sedang = new FuzzySet(16, 26, 26, 46);
FuzzySet *cepat = new FuzzySet(39, 59, 100, 100);

//Fuzzy output heater

```

```

FuzzySet *hidup = new FuzzySet(0, 0, 20, 40);
FuzzySet *mati = new FuzzySet(20, 40, 100, 100);

// fungsi untuk membuat fuzzifikasi
void declareFuzzy()
{

    // set FuzzyInput suhu
    FuzzyInput *suhu = new FuzzyInput(1);
    suhu->addFuzzySet(dingin);
    suhu->addFuzzySet(normal);
    suhu->addFuzzySet(panas);
    suhu->addFuzzySet(sangat_panas);
    fuzzy->addFuzzyInput(suhu);

    // set FuzzyInput kelembaban
    FuzzyInput *kelembaban = new FuzzyInput(2);
    kelembaban->addFuzzySet(kering);
    kelembaban->addFuzzySet(lembab);
    kelembaban->addFuzzySet(basah);
    fuzzy->addFuzzyInput(kelembaban);

    // set fuzzyOutput kipas
    FuzzyOutput *kipas = new FuzzyOutput(1);
    kipas->addFuzzySet(rendah);
    kipas->addFuzzySet(sedang);
    kipas->addFuzzySet(cepat);
    fuzzy->addFuzzyOutput(kipas);

    // set fuzzyOutput heater
    FuzzyOutput *heater = new FuzzyOutput(2);
    heater->addFuzzySet(hidup);
    heater->addFuzzySet(mati);
    fuzzy->addFuzzyOutput(heater);

    //Building fuzzy rule
    //SuhuDingin_KelembabanKering_KipasRendah
    FuzzyRuleAntecedent *ifDinginAndKering = new FuzzyRuleAntecedent(); //fuzzy rule input
    ifDinginAndKering->joinWithAND(dingin, kering);

    FuzzyRuleConsequent *thenKipasRendahHeaterHidup = new FuzzyRuleConsequent(); //fuzzy rule output
    thenKipasRendahHeaterHidup->addOutput(rendah);
    thenKipasRendahHeaterHidup->addOutput(hidup);

    FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifDinginAndKering, thenKipasRendahHeaterHidup);
    fuzzy->addFuzzyRule(fuzzyRule01);

    //SuhuDingin_KelembabanLembab_KipasCepat
    FuzzyRuleAntecedent *ifDinginAndLembab = new FuzzyRuleAntecedent();
    ifDinginAndLembab->joinWithAND(dingin, lembab);

    FuzzyRuleConsequent *thenKipasSedangHeaterHidup = new FuzzyRuleConsequent();
    thenKipasSedangHeaterHidup->addOutput(sedang);
    thenKipasSedangHeaterHidup->addOutput(hidup);

    FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifDinginAndLembab, thenKipasSedangHeaterHidup);
    fuzzy->addFuzzyRule(fuzzyRule02);

    //SuhuDingin_KelembabanBasah_KipasSedang
    FuzzyRuleAntecedent *ifDinginAndBasah = new FuzzyRuleAntecedent();

```

```
ifDinginAndBasah->joinWithAND(dingin, basah);

FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifDinginAndBasah, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule03);

//SuhuNormal_KelembabanKering_KipasRendah
FuzzyRuleAntecedent *ifNormalAndKering = new FuzzyRuleAntecedent();
ifNormalAndKering->joinWithAND(normal, kering);

FuzzyRuleConsequent *thenKipasCepatHeaterHidup = new FuzzyRuleConsequent();
thenKipasCepatHeaterHidup->addOutput(cepat);
thenKipasCepatHeaterHidup->addOutput(hidup);

FuzzyRule *fuzzyRule04 = new FuzzyRule(4, ifNormalAndKering, thenKipasCepatHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule04);

//SuhuNormal_KelembabanLembab_KipasSedang
FuzzyRuleAntecedent *ifNormalAndLembab = new FuzzyRuleAntecedent();
ifNormalAndLembab->joinWithAND(normal, lembab);

FuzzyRule *fuzzyRule05 = new FuzzyRule(5, ifNormalAndLembab, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule05);

//SuhuNormal_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifNormalAndBasah = new FuzzyRuleAntecedent();
ifNormalAndBasah->joinWithAND(normal, basah);

FuzzyRule *fuzzyRule06 = new FuzzyRule(6, ifNormalAndBasah, thenKipasRendahHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule06);

//SuhuPanas_KelembabanKering_KipasMati
FuzzyRuleAntecedent *ifPanasAndKering = new FuzzyRuleAntecedent();
ifPanasAndKering->joinWithAND(panas, kering);

FuzzyRule *fuzzyRule07 = new FuzzyRule(7, ifPanasAndKering, thenKipasCepatHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule07);

//SuhuPanas_KelembabanLembab_KipasSedang
FuzzyRuleAntecedent *ifPanasAndLembab = new FuzzyRuleAntecedent();
ifPanasAndLembab->joinWithAND(panas, lembab);

FuzzyRule *fuzzyRule08 = new FuzzyRule(8, ifPanasAndLembab, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule08);

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifPanasAndBasah = new FuzzyRuleAntecedent();
ifPanasAndBasah->joinWithAND(panas, basah);

FuzzyRule *fuzzyRule09 = new FuzzyRule(9, ifPanasAndBasah, thenKipasSedangHeaterHidup);
fuzzy->addFuzzyRule(fuzzyRule09);

//SuhuSangatPanas_KelembabanKering_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndKering = new FuzzyRuleAntecedent();
ifSangatPanasAndKering->joinWithAND(sangat_panas, kering);

FuzzyRuleConsequent *thenKipasCepatHeaterMati = new FuzzyRuleConsequent();
thenKipasCepatHeaterMati->addOutput(cepat);
thenKipasCepatHeaterMati->addOutput(mati);

FuzzyRule *fuzzyRule10 = new FuzzyRule(10, ifSangatPanasAndKering, thenKipasCepatHeaterMati);
```

```

fuzzy->addFuzzyRule(fuzzyRule10);

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndLembab = new FuzzyRuleAntecedent();
ifSangatPanasAndLembab->joinWithAND(sangat_panas, lembab);

FuzzyRule *fuzzyRule11 = new FuzzyRule(11, ifSangatPanasAndLembab, thenKipasCepatHeaterMati);
fuzzy->addFuzzyRule(fuzzyRule11);

//SuhuPanas_KelembabanBasah_KipasCepat
FuzzyRuleAntecedent *ifSangatPanasAndBasah = new FuzzyRuleAntecedent();
ifSangatPanasAndBasah->joinWithAND(sangat_panas, basah);

FuzzyRule *fuzzyRule12 = new FuzzyRule(12, ifSangatPanasAndBasah, thenKipasCepatHeaterMati);
fuzzy->addFuzzyRule(fuzzyRule12);
}

void setup ()
{
  Serial.begin(9600); //setting serial board ke monitor

  //setting pin menjadi OUTPUT
  pinMode(pinA, OUTPUT);
  pinMode(pinB, OUTPUT);
  pinMode(enableAB, OUTPUT);
  pinMode(pinSSR, OUTPUT);

  //setting pin SSR menjadi nyala/HIGH (mengalirkan listrik)
  digitalWrite(pinSSR, HIGH);

  //dht22
  dht22.begin(); // initialize the DHT sensor
  dht221.begin(); // initialize the DHT sensor

  // pemanggilan fungsi declareFuzzy agar dapat digunakan
  declareFuzzy();
}

void loop()
{
  // dht22
  // baca kelembaban
  float humi = dht22.readHumidity(); //pake ini
  // float humi = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor
  // baca suhu
  float temperature_C = dht22.readTemperature(); //pake ini
  // float temperature_C = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor

  // baca kelembaban
  float humi1 = dht221.readHumidity(); //pake ini
  // float humi1 = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor
  // baca suhu
  float temperature_C1 = dht221.readTemperature(); //pake ini
  // float temperature_C1 = random(100); // random nilai aja, dihapus kalau udah bisa ambil data dari sensor

  // mencari nilai rata2 dari kedua sensor
  float uhu = (temperature_C + temperature_C1) / 2;
  float bab = (humi + humi1) / 2;

```

```
Serial.println("\n===== INPUT =====");
// cek apakah sensor berfungsi dengan benar
if ( isnan(temperature_C && temperature_C1) || isnan(humi && humi1)) {
  Serial.println("Gagal membaca sensor!");
}
// cetak nilai sensor
else {
  Serial.print("Kelembaban Atas: ");
  Serial.print(humi);
  Serial.print("%");
  Serial.print(" | ");

  Serial.print("Suhu Atas: ");
  Serial.print(temperature_C);
  Serial.println("°C");

  Serial.print("Kelembaban Bawah: ");
  Serial.print(humi1);
  Serial.print("%");

  Serial.print(" | ");

  Serial.print("Suhu Bawah: ");
  Serial.print(temperature_C1);
  Serial.println("°C");
}

Serial.print("Rata-Rata Suhu : ");
Serial.println(uhu);
Serial.print("Rata-Rata Kelembaban : ");
Serial.println(bab);

//perhitungan fuzzy
fuzzy->setInput(1, uhu);
fuzzy->setInput(2, bab);
fuzzy->fuzzify();

float defuzkipas = fuzzy->defuzzify(1); //Output range 0-100%
float defuzheater = fuzzy->defuzzify(2); //Output range 0-100%

float pwmkipas = map(defuzkipas, 0, 100, 0, 255); //Merubah range 0-100 menjadi 0-255 (nilai pwm)
// float onoffheater = map(defuzheater, 0, 100, 0, 10); //Merubah range 0-100 menjadi 0-255 (nilai pwm)

Serial.println("\n===== OUTPUT =====");
// cetak nilai defuzifikasi, pwm, suhu, dan kelembaban
Serial.print("Nilai Defuzifikasi Kipas : ");
Serial.println(defuzkipas);
Serial.print("PWM Kipas : ");
Serial.println(pwmkipas);
Serial.print("Nilai Heater Hidup : ");
Serial.println(hidup->getPertinence());
Serial.print("Nilai Heater Mati : ");
Serial.println(mati->getPertinence());
Serial.print("Defuzifikasi Heater : ");
Serial.println(defuzheater);

if (defuzheater >= 40){
  // menonaktifkan pin SSR dengan LOW (tidak mengalirkan arus listrik)
  digitalWrite(pinSSR, LOW);
  Serial.println("Heater : Mati");
}
```



```

} else {
  digitalWrite(pinSSR, HIGH);
  Serial.println("Heater : Hidup");
}

// cetak kekuatan kipas (pwm 0-255), sesuaikan dengan rule fuzzy
if(pwmkipas == 0){
  Serial.println("Kecepatan Kipas Mati");
} else if (pwmkipas > 0 && pwmkipas < 102){
  Serial.println("Kecepatan Kipas Rendah");
} else if (pwmkipas >= 102 && pwmkipas <= 178.5){
  Serial.println("Kecepatan Kipas Sedang");
} else if (pwmkipas > 178.5 && pwmkipas <= 255){
  Serial.println("Kecepatan Kipas Cepat");
}

// assign pin enableAB sesuai nilai pwmkipas
digitalWrite(enableAB, pwmkipas);
digitalWrite(pinA, HIGH); //mengaktifkan pina
digitalWrite(pinB, LOW); //mematikan pinb

delay(30000); //30 detik
}

```

Lampiran 8. Code Program Website

Index.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Telang's Dehydrator</title>

  <script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-app.js"></script>
  <!-- <script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-auth.js"></script> -->
  <script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-database.js"></script>

  <script>
    const firebaseConfig = {
      apiKey: "AIzaSyCXrtGgViyTakciQsit3wR50KQDNiklz8k",
      authDomain: "tugas-akhir-254b5.firebaseio.com",
      databaseURL: "https://tugas-akhir-254b5-default-rtdb.asia-southeast1.firebaseio.com",
      projectId: "tugas-akhir-254b5",
      storageBucket: "http://tugas-akhir-254b5.appspot.com",
      messagingSenderId: "417599358443",
      appId: "1:417599358443:web:0524af7026c25f3035c5ae"
    };

    firebase.initializeApp(firebaseConfig);

    // const auth = firebase.auth();
    const db = firebase.database();
  </script>
  <!-- include highchartsjs to build the charts-->
  <script src="https://code.highcharts.com/highcharts.js"></script>
  <!-- include to use jquery-->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <!--include icons from fontawesome-->

```

```

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
crossorigin="anonymous">
<link rel="icon" type="image/png" href="favicon.png">
<link rel="stylesheet" type="text/css" href="style.css">
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>

<body onload=display_ct();>
<div class="topnav">
<h1>Telang's Dehydrator <i class="fas fa-clipboard-list"></i></h1>
</div>

<div class="content-sign-in" id="content-sign-in">
<div class="cards-2">
<div class="card">
<h4><b>On / Off</b></h4>
<label class="switch">
<input type="checkbox" id="toggle-switch">
<span class="slider round"></span>
</label>
</div>
<div class="card">
<h4><b>Date and Time</b></h4>
<span id='ct' ></span>
</div>
</div>
</div>

<h3><span class="date-time">Last update: <span id="lastUpdate"></span></span></h3>

<div class="cards">
<div class="card">
<p><i class="fas fa-thermometer-half" style="color:#FF5733;"></i> TEMPERATURE</p>
<p><span class="reading"><span id="suhu"></span> &deg;C</span></p>
<!-- <canvas id="temperatureChart"></canvas> -->
</div>
<div class="card">
<p><i class="fas fa-tint" style="color:#00add6;"></i> HUMIDITY</p>
<p><span class="reading"><span id="kelembaban"></span> &percent;</span></p>
<!-- <canvas id="humidityChart"></canvas> -->
</div>
<div class="card">
<p><i class="fas fa-angle-double-down" style="color:#e1e437;"></i> FAN SPEED</p>
<p><span class="reading"><span id="kipas"></span></p>
</div>
<div class="card">
<p><i class="fas fa-fire" style="color:#e90707;"></i> HEATER</p>
<p><span class="reading"><span id="heater"></span></p>
</div>
</div>
<h3><span class="date-time">Tercatat data terakhir yang didapat pada <span id="lastUpdate2"></span>
Suhu <span id="suhu2"></span> dan Kelembaban <span id="kelembaban2"></span> dengan Kecepatan
Kipas <span id="kipas2"></span> dan Heater <span id="heater2"></span></h3>

<div id="charts-div">
<!--SET NUMBER OF READINGS INPUT FIELD-->
<!-- <div>
<p> Number of readings: <input type="number" id="charts-range"></p>
</div> -->
<!--TEMPERATURE-CHART-->
<div class="cards-2">

```

```

    <div class="card">
      <p><i class="fas fa-thermometer-half" style="color:#059e8a;"></i> TEMPERATURE & HUMIDITY
      CHART</p>
      <div id="chart-temperature" class="chart-container"></div>
    </div>
  </div>
  <!--HUMIDITY-CHART-->
  <!-- <div class="cards">
    <div class="card">
      <p><i class="fas fa-tint" style="color:#00add6;"></i> HUMIDITY CHART</p>
      <div id="chart-humidity" class="chart-container"></div>
    </div>
  </div> -->
</div>

<!--BUTTONS TO HANDLE DATA-->
<p>
  <!--View data button-->
  <button id="view-data-button">View all data</button>
  <!--Hide data button-->
  <button id="hide-data-button" style="display:none;">Hide data</button>
  <!--Delete data button-->
  <button id="delete-button" class="deletebtn">Delete data</button>
</p>
<!--Modal to delete data-->
<div id="delete-modal" class="modal" style="display:none">
  <span onclick = "document.getElementById('delete-modal').style.display='none'" class="close"
  title="Close Modal"></span>
  <form id="delete-data-form" class="modal-content" action="">
    <div class="container">
      <h1>Delete Data</h1>
      <p>Are you sure you want to delete all data from database?</p>
      <div class="clearfix">
        <button type="button" onclick="document.getElementById('delete-modal').style.display='none'"
        class="cancelbtn">Cancel</button>
        <button type="submit" onclick="document.getElementById('delete-modal').style.display='none'"
        class="deletebtn">Delete</button>
      </div>
    </div>
  </form>
</div>

<!--TABLE WITH ALL DATA-->
<div class="cards-2">
  <div class="card" id="table-container" style="display:none;">
    <table id="readings-table">
      <tr id="thead">
        <th>Timestamp</th>
        <th>Temp (°C)</th>
        <th>Hum (%)</th>
        <th>Fan State</th>
        <th>Heater</th>
      </tr>
      <tbody id="tbody">
      </tbody>
    </table>
    <!-- <p><button id="load-data" style="display:none;">More results...</button></p> -->
  </div>
</div>

```

```

<h2 style="color: #6A1B9A; text-align: center; margin-top: 20px;">FUZZY RULE</h2>
<div class ="cards-2">
  <table class="fuzzy-table">
    <thead>
      <tr>
        <th>Suhu</th>
        <th>Kelembaban</th>
        <th>Kipas</th>
        <th>Heater</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Dingin</td>
        <td>Kering</td>
        <td>Rendah</td>
        <td>Hidup</td>
      </tr>
      <tr>
        <td>Dingin</td>
        <td>Lembab</td>
        <td>Sedang</td>
        <td>Hidup</td>
      </tr>
      <tr>
        <td>Dingin</td>
        <td>Basah</td>
        <td>Sedang</td>
        <td>Hidup</td>
      </tr>
      <tr>
        <td>Normal</td>
        <td>Kering</td>
        <td>Cepat</td>
        <td>Hidup</td>
      </tr>
      <tr>
        <td>Normal</td>
        <td>Lembab</td>
        <td>Sedang</td>
        <td>Hidup</td>
      </tr>
      <tr>
        <td>Normal</td>
        <td>Basah</td>
        <td>Sedang</td>
        <td>Hidup</td>
      </tr>
      <tr>
        <td>Panas</td>
        <td>Kering</td>
        <td>Cepat</td>
        <td>Hidup</td>
      </tr>
      <tr>
        <td>Panas</td>
        <td>Lembab</td>
        <td>Sedang</td>
        <td>Hidup</td>
      </tr>
    </tbody>
  </table>

```

```

<tr>
  <td>Panas</td>
  <td>Basah</td>
  <td>Cepat</td>
  <td>Hidup</td>
</tr>
<tr>
  <td>Sangat Panas</td>
  <td>Kering</td>
  <td>Cepat</td>
  <td>Mati</td>
</tr>
<tr>
  <td>Sangat Panas</td>
  <td>Kering</td>
  <td>Cepat</td>
  <td>Mati</td>
</tr>
<tr>
  <td>Sangat Panas</td>
  <td>Kering</td>
  <td>Cepat</td>
  <td>Mati</td>
</tr>
</tbody>
</table>
</div>
</div>
<!-- <script src="scripts/auth.js"></script> -->
<script src="scripts/index.js"></script>
<script src="scripts/charts-definition.js"></script>
</body>
</html>

```

Lampiran 9. Code perhitungan RGB

```

classdef tadenca < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure          matlab.ui.Figure
        BlueEditField      matlab.ui.control.EditField
        BlueEditFieldLabel matlab.ui.control.Label
        GreenEditField     matlab.ui.control.EditField
        GreenEditFieldLabel matlab.ui.control.Label
        RedEditField       matlab.ui.control.EditField
        RedEditFieldLabel  matlab.ui.control.Label
        HitungRGBButton    matlab.ui.control.Button
        BukaCitraButton    matlab.ui.control.Button
        UIAxes             matlab.ui.control.UIAxes
    end

    properties (Access = public)
        Property % Description
    end

    % Callbacks that handle component events

```

```

methods (Access = private)

% Button pushed function: BukaCitraButton
function BukaCitraButtonPushed(app, event)
    [nama_file, nama_folder] = uigetfile('*.*');
    drawnow
    figure(app.UIFigure)

    if ~isequal(nama_file,0)
        Img = imread(fullfile(nama_folder,nama_file));

        imshow(Img,'Parent',app.UIAxes)
        title(app.UIAxes,'Citra RGB')

        app.Property.Img = Img;
    end
end
% Button pushed function: HitungRGBButton
function HitungRGBButtonPushed(app, event)
    Img = app.Property.Img;

    R = Img(:,:,1);
    G = Img(:,:,2);
    B = Img(:,:,3);

    Red = mean(mean(R));
    Green = mean(mean(G));
    Blue = mean(mean(B));

    app.RedEditField.Value = num2str(Red);
    app.GreenEditField.Value = num2str(Green);
    app.BlueEditField.Value = num2str(Blue);
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app.UIFigure = uifigure('Visible', 'off');
    app.UIFigure.Position = [100 100 640 480];
    app.UIFigure.Name = 'MATLAB App';

    % Create UIAxes
    app.UIAxes = uiaxes(app.UIFigure);
    app.UIAxes.XTick = [];
    app.UIAxes.YTick = [];
    app.UIAxes.Position = [223 214 300 185];

    % Create BukaCitraButton
    app.BukaCitraButton = uibutton(app.UIFigure, 'push');
    app.BukaCitraButton.ButtonPushedFcn = createCallbackFcn(app, @BukaCitraButtonPushed, true);
    app.BukaCitraButton.Position = [68 377 100 23];
    app.BukaCitraButton.Text = 'Buka Citra';

    % Create HitungRGBButton
    app.HitungRGBButton = uibutton(app.UIFigure, 'push');

```

```

app.HitungRGBButton.ButtonPushedFcn = createCallbackFcn(app, @HitungRGBButtonPushed, true);
app.HitungRGBButton.Position = [68 333 100 23];
app.HitungRGBButton.Text = 'Hitung RGB';

% Create RedEditFieldLabel
app.RedEditFieldLabel = uilabel(app.UIFigure);
app.RedEditFieldLabel.HorizontalAlignment = 'right';
app.RedEditFieldLabel.Position = [249 172 28 22];
app.RedEditFieldLabel.Text = 'Red';

% Create RedEditField
app.RedEditField = uieditfield(app.UIFigure, 'text');
app.RedEditField.BackgroundColor = [1 0 0];
app.RedEditField.Position = [292 172 100 22];

% Create GreenEditFieldLabel
app.GreenEditFieldLabel = uilabel(app.UIFigure);
app.GreenEditFieldLabel.HorizontalAlignment = 'right';
app.GreenEditFieldLabel.Position = [238 131 39 22];
app.GreenEditFieldLabel.Text = 'Green';

% Create GreenEditField
app.GreenEditField = uieditfield(app.UIFigure, 'text');
app.GreenEditField.BackgroundColor = [0 1 0];
app.GreenEditField.Position = [292 131 100 22];

% Create BlueEditFieldLabel
app.BlueEditFieldLabel = uilabel(app.UIFigure);
app.BlueEditFieldLabel.HorizontalAlignment = 'right';
app.BlueEditFieldLabel.Position = [247 87 30 22];
app.BlueEditFieldLabel.Text = 'Blue';

% Create BlueEditField
app.BlueEditField = uieditfield(app.UIFigure, 'text');
app.BlueEditField.BackgroundColor = [0.0745 0.6235 1];
app.BlueEditField.Position = [292 87 100 22];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = tadenca

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargin == 0
clear app
end
end

% Code that executes before app deletion

```

```
function delete(app)
    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end
```